

 **EXP. 1 — Web Designing in HTML**
Simple

1. Create a basic HTML webpage displaying your college information using headings, paragraphs, and line breaks.

```
<!DOCTYPE html>
<html>
<head>
  <title>My College Information</title>
</head>
<body>
  <h1>ABC College of Engineering</h1>
  <h2>About Our College</h2>
  <p>
    ABC College is one of the top institutes providing quality education.<br>
    Established in 1998, located in Pune.<br>
    NAAC Accredited.
  </p>
  <h3>Courses Offered</h3>
  <p>B.Tech | M.Tech | Diploma</p>
</body>
</html>
```

2. Design an HTML page to display a list of courses using ordered and unordered lists.

```
<!DOCTYPE html>
<html>
<head><title>Courses</title></head>
<body>
  <h2>Our Courses</h2>

  <h3>Engineering Branches (Ordered List)</h3>
  <ol>
    <li>Computer Engineering</li>
    <li>IT Engineering</li>
    <li>Mechanical Engineering</li>
    <li>Civil Engineering</li>
  </ol>

  <h3>Diploma Courses (Unordered List)</h3>
  <ul>
    <li>Computer Engineering</li>
    <li>Mechanical Engineering</li>
  </ul>
</body>
</html>
```

Application Oriented

3. Develop an HTML webpage for a small shop including product details using tables and images.

```
<!DOCTYPE html>
<html>
<head><title>Shop Products</title></head>
<body>

  <h1>Mobile Shop</h1>

  <table border="1" cellpadding="10">
    <tr>
```

```
<th>Image</th>
<th>Product Name</th>
<th>Price</th>
</tr>

<tr>
<td></td>
<td>Samsung A52</td>
<td>₹22,000</td>
</tr>

<tr>
<td></td>
<td>Redmi Note 12</td>
<td>₹15,000</td>
</tr>
</table>

</body>
</html>
```

4. Create a multi-section webpage for a travel agency including navigation links, images, and contact details.

```
<!DOCTYPE html>
<html>
<head>
<title>Travel Agency</title>
</head>
<body>

<h1>Dream Travel Agency</h1>

<a href="#home">Home</a> |
<a href="#places">Places</a> |
<a href="#contact">Contact</a>

<hr>

<section id="home">
<h2>Welcome</h2>
<p>We provide best tour packages around the world.</p>

</section>

<section id="places">
<h2>Popular Destinations</h2>
<ul>
<li>Manali</li>
<li>Goa</li>
<li>Kashmir</li>
</ul>
</section>

<section id="contact">
<h2>Contact Us</h2>
<p>Email: info@dreamtravel.com</p>
<p>Phone: +91 9876543210</p>
</section>
```

```
</body>
</html>
```

✅ EXP. 2 — HTML + CSS

Simple

1. Create an HTML page & apply inline, internal & external CSS.

index.html

```
<!DOCTYPE html>
<html>
<head>
<title>CSS Example</title>

<!-- Internal CSS -->
<style>
  h1 { color: blue; }
  p { font-size: 18px; }
</style>

<link rel="stylesheet" href="style.css"> <!-- External CSS -->
</head>
<body>

<h1>CSS Demonstration</h1>

<p style="color: red;">This is inline CSS.</p>

<p>This paragraph uses internal CSS.</p>

<p class="external">This text uses external CSS.</p>

</body>
</html>
```

style.css (External CSS)

```
.external {
  color: green;
  font-weight: bold;
}
```

2. Display student details and style it using CSS borders, padding, colors.

```
<!DOCTYPE html>
<html>
<head>
<style>
.box {
  border: 2px solid black;
  padding: 20px;
  background: lightyellow;
  width: 300px;
}
</style>
</head>
<body>
```

```
<div class="box">
  <h2>Student Details</h2>
  <p>Name: Amar More</p>
  <p>Course: B.Tech CSE</p>
</div>

</body>
</html>
```

Application Oriented

3. Build a responsive landing page using Flexbox.

```
<!DOCTYPE html>
<html>
<head>
<style>
.container {
  display: flex;
  justify-content: space-around;
  padding: 20px;
}
.box {
  width: 30%;
  padding: 20px;
  background: lightblue;
}
</style>
</head>
<body>

<h1>Mobile Store Landing Page</h1>

<div class="container">
  <div class="box">Phone 1</div>
  <div class="box">Phone 2</div>
  <div class="box">Phone 3</div>
</div>

</body>
</html>
```

4. Create a portfolio webpage with navigation bar, cards, footer using CSS.

```
<!DOCTYPE html>
<html>
<head>
<style>
nav { background:black; padding:10px; }
nav a { color:white; margin:10px; text-decoration:none; }

.card {
  border:1px solid gray;
  padding:20px;
  width:250px;
  margin:20px;
  display:inline-block;
}
}
```

```

    footer { background:black; color:white; padding:10px; text-align:center; }
</style>
</head>
<body>

<nav>
    <a href="#">Home</a>
    <a href="#">Projects</a>
    <a href="#">Contact</a>
</nav>

<div class="card">
    <h3>Project 1</h3>
    <p>Details here</p>
</div>

<div class="card">
    <h3>Project 2</h3>
    <p>Details here</p>
</div>

<footer>
    © 2025 My Portfolio
</footer>

</body>
</html>

```

EXP. 3 — JavaScript Functions & Validations **Simple**

1. Program to calculate square, cube, factorial.

```

<script>
function calculate() {
    let n = parseInt(document.getElementById("num").value);

    document.getElementById("sq").innerHTML = n*n;
    document.getElementById("cu").innerHTML = n*n*n;

    let f=1;
    for(let i=1;i<=n;i++) f*=i;

    document.getElementById("fact").innerHTML = f;
}
</script>

```

```

<input id="num" type="number">
<button onclick="calculate()">Calculate</button>

<p>Square: <span id="sq"></span></p>
<p>Cube: <span id="cu"></span></p>
<p>Factorial: <span id="fact"></span></p>

```

2. Validation to check if text field is empty

```

<script>
function validate() {
    let t = document.getElementById("txt").value;

```

```

    if(t==""){
        alert("Field cannot be empty");
        return false;
    }
}
</script>

<input type="text" id="txt">
<button onclick="validate()">Submit</button>

```

Application Oriented

3. Registration form with JS validation

```

<script>
function myValidate(){
    let email=document.getElementById("email").value;
    let mob=document.getElementById("mob").value;
    let age=document.getElementById("age").value;

    if(email==" " || !email.includes("@")){
        alert("Enter valid email");
        return false;
    }
    if(mob.length!=10){
        alert("Enter 10 digit mobile");
        return false;
    }
    if(age<1){
        alert("Enter valid age");
        return false;
    }
}
</script>

<form onsubmit="return myValidate()">
    Email: <input id="email"><br>
    Mobile: <input id="mob"><br>
    Age: <input id="age" type="number"><br>
    <button>Submit</button>
</form>

```

4. Mini calculator using JS functions

```

<script>
function calc(op){
    let a=parseInt(document.getElementById("a").value);
    let b=parseInt(document.getElementById("b").value);

    if(op=='+') alert(a+b);
    if(op=='-') alert(a-b);
    if(op=='*') alert(a*b);
    if(op=='/') alert(a/b);
}
</script>

<input id="a"><input id="b"><br>
<button onclick="calc('+)">+</button>
<button onclick="calc('-)">-</button>

```

```
<button onclick="calc('*')">*</button>
<button onclick="calc('/')">/</button>
```

✅ EXP. 4 — PHP Variables, Expressions, Arrays Simple

1. PHP script showing variables & arithmetic operations

```
<?php
$a = 10;
$b = 5;

echo "Sum = ".$a+$b."<br>";
echo "Difference = ".$a-$b."<br>";
echo "Product = ".$a*$b."<br>";
echo "Division = ".$a/$b;
?>
```

2. PHP program to store 5 student names & display using loops

```
<?php
$students = ["Amar", "Raj", "Rohan", "Sanjay", "Pooja"];

foreach($students as $s){
    echo $s."<br>";
}
?>
```

Application Oriented

3. Develop a PHP program that calculates the electricity bill using conditional statements.

```
<?php
$units = 120;
$bill = 0;

if($units <= 50){
    $bill = $units * 3.50;
}
elseif($units <= 150){
    $bill = (50 * 3.50) + (($units - 50) * 4.00);
}
elseif($units <= 250){
    $bill = (50 * 3.50) + (100 * 4.00) + (($units - 150) * 5.20);
}
else{
    $bill = (50 * 3.50) + (100 * 4.00) + (100 * 5.20) + (($units - 250) * 6.50);
}

echo "Electricity Bill = ₹" . $bill;
?>
```

4. Create a PHP script to manage an online shopping cart using arrays and loops.

```
<?php
$cart = [
    ["item" => "Shoes", "price" => 1500],
    ["item" => "Watch", "price" => 2000],
    ["item" => "T-Shirt", "price" => 500]
];
```

```
$total = 0;

foreach($cart as $product){
    echo $product["item"] . " - ₹" . $product["price"] . "<br>";
    $total += $product["price"];
}

echo "<h3>Total Bill: ₹$total</h3>";
?>
```

✅ EXP. 5 — OOP and Advanced OOP in PHP

Simple

1. Write a PHP program to demonstrate a class, object, and method.

```
<?php
class Student {
    public function display(){
        echo "Hello, I am a Student!";
    }
}

$obj = new Student();
$obj->display();
?>
```

2. PHP program showing inheritance (base class + derived class).

```
<?php
class Vehicle {
    public function type(){
        echo "This is a Vehicle<br>";
    }
}

class Car extends Vehicle {
    public function model(){
        echo "This is a Car Model.";
    }
}

$obj = new Car();
$obj->type();
$obj->model();
?>
```

Application Oriented

3. Implement a PHP program for a bank account system (deposit, withdraw, balance).

```
<?php
class Bank {
    private $balance = 0;

    function deposit($amt){
        $this->balance += $amt;
        echo "Deposited: ₹$amt <br>";
    }

    function withdraw($amt){
```

```

        if($amt <= $this->balance){
            $this->balance -= $amt;
            echo "Withdrawn: ₹$amt <br>";
        } else {
            echo "Insufficient Balance<br>";
        }
    }

    function showBalance(){
        echo "Current Balance: ₹" . $this->balance . "<br>";
    }
}

$acc = new Bank();
$acc->deposit(5000);
$acc->withdraw(1200);
$acc->showBalance();
?>

```

4. Employee management using constructors, destructors, interface, trait.

<?php

```

interface Work {
    public function doWork();
}

trait Bonus {
    public function giveBonus(){
        echo "Bonus granted!<br>";
    }
}

class Employee implements Work {
    use Bonus;

    public $name;
    public $salary;

    function __construct($name, $salary){
        $this->name = $name;
        $this->salary = $salary;
        echo "Employee Created: $name <br>";
    }

    public function doWork(){
        echo "$this->name is working.<br>";
    }

    function __destruct(){
        echo "Employee data removed.<br>";
    }
}

$emp = new Employee("Amar", 40000);
$emp->doWork();
$emp->giveBonus();
?>

```

✅ EXP. 6 — Upload Various Types of Files (With Validation)

Simple

1. Upload a single image with file size restriction.

HTML Form:

```
<form action="upload.php" method="post" enctype="multipart/form-data">
  Select Image:<br>
  <input type="file" name="photo">
  <button type="submit">Upload</button>
</form>
```

upload.php

```
<?php
$file = $_FILES['photo'];
$size = $file['size'];
$type = $file['type'];

if($size > 2000000){ // 2MB
    die("File too large!");
}

if($type != "image/jpeg" && $type != "image/png"){
    die("Only JPG/PNG allowed");
}

move_uploaded_file($file['tmp_name'], "uploads/".$file['name']);
echo "Uploaded Successfully!";
?>
```

2. Upload multiple files and display names

```
<?php
foreach($_FILES['files']['name'] as $key => $val){
    move_uploaded_file($_FILES['files']['tmp_name'][$key], "uploads/".$val);
    echo "Uploaded: $val <br>";
}
?>
```

Application Oriented

3. Build a document upload module with validation (PDF, DOCX).

```
<?php
$file = $_FILES['doc'];
$ext = pathinfo($file['name'], PATHINFO_EXTENSION);

$allowed = ["pdf", "docx"];

if(!in_array($ext, $allowed)){
    die("Only PDF/DOCX allowed");
}

move_uploaded_file($file['tmp_name'], "docs/".$file['name']);

echo "Document Uploaded!";
?>
```

4. PHP file upload with thumbnail preview + server-side validation

Thumbnail preview = frontend JS, example:

```
<input type="file" id="img" onchange="preview()">
<img id="thumb" width="100">

<script>
function preview(){
    document.getElementById('thumb').src = URL.createObjectURL(document.getElementById('img').files[0]);
}
</script>
```

✅ EXP. 7 — Create & Handle Session in PHP

Simple

1. Create a session & store username

```
<?php
session_start();
$_SESSION['username'] = "Amar";
echo "Session Created!";
?>
```

2. Check whether session is active

```
<?php
session_start();

if(isset($_SESSION['username'])){
    echo "Active Session: " . $_SESSION['username'];
}else{
    echo "No active session!";
}
?>
```

Application Oriented

3. Session-based counter storing visit count

```
<?php
session_start();

if(!isset($_SESSION['count'])){
    $_SESSION['count'] = 1;
} else {
    $_SESSION['count']++;
}

echo "Page visited " . $_SESSION['count'] . " times.";
?>
```

4. Preferences page storing user theme selection in session

theme.php

```
<form method="post">
    Select Theme:
    <select name="theme">
        <option>Light</option>
        <option>Dark</option>
    </select>
    <button>Select</button>
```

```
</form>
```

theme_save.php

```
<?php
session_start();
$_SESSION['theme'] = $_POST['theme'];
echo "Theme saved!";
?>
```

apply_theme.php

```
<?php
session_start();
echo "Your selected theme is: " . $_SESSION['theme'];
?>
```

EXP. 8 — Insert User Data Into MySQL Using PHP

Prerequisite Database Connection File (db.php)

Create once and reuse in all programs:

```
<?php
$conn = mysqli_connect("localhost", "root", "", "college");

if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}
?>
```

Simple

1. Insert student basic information into MySQL

HTML Form — student.html

```
<form action="insert_student.php" method="post">
    Name: <input type="text" name="name"><br>
    Roll No: <input type="text" name="roll"><br>
    <button type="submit">Submit</button>
</form>
```

insert_student.php

```
<?php
include("db.php");

$name = $_POST['name'];
$roll = $_POST['roll'];

$sql = "INSERT INTO students(name, roll_no) VALUES('$name', '$roll')";

if(mysqli_query($conn, $sql)){
    echo "Student Inserted Successfully!";
}else{
    echo "Error: " . mysqli_error($conn);
}
?>
```

2 Insert employee details (name, salary, department)

```
<?php
include("db.php");

$name = $_POST['name'];
```

```

$salary = $_POST['salary'];
$dept = $_POST['dept'];

$sql = "INSERT INTO employee(name, salary, dept) VALUES('$name', '$salary', '$dept')";

if(mysqli_query($conn, $sql)){
    echo "Employee Inserted!";
}else{
    echo mysqli_error($conn);
}
?>

```

Application Oriented

3. PHP application to store product details and manage inventory

HTML Form

```

<form action="product_add.php" method="post">
    Product Name: <input name="name"><br>
    Price: <input name="price"><br>
    Quantity: <input name="qty"><br>
    <button>Add Product</button>
</form>

```

product_add.php

```

<?php
include "db.php";

$n = $_POST['name'];
$p = $_POST['price'];
$q = $_POST['qty'];

$qry = "INSERT INTO products(name, price, qty) VALUES('$n','$p','$q')";
mysqli_query($conn, $qry);

echo "Product Added!";
?>

```

4. Registration form with validation + insert into MySQL

register.html

```

<form action="register.php" method="post" onsubmit="return validate()">
Name: <input id="name" name="name"><br>
Email: <input id="email" name="email"><br>
Password: <input id="pass" name="pass" type="password"><br>
<button>Register</button>
</form>

```

```

<script>
function validate(){
    if(document.getElementById("name").value==""){
        alert("Enter name");
        return false;
    }
    if(!document.getElementById("email").value.includes("@")){
        alert("Enter valid email");
        return false;
    }
    return true;
}

```

```
}  
</script>
```

register.php

```
<?php  
include "db.php";  
  
$name = $_POST['name'];  
$email = $_POST['email'];  
$pass = $_POST['pass'];  
  
$sql = "INSERT INTO users(name, email, password) VALUES('$name', '$email', '$pass')";  
  
mysqli_query($conn, $sql);  
  
echo "Registration Successful!";  
?>
```

EXP. 9 — Manage Session in PHP With Login Facility **Simple**

1. Simple login using sessions

login.html

```
<form action="login.php" method="post">  
Username: <input name="user"><br>  
Password: <input name="pass" type="password"><br>  
<button>Login</button>  
</form>
```

login.php

```
<?php  
session_start();  
  
$user = $_POST['user'];  
$pass = $_POST['pass'];  
  
if($user=="admin" && $pass=="123"){  
    $_SESSION['login'] = $user;  
    echo "Logged in!";  
}else{  
    echo "Invalid Credentials";  
}  
?>
```

2. Destroy session when user logs out

logout.php

```
<?php  
session_start();  
session_destroy();  
  
echo "You are logged out!";  
?>
```

Application Oriented

3. Complete login/logout module

login.php

```

<?php
session_start();
include "db.php";

$user = $_POST['user'];
$pass = $_POST['pass'];

$q = mysqli_query($conn, "SELECT * FROM users WHERE name='$user' AND password='$pass'");

if(mysqli_num_rows($q)==1){
    $_SESSION['user'] = $user;
    echo "Welcome, $user";
}else{
    echo "Invalid login";
}
?>

```

4. Role-based login (Admin/User)

login.php

```

<?php
session_start();
include "db.php";

$username = $_POST['user'];
$password = $_POST['pass'];

$q = mysqli_query($conn, "SELECT role FROM users WHERE name='$username' AND password='$password'");

if(mysqli_num_rows($q)==1){
    $row = mysqli_fetch_assoc($q);
    $_SESSION['role'] = $row['role'];

    if($row['role']=="admin"){
        echo "Welcome Admin!";
    } else {
        echo "Welcome User!";
    }
}
else{
    echo "Invalid Credentials";
}
?>

```

EXP. 10 — Cookies in PHP

Simple

1. Set a cookie & display its value

set.php

```

<?php
setcookie("username", "Amar", time()+3600);
echo "Cookie Set!";
?>

```

display.php

```

<?php
echo "Cookie Value: ".$_COOKIE['username'];
?>

```

2 Update an existing cookie & display old/new value

```
<?php
setcookie("username", "OldValue", time()+3600);
echo "Old: ".$_COOKIE['username'];

setcookie("username", "NewValue", time()+3600);
echo "<br>Updated!";
?>
```

Application Oriented

3. Cookie-based personalization (theme switcher)

theme_set.php

```
<?php
$theme = $_GET['theme'];
setcookie("site_theme", $theme, time()+3600);
echo "Theme Updated!";
?>
```

apply_theme.php

```
<?php
echo "Current Theme: " . $_COOKIE['site_theme'];
?>
```

4. “Recently visited products” using cookies (e-commerce)

```
<?php
$product = $_GET['p'];

if(!isset($_COOKIE['recent'])){
    $list = [$product];
} else {
    $list = explode(",", $_COOKIE['recent']);
    array_unshift($list, $product);
    $list = array_unique($list);
}

setcookie("recent", implode(",", $list), time()+3600);

echo "Recently Visited: " . implode(" → ", $list);
?>
```