**MODULE 10**

**A)Develop a PHP web Application which demonstrates how to upload image into database and display it.**

Uploading the image/videos into the database and displaying it using PHP is the way of uploading the image into the database and fetching it from the database. Using the PHP code, the user uploads the image or videos they are safely getting entry into the database and the images should be saved into a particular location by fetching these images from the database.  
If any of the websites contain the functionality to upload images/videos with some detail, then by using this code we will upload the image into your database and whether you would like to ascertain what the person has got to be uploaded. And by this code the image which is uploaded that where save in your system where you are given the location.

**Approach:** Make sure you have [XAMPP](https://www.geeksforgeeks.org/how-to-install-xampp-on-windows/) or [WAMP](https://www.geeksforgeeks.org/how-to-install-and-set-up-a-wamp-server/) server installed on your machine. In this tutorial, we will be using the WAMP server.

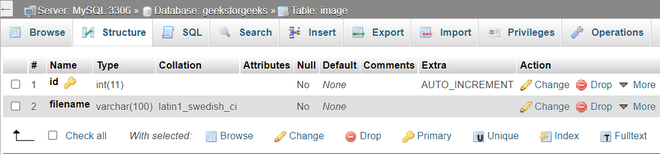
**1. Create Database:**First, we will create a database named ‘**geeksforgeeks**‘. You can use your existing database or create a new one.



**2. Create Table**: Create a table named ‘**image**‘. The table contains two fields:

* id – int(11)
* filename – varchar(100)

The id should be in **Auto incremented(AI)**. Your table structure should look like this:



**Program:** Now, we will create an [HTML form](https://www.geeksforgeeks.org/html-forms/) for uploading image files (you can upload any type of file like .pdf or .mp4) and will display the uploaded image.

* **HTML code:**

**<!DOCTYPE html>**

**<html>**

**<head>**

**<title>Image Upload</title>**

**<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.1/dist/css/bootstrap.min.css">**

**<link rel="stylesheet" type="text/css" href="style.css" />**

**</head>**

**<body>**

**<div id="content">**

**<form method="POST" action="" enctype="multipart/form-data">**

**<div class="form-group">**

**<input class="form-control" type="file" name="uploadfile" value="" />**

**</div>**

**<div class="form-group">**

**<button class="btn btn-primary" type="submit" name="upload">UPLOAD</button>**

**</div>**

**</form>**

**</div>**

**<div id="display-image">**

**<?php**

**$query = " select \* from image ";**

**$result = mysqli\_query($db, $query);**

**while ($data = mysqli\_fetch\_assoc($result)) {**

**?>**

**<img src="./image/<?php echo $data['filename']; ?>">**

**<?php**

**}**

**?>**

**</div>**

**</body>**

**</html>**

* **CSS code:** The **style.css** is the file that styles the form into a new design and the code is given below.

**\*{**

**margin: 0;**

**padding: 0;**

**box-sizing: border-box;**

**}**

**#content{**

**width: 50%;**

**justify-content: center;**

**align-items: center;**

**margin: 20px auto;**

**border: 1px solid #cbcbcb;**

**}**

**form{**

**width: 50%;**

**margin: 20px auto;**

**}**

**#display-image{**

**width: 100%;**

**justify-content: center;**

**padding: 5px;**

**margin: 15px;**

**}**

**img{**

**margin: 5px;**

**width: 350px;**

**height: 250px;**

**}**

**PHP code:** The PHP code is for the uploading images, the file name is saved with the **index.php**, you can also save it with another name as you prefer.

**<?php**

**error\_reporting(0);**

**$msg = "";**

**// If upload button is clicked ...**

**if (isset($\_POST['upload'])) {**

**$filename = $\_FILES["uploadfile"]["name"];**

**$tempname = $\_FILES["uploadfile"]["tmp\_name"];**

**$folder = "./image/" . $filename;**

**$db = mysqli\_connect("localhost", "root", "", "geeksforgeeks");**

**// Get all the submitted data from the form**

**$sql = "INSERT INTO image (filename) VALUES ('$filename')";**

**// Execute query**

**mysqli\_query($db, $sql);**

**// Now let's move the uploaded image into the folder: image**

**if (move\_uploaded\_file($tempname, $folder)) {**

**echo "<h3> Image uploaded successfully!</h3>";**

**} else {**

**echo "<h3> Failed to upload image!</h3>";**

**}**

**}**

**?>**

**b)Develop a Php web Application which demonstrates the process of encode and decode JSON by the help of json\_encode () and json\_decode () functions.**

[JSON](https://www.geeksforgeeks.org/javascript-json/) stands for JavaScript Object Notation. Like XML, it is a text-based format for the exchange of data which is easier to read and write and it is lighter than other formats. JSON is based on two basic structures namely [Objects](https://www.geeksforgeeks.org/objects-in-javascript/) and[Arrays.](https://www.geeksforgeeks.org/arrays-in-javascript/)

**Parsing JSON data in PHP:**There are built-in functions in PHP for both encoding and decoding JSON data. These functions are **[json\_encode()](https://www.geeksforgeeks.org/php-json_encode-function/)** and **[json\_decode()](https://www.geeksforgeeks.org/php-json_decode-function/)**. These functions works only with UTF-8 encoded string.

**Decoding JSON data in PHP:**It is very easy to decode JSON data in PHP. You just have to use **json\_decode()**function to convert JSON objects to the appropriate PHP data type.

**Example:**By default the**json\_decode()** function returns an object. You can optionally specify a second parameter that accepts a boolean value. When it is set as “true”, [JSON objects](https://www.geeksforgeeks.org/javascript-json-objects/) are decoded into[associative arrays](https://www.geeksforgeeks.org/associative-arrays-in-php/).

**PHP**

**<?php**

**$student\_data = '{"Ram":96,"Prashant":76,"Varun":65,"Mark":34}';**

**// Decoding above JSON String into JSON object**

**$decoded = json\_decode($student\_data);**

**// Dump the $decoded variable**

**var\_dump($decoded);**

**?>**

**Encoding JSON data in PHP:**Encoding JSON data is as easy as encoding JSON data in PHP. We use the**[json\_encode()](https://www.geeksforgeeks.org/php-json_encode-function/)**function, the data being encoded can be any PHP data type except a resource like a filehandle.

**Example 1:**The following code demonstrates encoding PHP associative array.

**PHP**

**<?php**

**// PHP associative array**

**$student\_data = array(**

**"Ram"=>96,**

**"Prashant"=>76,**

**"Varun"=>65,**

**"Mark"=>34**

**);**

**// Encoding PHP Associative array using json\_encode()**

**$encoded = json\_encode($student\_data);**

**// Echo the data**

**echo $encoded;**

**?>**

**Output:**

{"Ram":96,"Prashant":76,"Varun":65,"Mark":34}