Dynamic Web Application for Storing Files with Text and Documents on AWS

1. Objective:

The objective of this project is to create a dynamic web application that allows users to upload files (such as images, PDFs, etc.) along with text descriptions. The uploaded files will be stored in an AWS S3 bucket, while the metadata (file name, description, and file URL) will be stored in an Amazon RDS MySQL database. The application will be developed using PHP and hosted on an EC2 instance.

2. Prerequisites

- **AWS Account:** Sign up or log in to your AWS account.
- **AWS Services Required:**
  - **S3:** For storing files.
  - **RDS:** For storing metadata (file details, descriptions, etc.).
  - **EC2:** To host the PHP application.
- **Required Tools:**
  - AWS CLI or SDK for AWS configuration.
  - Composer for managing PHP dependencies.
- **Knowledge Required:**
  - Basic understanding of PHP.
  - Basic AWS knowledge (EC2, S3, RDS,IAM).

3.Project Setup Steps:

1. Set Up an AWS Account:

- Sign up for AWS at AWS website.
- Access S3, RDS, and EC2 services in your AWS console to begin setup.

2. Set Up AWS S3 Bucket:

1. Go to **S3** in the AWS console.
2. Create a new bucket, choose a unique name, and select the region.
3. Configure permissions to control access (private/public).
4. Enable versioning if you want to keep multiple versions of files.

3. Set Up AWS RDS MySQL Database:

1. Go to **RDS** and create a new MySQL database.
2. Select **MySQL** as the engine and configure the database parameters.
3. Set up security to allow EC2 access to the database.
4. Record the **RDS endpoint** URL for connecting to the database from the application.

4. Launch EC2 Instance:

1. Go to **EC2** and launch an instance (Amazon Linux 2 or Ubuntu).

2. Choose the **t2.micro** instance type (Free Tier eligible).
3. Configure security rules to allow HTTP traffic on port 80.
4. Note the **public IP address** of your EC2 instance.

5.Install LAMP Stack on EC2:

1. **SSH** into the EC2 instance.
2. Install Apache, PHP, and MySQL client:
   - sudo yum update -y
   - sudo yum install httpd php php-mysqlnd php-fpm -y
3. Start Apache and enable it to run on boot:
   - sudo systemctl start httpd
   - sudo systemctl enable httpd

6. Install AWS SDK for PHP:

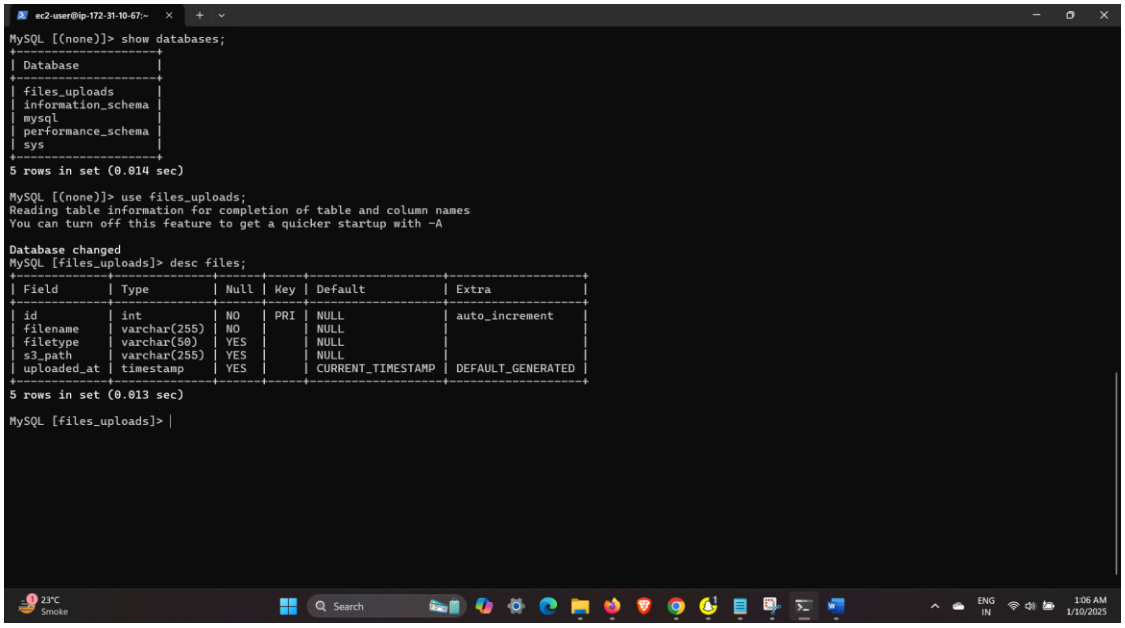
1. **Install Composer** (PHP dependency manager):
   - curl -sS https://getcomposer.org/installer | php
   - sudo mv composer.phar /usr/local/bin/composer
2. Install AWS SDK for PHP:
   - composer require aws/aws-sdk-php

7. Create the PHP Application:

> Create Database Files_uploads

> create Table files

Create a table in MySQL to store file metadata like this-

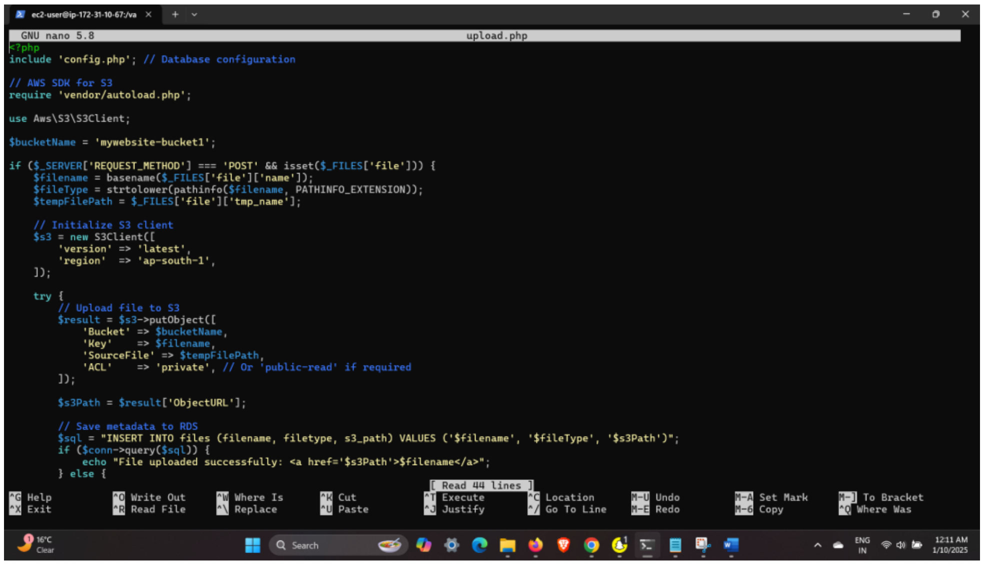


```
MySQL [(none)]> show databases;
+--------------------+
| Database           |
+--------------------+
| files_uploads      |
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
5 rows in set (0.014 sec)

MySQL [(none)]> use files_uploads;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
MySQL [files_uploads]> desc files;
+-------------+--------------+------+-----+-------------------+-------------------+
| Field       | Type         | Null | Key | Default           | Extra             |
+-------------+--------------+------+-----+-------------------+-------------------+
| id          | int          | NO   | PRI | NULL              | auto_increment    |
| filename    | varchar(255) | NO   |     | NULL              |                   |
| filetype    | varchar(50)  | YES  |     | NULL              |                   |
| s3_path     | varchar(255) | YES  |     | NULL              |                   |
| uploaded_at | timestamp    | YES  |     | CURRENT_TIMESTAMP | DEFAULT_GENERATED |
+-------------+--------------+------+-----+-------------------+-------------------+
5 rows in set (0.013 sec)

MySQL [files_uploads]> |
```

8. Create File Upload Form:

> Cd /var/www/html

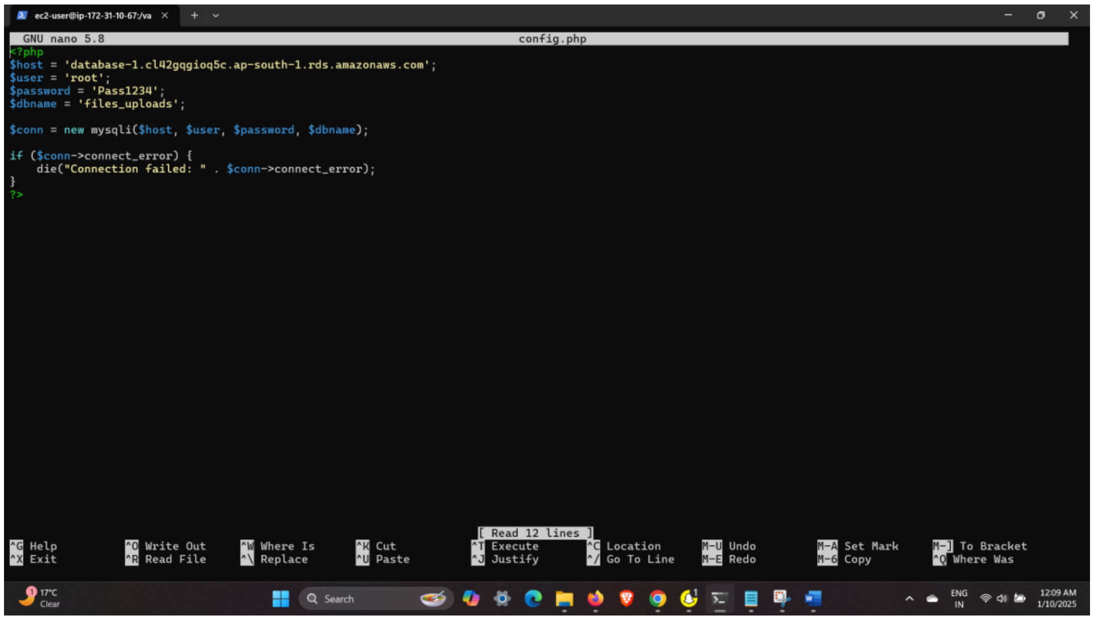
> sudo nano upload.php



```
GNU nano 5.8                                upload.php
<?php
include 'config.php'; // Database configuration

// AWS SDK for S3
require 'vendor/autoload.php';

use Aws\S3\S3Client;

$bucketName = 'mywebsite-bucket1';

if ($_SERVER['REQUEST_METHOD'] === 'POST' && isset($_FILES['file'])) {
    $filename = basename($_FILES['file']['name']);
    $fileType = strtolower(pathinfo($filename, PATHINFO_EXTENSION));
    $tempFilePath = $_FILES['file']['tmp_name'];

    // Initialize S3 client
    $s3 = new S3Client([
        'version' => 'latest',
        'region'  => 'ap-south-1',
    ]);

    try {
        // Upload file to S3
        $result = $s3->putObject([
            'Bucket' => $bucketName,
            'Key'    => $filename,
            'SourceFile' => $tempFilePath,
            'ACL'    => 'private', // Or 'public-read' if required
        ]);

        $s3Path = $result['ObjectURL'];

        // Save metadata to RDS
        $sql = "INSERT INTO files (filename, filetype, s3_path) VALUES ('$filename', '$fileType', '$s3Path')";
        if ($conn->query($sql)) {
            echo "File uploaded successfully: <a href='$s3Path'>$filename</a>";
        } else {
                                  [ Read 44 lines ]
^G Help      ^O Write Out  ^W Where Is  ^K Cut    ^T Execute  ^C Location    M-U Undo  M-A Set Mark  M-] To Bracket
^X Exit      ^R Read File  ^\ Replace   ^U Paste  ^J Justify  ^/ Go To Line  M-E Redo  M-6 Copy      ^Q Where Was
```

9.Make Config.php For Configurition

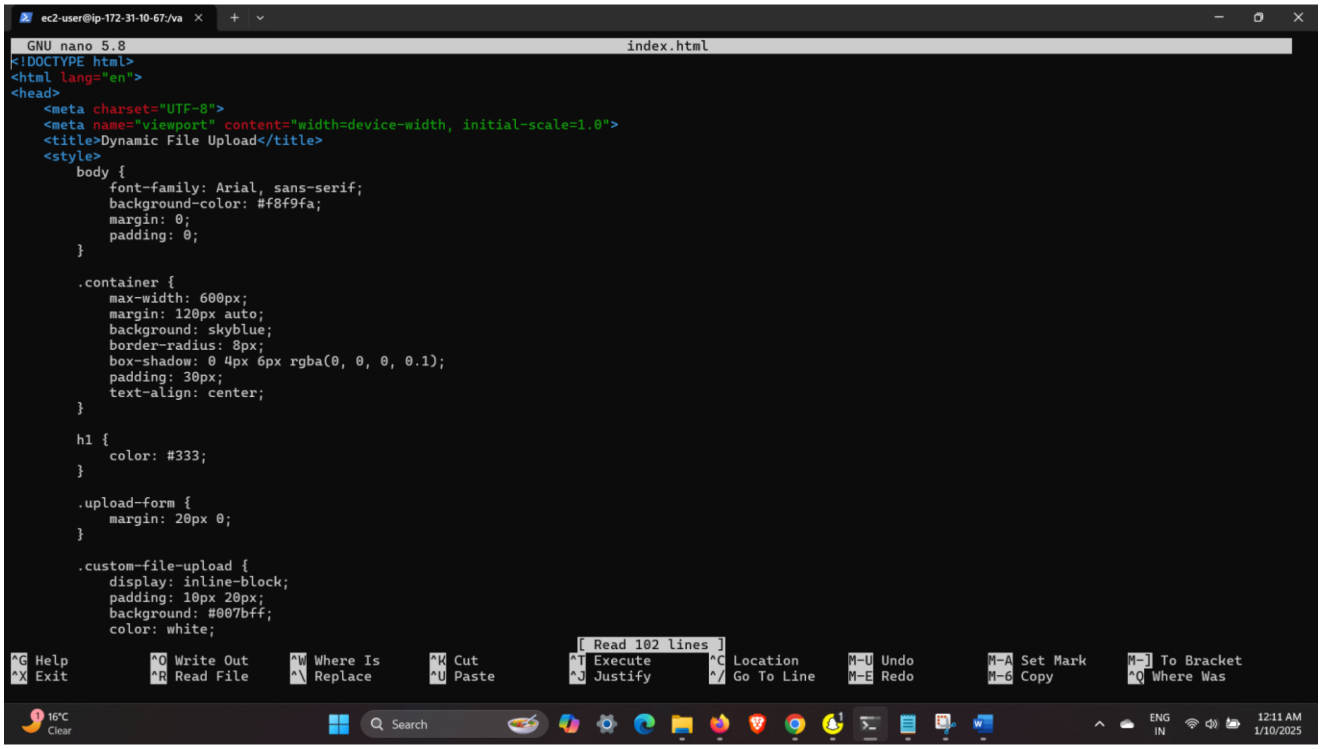
>sudo nano config.php



```
GNU nano 5.8                                config.php
<?php
$host = 'database-1.cl42gqgioq5c.ap-south-1.rds.amazonaws.com';
$user = 'root';
$password = 'Pass1234';
$dbname = 'files_uploads';

$conn = new mysqli($host, $user, $password, $dbname);

if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
?>

                                  [ Read 12 lines ]
^G Help      ^O Write Out  ^W Where Is  ^K Cut    ^T Execute  ^C Location    M-U Undo  M-A Set Mark  M-] To Bracket
^X Exit      ^R Read File  ^\ Replace   ^U Paste  ^J Justify  ^/ Go To Line  M-E Redo  M-6 Copy      ^Q Where Was
```

10.Create file index.html

> sudo nano index.html



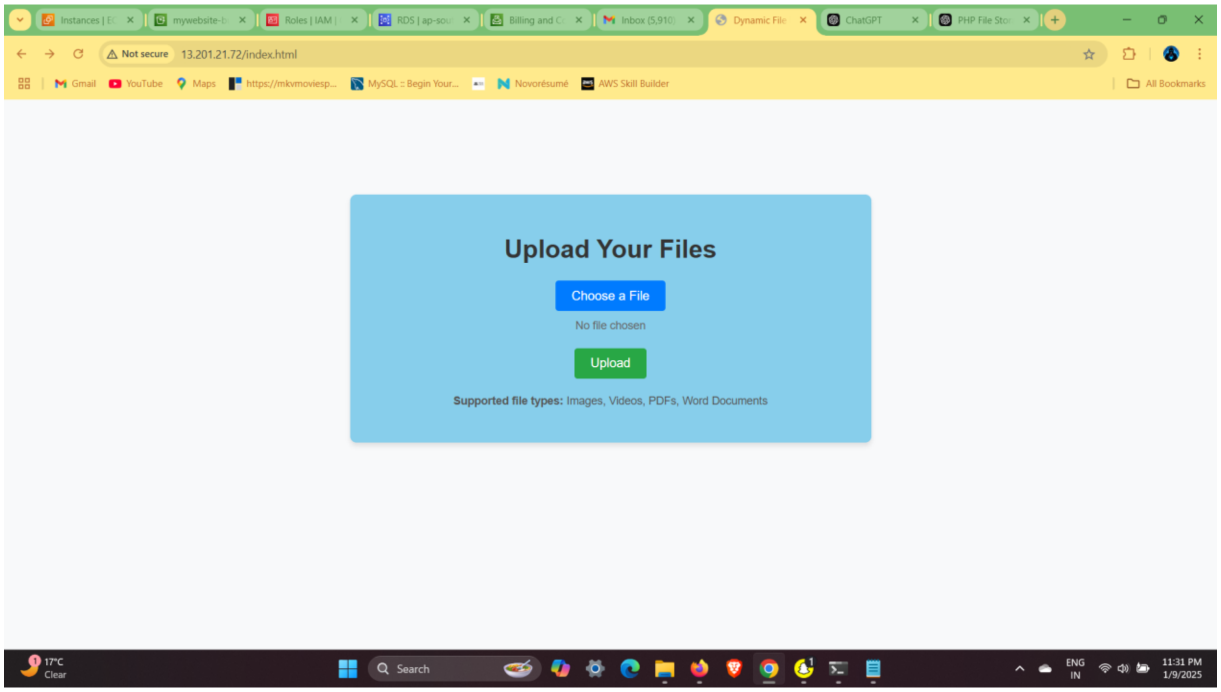
```
  GNU nano 5.8                                   index.html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Dynamic File Upload</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            background-color: #f8f9fa;
            margin: 0;
            padding: 0;
        }

        .container {
            max-width: 600px;
            margin: 120px auto;
            background: skyblue;
            border-radius: 8px;
            box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);
            padding: 30px;
            text-align: center;
        }

        h1 {
            color: #333;
        }

        .upload-form {
            margin: 20px 0;
        }

        .custom-file-upload {
            display: inline-block;
            padding: 10px 20px;
            background: #007bff;
            color: white;
                                   [ Read 102 lines ]
^G Help        ^O Write Out   ^W Where Is    ^K Cut         ^T Execute     ^C Location    M-U Undo       M-A Set Mark   M-] To Bracket
^X Exit        ^R Read File   ^\ Replace     ^U Paste       ^J Justify     ^/ Go To Line  M-E Redo       M-6 Copy       ^Q Where Was
```

11.Test the Application:

- Open the EC2 public IP in a browser.

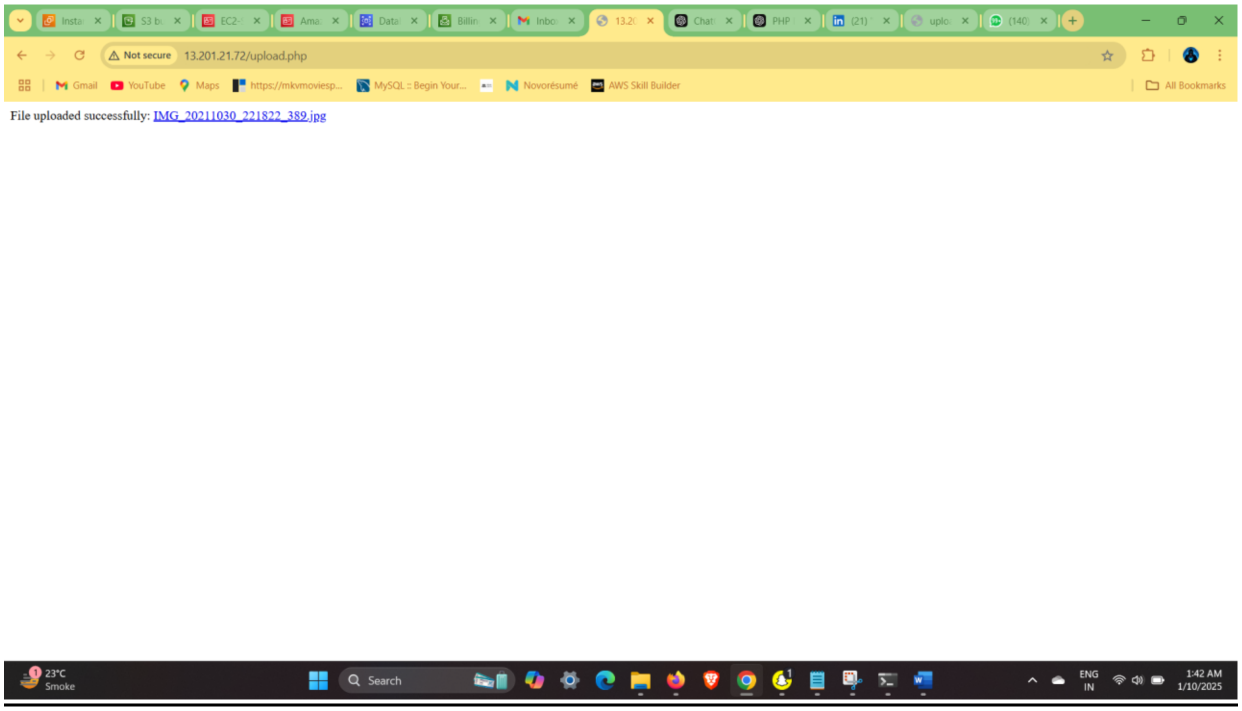
- Use the form to upload a file and description.

✓ Final output of index.html



Supported file types: Images, Videos, PDFs, Word Documents

✓ Redirecting To PHP Page

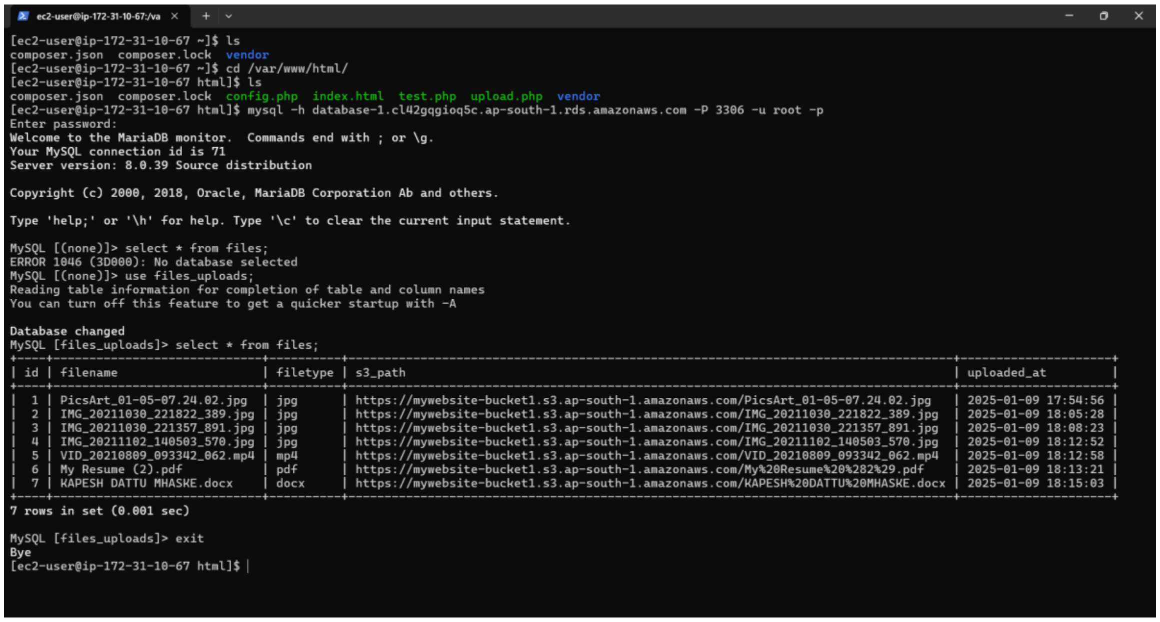


13.201.21.72/upload.php

File uploaded successfully: IMG_20211030_221822_389.jpg

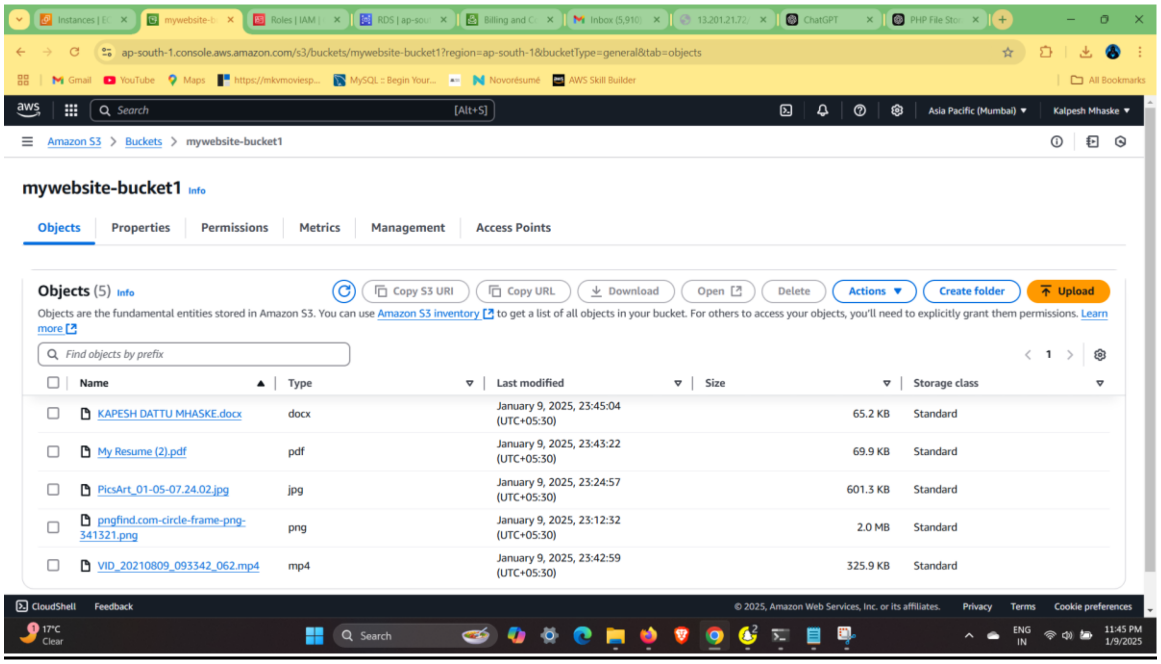
✓ Text & S3_Path storing in RDS MySQL database

➢ SELECT * FROM files;



```
[ec2-user@ip-172-31-10-67 ~]$ ls
composer.json  composer.lock  vendor
[ec2-user@ip-172-31-10-67 ~]$ cd /var/www/html/
[ec2-user@ip-172-31-10-67 html]$ ls
composer.json  composer.lock  config.php  index.html  test.php  upload.php  vendor
[ec2-user@ip-172-31-10-67 html]$ mysql -h database-1.cl42gqgioq5c.ap-south-1.rds.amazonaws.com -P 3306 -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 71
Server version: 8.0.39 Source distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [(none)]> select * from files;
ERROR 1046 (3D000): No database selected
MySQL [(none)]> use files_uploads;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
MySQL [files_uploads]> select * from files;
+----+-----------------------------+----------+------------------------------------------------------------------------------------+---------------------+
| id | filename                    | filetype | s3_path                                                                            | uploaded_at         |
+----+-----------------------------+----------+------------------------------------------------------------------------------------+---------------------+
|  1 | PicsArt_01-05-07.24.02.jpg  | jpg      | https://mywebsite-bucket1.s3.ap-south-1.amazonaws.com/PicsArt_01-05-07.24.02.jpg   | 2025-01-09 17:54:56 |
|  2 | IMG_20211030_221822_389.jpg | jpg      | https://mywebsite-bucket1.s3.ap-south-1.amazonaws.com/IMG_20211030_221822_389.jpg  | 2025-01-09 18:05:28 |
|  3 | IMG_20211030_221357_891.jpg | jpg      | https://mywebsite-bucket1.s3.ap-south-1.amazonaws.com/IMG_20211030_221357_891.jpg  | 2025-01-09 18:08:23 |
|  4 | IMG_20211102_140503_570.jpg | jpg      | https://mywebsite-bucket1.s3.ap-south-1.amazonaws.com/IMG_20211102_140503_570.jpg  | 2025-01-09 18:12:52 |
|  5 | VID_20210809_093342_062.mp4 | mp4      | https://mywebsite-bucket1.s3.ap-south-1.amazonaws.com/VID_20210809_093342_062.mp4  | 2025-01-09 18:12:58 |
|  6 | My Resume (2).pdf           | pdf      | https://mywebsite-bucket1.s3.ap-south-1.amazonaws.com/My%20Resume%20%282%29.pdf    | 2025-01-09 18:13:21 |
|  7 | KAPESH DATTU MHASKE.docx    | docx     | https://mywebsite-bucket1.s3.ap-south-1.amazonaws.com/KAPESH%20DATTU%20MHASKE.docx | 2025-01-09 18:15:03 |
+----+-----------------------------+----------+------------------------------------------------------------------------------------+---------------------+
7 rows in set (0.001 sec)

MySQL [files_uploads]> exit
Bye
[ec2-user@ip-172-31-10-67 html]$ |
```

✓ Files Stored On S3



ap-south-1.console.aws.amazon.com/s3/buckets/mywebsite-bucket1?region=ap-south-1&bucketType=general&tab=objects

Amazon S3 > Buckets > mywebsite-bucket1

mywebsite-bucket1 Info

Objects | Properties | Permissions | Metrics | Management | Access Points

Objects (5) Info

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

Name	Type	Last modified	Size	Storage class
KAPESH DATTU MHASKE.docx	docx	January 9, 2025, 23:45:04 (UTC+05:30)	65.2 KB	Standard
My Resume (2).pdf	pdf	January 9, 2025, 23:43:22 (UTC+05:30)	69.9 KB	Standard
PicsArt_01-05-07.24.02.jpg	jpg	January 9, 2025, 23:24:57 (UTC+05:30)	601.3 KB	Standard
pngfind.com-circle-frame-png-341321.png	png	January 9, 2025, 23:12:32 (UTC+05:30)	2.0 MB	Standard
VID_20210809_093342_062.mp4	mp4	January 9, 2025, 23:42:59 (UTC+05:30)	325.9 KB	Standard

✓ **Final Adjustments and Security:**

- Ensure that EC2 security group allows HTTP traffic (port 80).
- Implement user authentication and access control to secure file uploads.
- Use encryption (both at rest and in transit) for enhanced security.

✓ Conclusion:

This project demonstrates the creation of a dynamic web application to store files and text descriptions on AWS. By using AWS services such as S3 for file storage and RDS for database management, this solution provides a scalable and secure way to store and manage user-uploaded content.

BY -KALPESH MHASKE