

# PHP Practical Summary

---

## Agenda Covered:

- Predefined Variables
- File System (Create/Delete Folders & Files)
- File Handling (Open, Read, Write, Close)
- Form Handling
- Global Variables in PHP
- Dealing with JSON

## 1. Handling Forms in PHP

register.php contains an HTML form that sends user data to server.php using the POST method.

### Main Form Fields:

- name: text input
- email: email input
- password: password input
- image: file input (profile image)

### Example:

```
<form action="server.php?message=defgh" method="POST"
enctype="multipart/form-data" class="row w-75 m-auto">
  <input name="name" type="text" class="form-control">
  <input name="email" type="email" class="form-control">
  <input name="password" type="password" class="form-control">
  <input name="image" type="file" class="form-control">
  <button class="btn btn-primary">submit</button>
</form>
```

### Error Display:

```
<?php
if (isset($_GET['error'])) {
    echo "<p class='alert alert-danger'>" . $_GET['error'] . "</p>";
}
?>
```

## 2. Global Variables in PHP

PHP provides superglobal arrays like `$_GET`, `$_POST`, `$_FILES`, `$_SERVER`, `$_SESSION`, etc.

### Examples:

```
$_POST['name'] // Gets form input value with name 'name'
$_FILES['image'] // Handles uploaded file data
```

## 3. Handling File Uploads (server.php)

### Validating Image Type:

```
$allowedTypes = ['image/jpeg', 'image/png', 'image/jpg', 'image/gif',
'image/webp'];
if (!in_array($image['type'], $allowedTypes)) {
    header("Location: register.php?error=You can upload images only");
    exit();
}
```

### Saving the Image:

```
$imgNewName = uniqid() . $image['name'];
move_uploaded_file($image['tmp_name'], "../usersProfile/" .
$imgNewName);
```

### Writing to JSON:

```
$data = file_get_contents("database.json");
$dataArray = json_decode($data, true);
$dataArray[] = ['name' => $name, 'email' => $email, 'image' =>
$imgNewName];
file_put_contents("database.json", json_encode($dataArray));
```

## 4. File System in PHP

### Create a new folder:

```
mkdir("myFolder");
```

### Delete a folder:

```
rmdir("myFolder");
```

### Create a file:

```
fopen("example.txt", "w");
```

### Delete a file:

```
unlink("example.txt");
```

## 5. File Handling in PHP

### Open file:

```
$file = fopen("example.txt", "w");
```

#### Write to file:

```
fwrite($file, "Hello World");
```

#### Close file:

```
fclose($file);
```

#### Read from file:

```
$content = file_get_contents("example.txt");
```

## 6. Dealing with JSON in PHP

#### Read JSON file:

```
$json = file_get_contents("database.json");  
$data = json_decode($json, true);
```

#### Write JSON file:

```
$data[] = ['name' => 'Ali', 'email' => 'ali@gmail.com'];  
file_put_contents("database.json", json_encode($data));
```

## 7. Displaying User Data (users\_data.php)

#### Reading and displaying:

```
$data = file_get_contents('database.json');  
$usersData = json_decode($data, true);
```

#### Table display:

```
foreach ($usersData as $user) {  
    echo "<tr>";  
    echo "<td>" . $user['name'] . "</td>";  
    echo "<td>" . $user['email'] . "</td>";  
    echo "<td><img width='100px' src='./usersProfile/" . $user['image'] .  
    "'></td>";  
    echo "</tr>";  
}
```

#### Notes:

- Always check uploaded file type before moving it.
- Use json\_decode with true to get an associative array.
- Handle errors using query parameters (\$\_GET['error']).
- Use uniqid() to avoid image name collision.