PYTHON ASSIGNMENT

Implement s3 file manager using any python web framework functions

Step 1: Installation of boto3 and flask

```
>>>> pip install flask boto3
```

Step 2: creating the folder and files

```
  s3_manager/
    ├── app.py
    ├── templates/
    │    └── index.html
```

```
Commands: mkdir s3_manager
          nano app.py
          mkdir templates
          nano index.html
```

Step 3 : python code in app.py

```
 from flask import Flask, request, redirect, url_for, render_template
import boto3
from botocore.exceptions import NoCredentialsError, ClientError
import os

app = Flask(__name__)

# Configure AWS credentials and S3 client
AWS_ACCESS_KEY_ID = 'your-access-key-id'
AWS_SECRET_ACCESS_KEY = 'your-secret-access-key'
s3 = boto3.client(
    's3',
    aws_access_key_id=AWS_ACCESS_KEY_ID,
    aws_secret_access_key=AWS_SECRET_ACCESS_KEY
)

@app.route('/')
def index():
    try:
        # List all buckets
        buckets = s3.list_buckets().get('Buckets', [])
        bucket_names = [bucket['Name'] for bucket in buckets]
        return render_template('index.html', bucket_names=bucket_names)
    except NoCredentialsError:
        return "Credentials not available"
    except ClientError as e:
        return f"An error occurred: {e}"
```

```
@app.route('/bucket/<bucket_name>')
def bucket_contents(bucket_name):
    try:
        # List objects in the specified bucket
        objects = s3.list_objects_v2(Bucket=bucket_name).get('Contents', [])
        files = [{'Key': obj['Key']} for obj in objects]
        return render_template('index.html', bucket_name=bucket_name, files=files)
    except NoCredentialsError:
        return "Credentials not available"
    except ClientError as e:
        return f"An error occurred: {e}"

@app.route('/create_bucket', methods=['POST'])
def create_bucket():
    bucket_name = request.form['bucket_name']
    try:
        s3.create_bucket(Bucket=bucket_name)
        return redirect(url_for('index'))
    except NoCredentialsError:
        return "Credentials not available"
    except ClientError as e:
        return f"An error occurred: {e}"

@app.route('/delete_bucket', methods=['POST'])
def delete_bucket():
    bucket_name = request.form['bucket_name']
    try:
        s3.delete_bucket(Bucket=bucket_name)
        return redirect(url_for('index'))
    except NoCredentialsError:
        return "Credentials not available"
    except ClientError as e:
        return f"An error occurred: {e}"

@app.route('/upload', methods=['POST'])
def upload_file():
    bucket_name = request.form['bucket_name']
    file = request.files['file']
    try:
        s3.upload_fileobj(file, bucket_name, file.filename)
        return redirect(url_for('bucket_contents', bucket_name=bucket_name))
    except NoCredentialsError:
        return "Credentials not available"
    except ClientError as e:
        return f"An error occurred: {e}"

@app.route('/delete_file', methods=['POST'])
def delete_file():
    bucket_name = request.form['bucket_name']
    file_key = request.form['file_key']
    try:
        s3.delete_object(Bucket=bucket_name, Key=file_key)
        return redirect(url_for('bucket_contents', bucket_name=bucket_name))
    except NoCredentialsError:
        return "Credentials not available"
    except ClientError as e:
        return f"An error occurred: {e}"

@app.route('/create_folder', methods=['POST'])
def create_folder():
    bucket_name = request.form['bucket_name']
    folder_name = request.form['folder_name']
    try:
        s3.put_object(Bucket=bucket_name, Key=folder_name + '/')
        return redirect(url_for('bucket_contents', bucket_name=bucket_name))
    except NoCredentialsError:
        return "Credentials not available"
    except ClientError as e:
        return f"An error occurred: {e}"

@app.route('/delete_folder', methods=['POST'])
def delete_folder():
    bucket_name = request.form['bucket_name']
    folder_name = request.form['folder_name']
    try:
        objects = s3.list_objects_v2(Bucket=bucket_name, Prefix=folder_name).get('Contents', [])
        for obj in objects:
            s3.delete_object(Bucket=bucket_name, Key=obj['Key'])
        return redirect(url_for('bucket_contents', bucket_name=bucket_name))
    except NoCredentialsError:
        return "Credentials not available"
    except ClientError as e:
        return f"An error occurred: {e}"
```

```
@app.route('/copy_file', methods=['POST'])
def copy_file():
    source_bucket = request.form['source_bucket']
    destination_bucket = request.form['destination_bucket']
    file_key = request.form['file_key']
    try:
        s3.copy_object(
            CopySource={'Bucket': source_bucket, 'Key': file_key},
            Bucket=destination_bucket,
            Key=file_key
        )
        return redirect(url_for('bucket_contents', bucket_name=destination_bucket))
    except NoCredentialsError:
        return "Credentials not available"
    except ClientError as e:
        return f"An error occurred: {e}"

@app.route('/move_file', methods=['POST'])
def move_file():
    source_bucket = request.form['source_bucket']
    destination_bucket = request.form['destination_bucket']
    file_key = request.form['file_key']
    try:
        s3.copy_object(
            CopySource={'Bucket': source_bucket, 'Key': file_key},
            Bucket=destination_bucket,
            Key=file_key
        )
        s3.delete_object(Bucket=source_bucket, Key=file_key)
        return redirect(url_for('bucket_contents', bucket_name=destination_bucket))
    except NoCredentialsError:
        return "Credentials not available"
    except ClientError as e:
        return f"An error occurred: {e}"

if __name__ == '__main__':
    app.run(debug=True)
```

Step 4 : Frontend using HTML and CSS

```
<!DOCTYPE html>
<html>
<head>
```

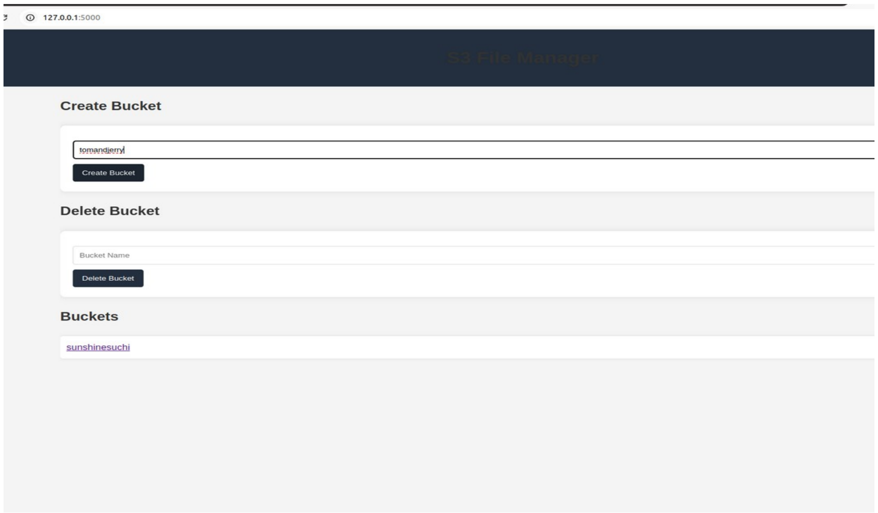
```
<title>S3 File Manager</title>
<style>
    body {
        font-family: Arial, sans-serif;
        line-height: 1.6;
        margin: 0;
        padding: 0;
        background-color: #f4f4f4;
    }
    header {
        background: #232f3e;
        color: #fff;
        padding: 10px 0;
        text-align: center;
    }
    .container {
        width: 80%;
        margin: auto;
        overflow: hidden;
    }
    img {
        max-width: 100%;
        height: auto;
    }
    form {
        background: #fff;
        padding: 20px;
        margin-bottom: 20px;
        border-radius: 8px;
        box-shadow: 0 0 10px rgba(0,0,0,0.1);
    }
    h1, h2 {
        color: #333;
    }
    input[type="text"], input[type="file"] {
        padding: 10px;
        margin: 10px 0;
        border: 1px solid #ddd;
        border-radius: 4px;
        width: 100%;
    }
    button {
            background: #232f3e;
            color: #fff;
            border: none;
            padding: 10px 15px;
            border-radius: 4px;
            cursor: pointer;
        }
        button:hover {
            background: #1d2630;
        }
        ul {
            list-style: none;
            padding: 0;
        }
        li {
            background: #fff;
            margin: 10px 0;
            padding: 10px;
            border-radius: 4px;
            box-shadow: 0 0 5px rgba(0,0,0,0.1);
        }
        .file-actions {
            display: flex;
            gap: 10px;
        }
    </style>
</head>
<body>
    <header>
        <div class="container">
            <h1>S3 File Manager</h1>
        </div>
    </header>

    <div class="container">
        <!-- Create Bucket Form -->
        <h2>Create Bucket</h2>
        <form action="{{ url_for('create_bucket') }}" method="post">
            <input type="text" name="bucket_name" placeholder="Bucket Name" required>
            <button type="submit">Create Bucket</button>
        </form>
```

```
<!-- Delete Bucket Form -->
<h2>Delete Bucket</h2>
<form action="{{ url_for('delete_bucket') }}" method="post">
    <input type="text" name="bucket_name" placeholder="Bucket Name" required>
    <button type="submit">Delete Bucket</button>
</form>

<!-- List Buckets -->
<h2>Buckets</h2>
<ul>
    {% for bucket in bucket_names %}
        <li><a href="{{ url_for('bucket_contents', bucket_name=bucket) }}">{{ bucket }}</a></li>
    {% endfor %}
</ul>

<!-- List Files in Bucket -->
{% if bucket_name %}
    <h2>Files in {{ bucket_name }}</h2>
    <form action="{{ url_for('upload_file') }}" method="post" enctype="multipart/form-data">
        <input type="hidden" name="bucket_name" value="{{ bucket_name }}">
        <input type="file" name="file" required>
        <button type="submit">Upload File</button>
    </form>

    <form action="{{ url_for('create_folder') }}" method="post">
        <input type="hidden" name="bucket_name" value="{{ bucket_name }}">
        <input type="text" name="folder_name" placeholder="Folder Name" required>
        <button type="submit">Create Folder</button>
    </form>

    <form action="{{ url_for('delete_folder') }}" method="post">
        <input type="hidden" name="bucket_name" value="{{ bucket_name }}">
        <input type="text" name="folder_name" placeholder="Folder Name" required>
        <button type="submit">Delete Folder</button>
    </form>

    <form action="{{ url_for('delete_file') }}" method="post">
        <input type="hidden" name="bucket_name" value="{{ bucket_name }}">
        <input type="text" name="file_key" placeholder="File Key" required>
        <button type="submit">Delete File</button>
    </form>
```

```
            <h2>Files</h2>
            <ul>
                {% for file in files %}
                    <li>
                        {{ file.Key }}
                        <div class="file-actions">
                            <form action="{{ url_for('copy_file') }}" method="post">
                                <input type="hidden" name="source_bucket" value="{{ bucket_name }}">
                                <input type="hidden" name="file_key" value="{{ file.Key }}">
                                <input type="text" name="destination_bucket" placeholder="Destination
Bucket" required>
                                <button type="submit">Copy</button>
                            </form>
                            <form action="{{ url_for('move_file') }}" method="post">
                                <input type="hidden" name="source_bucket" value="{{ bucket_name }}">
                                <input type="hidden" name="file_key" value="{{ file.Key }}">
                                <input type="text" name="destination_bucket" placeholder="Destination
Bucket" required>
                                 <button type="submit">Move</button>
                            </form>
                        </div>
                    </li>
                {% endfor %}
            </ul>
        {% endif %}
    </div>
</body>
</html>
```

RESULTS SNAPSHOTS: Run the command python3 app.py

1.creating the bucket



127.0.0.1:5000

S3 File Manager

Create Bucket

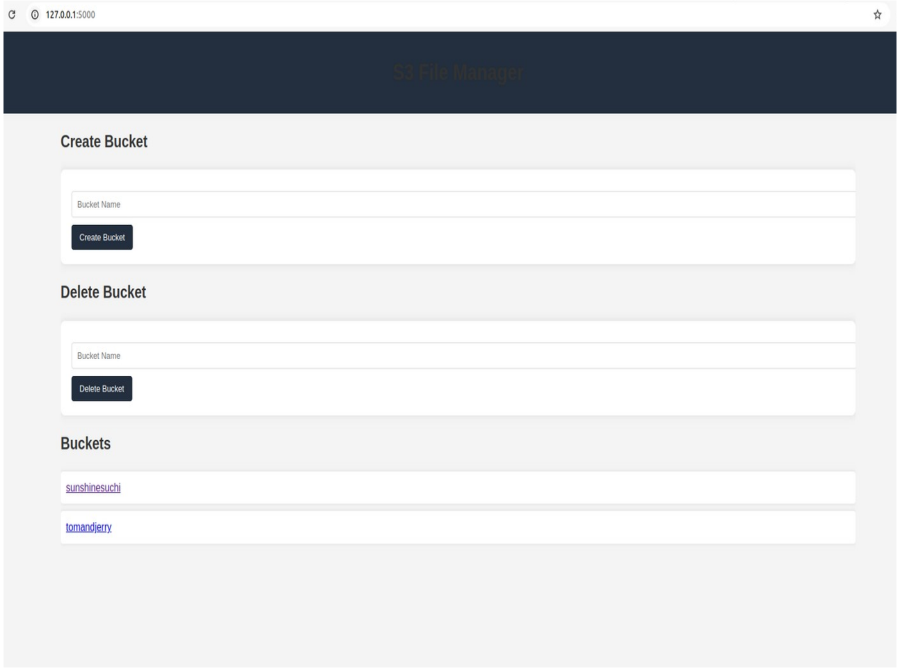
tomandjerry

Delete Bucket

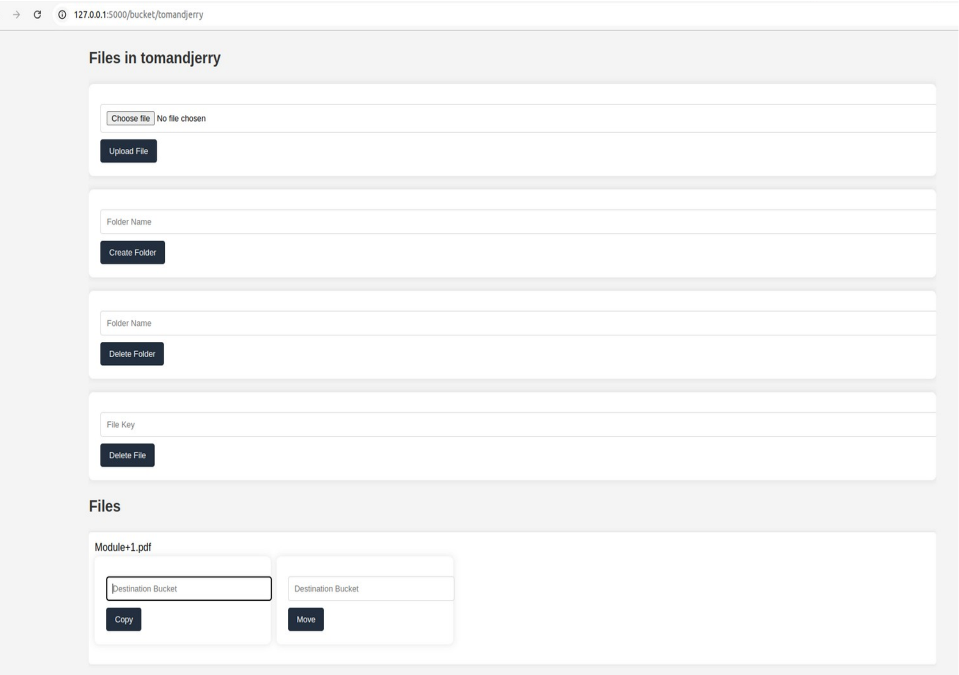
Buckets

sunshinesuchi

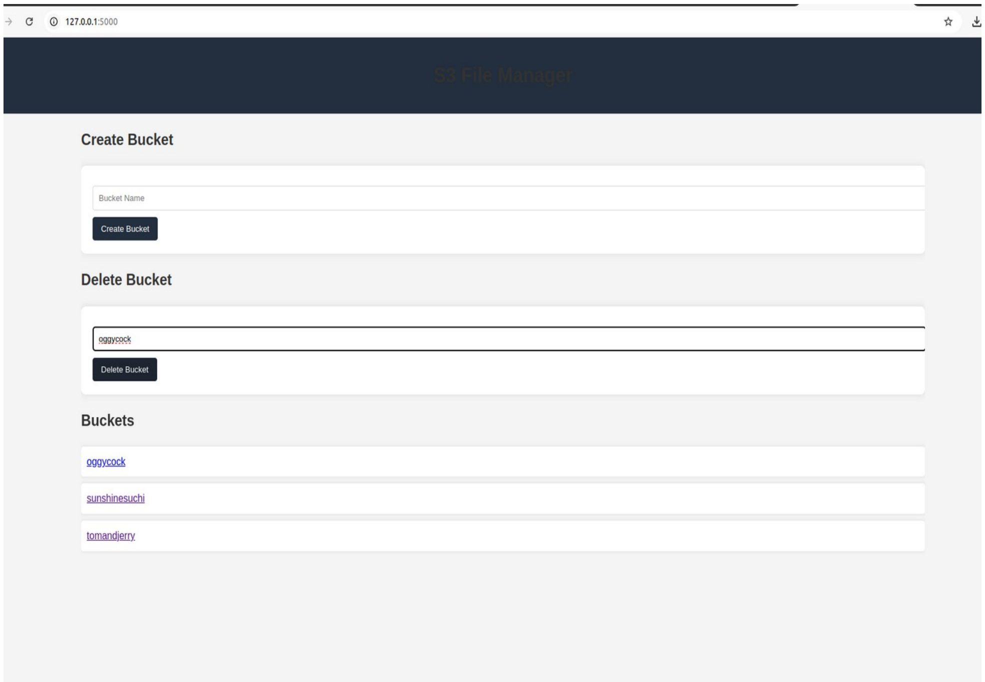
bucket created



uploading files



deleting the bucket


127.0.0.1:5000

S3 File Manager

Create Bucket

Delete Bucket

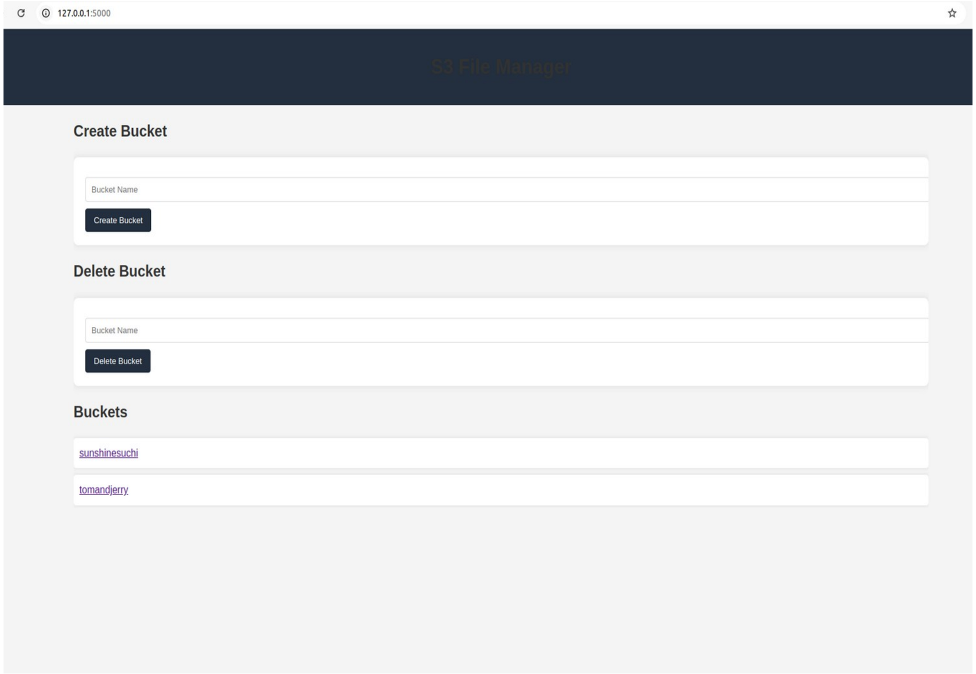
oggycock

Buckets

oggycock

sunshinesuchi

tomandjerry


127.0.0.1:5000

S3 File Manager

Create Bucket

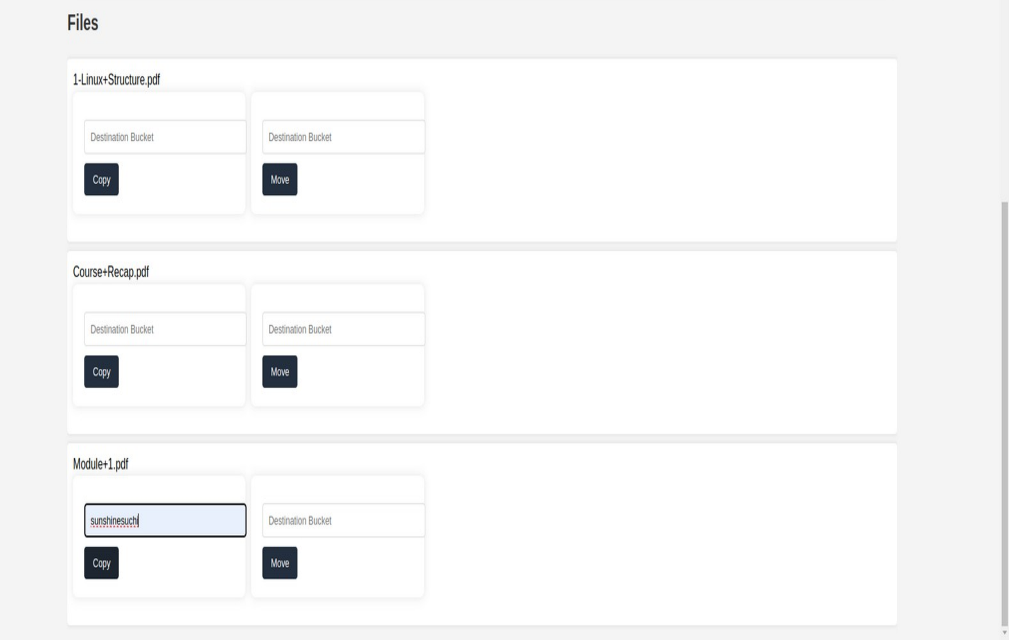
Delete Bucket

Buckets

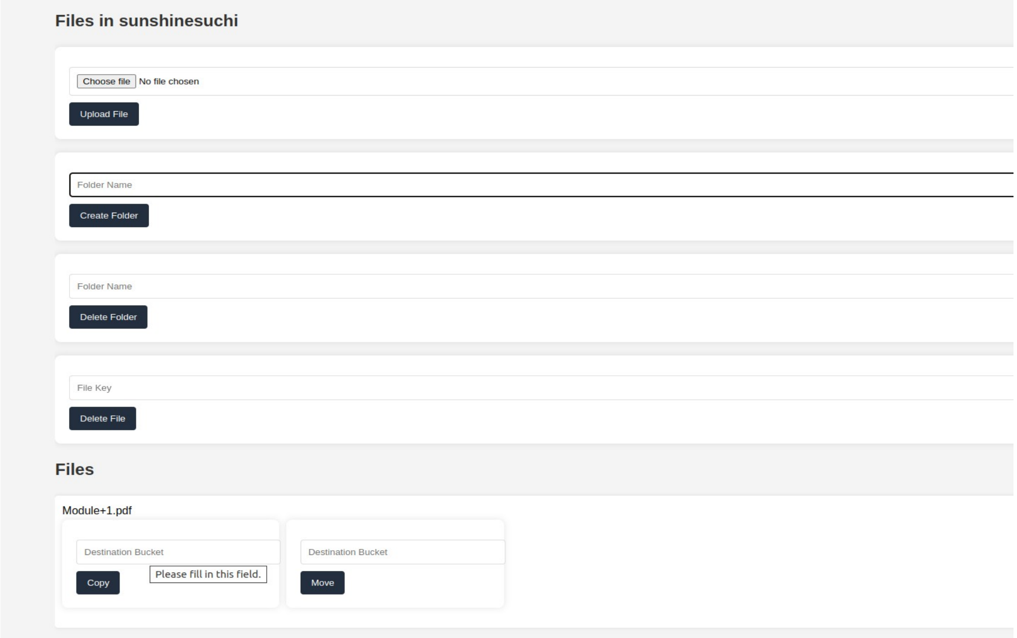
sunshinesuchi

tomandjerry

copy /move file within s3



sunshinesuchi



Please fill in this field.