

Web Programming

PHP Programming: Fundamentals to Server-Side
Projects



Fenerbahce University

Module 1 — PHP Setup & First Pages

Module overview

3/250

1

Goals:

- Understand server-side rendering
- Create index.php pages
- Run a local PHP server
- Mix PHP output with HTML

Learning result

By the end of this module:

- ✓ Understand server-side rendering
- ✓ Create index.php pages
- ✓ Run a local PHP server
- ✓ Mix PHP output with HTML

Mini result:

One .php file contains the PHP logic and the HTML output.

First index.php Page

Module 1 — PHP Setup & First Pages

6/250

index.php

```
<?php
$course = "PHP Basics";
$greeting = "Welcome to the first page";
?>
<h1><?php echo $course; ?></h1>
<p><?php echo $greeting; ?></p>
```

Sample Output

PHP Basics
Welcome to the first page

Echo Inside HTML

Module 1 — PHP Setup & First Pages

7/250

index.php

```
<?php
$studentName = "Ada";
$lesson = "PHP";
?>
<h1>Hello, <?php echo $studentName; ?></h1>
<p>Today we study <?php echo $lesson; ?>.</p>
```

Sample Output

Hello, Ada
Today we study PHP.

PHP Opening and Closing Tags

Module 1 — PHP Setup & First Pages

8/250

tags.php

```
<?php
$status = "ready";
?>
<h1>PHP Tag Demo</h1>
<p>The page is <?php echo $status; ?>.</p>
<?php
$nextStep = "write HTML after PHP";
?>
<p>Next: <?php echo $nextStep; ?></p>
```

Sample Output

PHP Tag Demo
The page is ready.
Next: write HTML after PHP

Comments and Readability

Module 1 — PHP Setup & First Pages

9/250

comments.php

```
<?php
// This value will be shown in the HTML.
$title = "Readable PHP";
$message = "Comments explain the intention.";
?>
<!-- HTML comments are visible in page source. -->
<h1><?php echo $title; ?></h1>
<p><?php echo $message; ?></p>
```

Sample Output

Readable PHP
Comments explain the intention.

Variables on a Page

Module 1 — PHP Setup & First Pages

10/250

profile.php

```
<?php
$name = "Mert";
$city = "Istanbul";
$department = "Web Design";
?>
<h1><?php echo $name; ?></h1>
<p>City: <?php echo $city; ?></p>
<p>Department: <?php echo $department; ?></p>
```

Sample Output

Mert
City: Istanbul
Department: Web Design

Run the Built-in Server

Module 1 — PHP Setup & First Pages

11/250

server demo

```
<!-- index.php -->
<?php
$port = 8000;
?>
<h1>Local PHP Server</h1>
<p>Run this command:</p>
<code>php -S localhost:<?php echo $port; ?></code>
```

Sample Output

Local PHP Server
Run this command:
php -S localhost:8000

Link to Another PHP Page

Module 1 — PHP Setup & First Pages

12/250

index.php

```
<?php
$productId = 12;
?>
<h1>Product List</h1>
<a href="details.php?id=?php echo $productId; ?">
  Open product <?php echo $productId; ?>
</a>
```

Sample Output

Product List

Open product 12

Link URL: details.php?id=12

Include Header and Footer

Module 1 — PHP Setup & First Pages

13/250

about.php

```
<?php
$pageTitle = "About Us";
include "includes/header.php";
?>
<main>
  <h1><?php echo $pageTitle; ?></h1>
  <p>Shared header and footer are reused.</p>
</main>
<?php include "includes/footer.php"; ?>
```

Sample Output

About Us
Shared header and footer are
reused.

header.php + about.php +
footer.php

Dynamic Date Output

Module 1 — PHP Setup & First Pages

14/250

date.php

```
<?php
$today = date("d.m.Y");
$time = date("H:i");
?>
<h1>Server Date</h1>
<p>Date: <?php echo $today; ?></p>
<p>Time: <?php echo $time; ?></p>
```

Sample Output

Server Date
Date: 26.04.2026
Time: 14:30

Minimal GET Form Preview

Module 1 — PHP Setup & First Pages

15/250

search.php

```
<?php
$keyword = $_GET["q"] ?? "";
?>
<form method="get">
  <input name="q" placeholder="Search">
  <button>Search</button>
</form>
<p>Keyword: <?php echo htmlspecialchars($keyword); ?></p>
```

Sample Output

GET q=php

Keyword: php

Browser Request Overview

Module 1 — PHP Setup & First Pages

16/250

request.php

```
<?php
$method = $_SERVER["REQUEST_METHOD"];
$path = $_SERVER["PHP_SELF"];
?>
<h1>Request Info</h1>
<p>Method: <?php echo $method; ?></p>
<p>Page: <?php echo $path; ?></p>
```

Sample Output

Request Info
Method: GET
Page: /request.php

Page Source vs PHP Source

Module 1 — PHP Setup & First Pages

17/250

source.php

```
<?php
$secret = "created on the server";
?>
<h1>Page Source</h1>
<p>The browser receives final HTML.</p>
<p><?php echo $secret; ?></p>
```

Sample Output

Page Source

The browser receives final HTML.
created on the server

PHP variable name is not
visible.

Error Display Basics

Module 1 — PHP Setup & First Pages

18/250

errors.php

```
<?php
ini_set("display_errors", "1");
error_reporting(E_ALL);
$message = "Development errors are visible.";
?>
<h1>Error Settings</h1>
<p><?php echo $message; ?></p>
```

Sample Output

Error Settings

Development errors are visible.

Use only during development.

Practice: Profile Page

Module 1 — PHP Setup & First Pages

19/250

profile-practice.php

```
<?php
$name = "Zeynep";
$school = "FBU";
$skill = "HTML + CSS + PHP";
?>
<h1><?php echo $name; ?></h1>
<p>School: <?php echo $school; ?></p>
<p>Skill: <?php echo $skill; ?></p>
```

Sample Output

Zeynep
School: FBU
Skill: HTML + CSS + PHP

Checkpoint: First PHP Page

Module 1 — PHP Setup & First Pages

20/250

checkpoint.php

```
<?php
$title = "Checkpoint";
$done = "index.php works";
$next = "add a form";
?>
<h1><?php echo $title; ?></h1>
<p><?php echo $done; ?></p>
<p>Next task: <?php echo $next; ?></p>
```

Sample Output

```
Checkpoint
index.php works
Next task: add a form
```

Module 2 — Variables, Types & Safe Output

Module overview

21/250

2

Goals:

- Store values in variables
- Use strings, numbers, booleans and null
- Cast form values
- Escape dynamic output

Learning result

By the end of this module:

- ✓ Store values in variables
- ✓ Use strings, numbers, booleans and null
- ✓ Cast form values
- ✓ Escape dynamic output

Mini result:

One .php file contains the PHP logic and the HTML output.

String Variables

Module 2 — Variables, Types & Safe Output

22/250

strings.php

```
<?php
$firstName = "Ali";
$lastName = "Yilmaz";
$fullName = $firstName . " " . $lastName;
?>
<h1>Student</h1>
<p>Name: <?php echo $fullName; ?></p>
```

Sample Output

Student
Name: Ali Yilmaz

Number Variables

Module 2 — Variables, Types & Safe Output

23/250

numbers.php

```
<?php
$price = 120;
$quantity = 3;
$total = $price * $quantity;
?>
<h1>Order</h1>
<p>Price: <?php echo $price; ?> TL</p>
<p>Total: <?php echo $total; ?> TL</p>
```

Sample Output

```
Order
Price: 120 TL
Total: 360 TL
```

Boolean Variables

Module 2 — Variables, Types & Safe Output

24/250

booleans.php

```
<?php
$isLoggedIn = true;
$message = $isLoggedIn ? "Dashboard is open" : "Please log in";
?>
<h1>Login State</h1>
<p><?php echo $message; ?></p>
```

Sample Output

Login State
Dashboard is open

Null and Default Values

Module 2 — Variables, Types & Safe Output

25/250

defaults.php

```
<?php
$nickname = null;
$displayName = $nickname ?? "Guest";
?>
<h1>User</h1>
<p>Name: <?php echo $displayName; ?></p>
```

Sample Output

User
Name: Guest

Type Checking

Module 2 — Variables, Types & Safe Output

26/250

types.php

```
<?php
$age = 21;
$type = gettype($age);
$isNumber = is_int($age) ? "yes" : "no";
?>
<h1>Type Check</h1>
<p>Type: <?php echo $type; ?></p>
<p>Integer? <?php echo $isNumber; ?></p>
```

Sample Output

```
Type Check
Type: integer
Integer? yes
```

String Interpolation

Module 2 — Variables, Types & Safe Output

27/250

interpolation.php

```
<?php
$name = "Ece";
$course = "PHP";
$message = "Hello $name, welcome to $course.";
?>
<h1>Message</h1>
<p><?php echo $message; ?></p>
```

Sample Output

Message
Hello Ece, welcome to PHP.

Concatenation in HTML

Module 2 — Variables, Types & Safe Output

28/250

concat.php

```
<?php
$brand = "FBU";
$page = "Web Lab";
$title = $brand . " - " . $page;
?>
<h1><?php echo $title; ?></h1>
<p>Concatenation joins text values.</p>
```

Sample Output

FBU - Web Lab
Concatenation joins text values.

Constants

Module 2 — Variables, Types & Safe Output

29/250

constants.php

```
<?php
define("APP_NAME", "Student Portal");
const VERSION = "1.0";
?>
<h1><?php echo APP_NAME; ?></h1>
<p>Version: <?php echo VERSION; ?></p>
```

Sample Output

Student Portal
Version: 1.0

Casting Form Text to Numbers

Module 2 — Variables, Types & Safe Output

30/250

calculator-get.php

```
<?php
$a = (float)($_GET["a"] ?? 0);
$b = (float)($_GET["b"] ?? 0);
$sum = $a + $b;
?>
<form method="get">
  <input name="a" value="2">
  <input name="b" value="5">
  <button>Add</button>
</form>
<p>Sum: <?php echo $sum; ?></p>
```

Sample Output

GET a=2&b=5

Sum: 7

Math Expressions

Module 2 — Variables, Types & Safe Output

31/250

math.php

```
<?php
$price = 200;
$taxRate = 0.20;
$tax = $price * $taxRate;
$total = $price + $tax;
?>
<h1>Invoice</h1>
<p>Tax: <?php echo $tax; ?> TL</p>
<p>Total: <?php echo $total; ?> TL</p>
```

Sample Output

```
Invoice
Tax: 40 TL
Total: 240 TL
```

Comparison Basics

Module 2 — Variables, Types & Safe Output

32/250

comparison.php

```
<?php
$score = 72;
$result = $score >= 60 ? "Passed" : "Failed";
?>
<h1>Exam Result</h1>
<p>Score: <?php echo $score; ?></p>
<p>Status: <?php echo $result; ?></p>
```

Sample Output

Exam Result
Score: 72
Status: Passed

Template Values

Module 2 — Variables, Types & Safe Output

33/250

template-values.php

```
<?php
$pageTitle = "Course Detail";
$courseCode = "WEB101";
?>
<header>
  <h1><?php echo $pageTitle; ?></h1>
</header>
<p>Code: <?php echo $courseCode; ?></p>
```

Sample Output

Course Detail
Code: WEB101

Escaping Output

Module 2 — Variables, Types & Safe Output

34/250

escaping.php

```
<?php
$comment = "<b>Hello</b> from user";
$safeComment = htmlspecialchars($comment);
?>
<h1>Safe Output</h1>
<p><?php echo $safeComment; ?></p>
```

Sample Output

Safe Output

Hello from user

HTML tags are shown as text.

Displaying Lists from Variables

Module 2 — Variables, Types & Safe Output

35/250

list-vars.php

```
<?php
$first = "HTML";
$second = "CSS";
$third = "PHP";
?>
<h1>Topics</h1>
<ul>
  <li><?php echo $first; ?></li>
  <li><?php echo $second; ?></li>
  <li><?php echo $third; ?></li>
</ul>
```

Sample Output

Topics

- HTML
- CSS
- PHP

Default Form Values

Module 2 — Variables, Types & Safe Output

36/250

default-form.php

```
<?php
$city = $_GET["city"] ?? "Istanbul";
?>
<form method="get">
  <input name="city" value="<?php echo htmlspecialchars($city); ?>">
  <button>Save</button>
</form>
<p>Selected city: <?php echo htmlspecialchars($city); ?></p>
```

Sample Output

Selected city: Istanbul

Input value is also Istanbul.

Query Parameter Variable

Module 2 — Variables, Types & Safe Output

37/250

details.php

```
<?php
$id = (int)($_GET["id"] ?? 0);
$message = $id > 0 ? "Product $id" : "No product selected";
?>
<h1>Details</h1>
<p><?php echo $message; ?></p>
```

Sample Output

GET id=42

Details

Product 42

Calculate Age from Year

Module 2 — Variables, Types & Safe Output

38/250

age.php

```
<?php
$birthYear = 2004;
$currentYear = 2026;
$age = $currentYear - $birthYear;
?>
<h1>Age Calculator</h1>
<p>Age: <?php echo $age; ?></p>
```

Sample Output

Age Calculator
Age: 22

Price Formatting

Module 2 — Variables, Types & Safe Output

39/250

price-format.php

```
<?php
$price = 1499.9;
$formatted = number_format($price, 2, ".", ",");
?>
<h1>Product Price</h1>
<p><?php echo $formatted; ?> TL</p>
```

Sample Output

Product Price
1,499.90 TL

Boolean Status Badge

Module 2 — Variables, Types & Safe Output

40/250

status.php

```
<?php
$isActive = false;
$statusText = $isActive ? "Active" : "Passive";
?>
<h1>Account</h1>
<p>Status: <?php echo $statusText; ?></p>
```

Sample Output

Account
Status: Passive

List Summary

Module 2 — Variables, Types & Safe Output

41/250

summary.php

```
<?php
$itemCount = 4;
$owner = "Lab Group A";
$summary = "$owner has $itemCount tasks.";
?>
<h1>Summary</h1>
<p><?php echo $summary; ?></p>
```

Sample Output

Summary
Lab Group A has 4 tasks.

Variable Scope Preview

Module 2 — Variables, Types & Safe Output

42/250

scope-preview.php

```
<?php
$title = "Outside value";
function makeLabel() {
    $title = "Inside function";
    return $title;
}
$label = makeLabel();
?>
<h1>Scope Preview</h1>
<p><?php echo $label; ?></p>
```

Sample Output

Scope Preview
Inside function

Debugging with var_dump

Module 2 — Variables, Types & Safe Output

43/250

debug.php

```
<?php
$age = "21";
$debug = gettype($age);
?>
<h1>Debug</h1>
<p>Type before casting: <?php echo $debug; ?></p>
<pre><?php var_dump($age); ?></pre>
```

Sample Output

```
Debug
Type before casting: string
string(2) "21"
```

Practice: Personal Card

Module 2 — Variables, Types & Safe Output

44/250

personal-card.php

```
<?php
$name = "Deniz";
$job = "Frontend Student";
$email = "deniz@example.com";
?>
<h1><?php echo $name; ?></h1>
<p><?php echo $job; ?></p>
<p><?php echo $email; ?></p>
```

Sample Output

Deniz
Frontend Student
deniz@example.com

Checkpoint: Variables and Types

Module 2 — Variables, Types & Safe Output

45/250

variables-check.php

```
<?php
$product = "Notebook";
$price = 75;
$inStock = true;
$status = $inStock ? "available" : "sold out";
?>
<h1><?php echo $product; ?></h1>
<p>Price: <?php echo $price; ?> TL</p>
<p>Status: <?php echo $status; ?></p>
```

Sample Output

```
Notebook
Price: 75 TL
Status: available
```

Module 3 — Operators & Control Flow

Module overview

46/250

3

Goals:

- Use if/else decisions
- Repeat HTML with loops
- Validate conditions
- Build small dynamic pages

Learning result

By the end of this module:

- ✓ Use if/else decisions
- ✓ Repeat HTML with loops
- ✓ Validate conditions
- ✓ Build small dynamic pages

Mini result:

One .php file contains the PHP logic and the HTML output.

if Greeting

Module 3 — Operators & Control Flow

47/250

if-greeting.php

```
<?php
$hour = 10;
if ($hour < 12) {
    $message = "Good morning";
}
?>
<h1>Greeting</h1>
<p><?php echo $message; ?></p>
```

Sample Output

Greeting
Good morning

if / else Branch

Module 3 — Operators & Control Flow

48/250

stock.php

```
<?php
$stock = 0;
if ($stock > 0) {
    $message = "Product is available";
} else {
    $message = "Product is out of stock";
}
?>
<h1>Stock</h1>
<p><?php echo $message; ?></p>
```

Sample Output

Stock
Product is out of stock

elseif Grade Example

Module 3 — Operators & Control Flow

49/250

grade.php

```
<?php
$score = 84;
if ($score >= 90) {
    $grade = "A";
} elseif ($score >= 70) {
    $grade = "B";
} else {
    $grade = "C";
}
?>
<h1>Grade</h1>
<p><?php echo $grade; ?></p>
```

Sample Output

Grade
B

switch for Shipping

Module 3 — Operators & Control Flow

50/250

shipping.php

```
<?php
$city = "Istanbul";
switch ($city) {
    case "Istanbul": $fee = 30; break;
    case "Ankara": $fee = 40; break;
    default: $fee = 55;
}
?>
<h1>Shipping</h1>
<p>Fee: <?php echo $fee; ?> TL</p>
```

Sample Output

Shipping
Fee: 30 TL

Ternary Operator

Module 3 — Operators & Control Flow

51/250

ternary.php

```
<?php
$age = 19;
$label = $age >= 18 ? "Adult" : "Under 18";
?>
<h1>User Type</h1>
<p><?php echo $label; ?></p>
```

Sample Output

User Type
Adult

Null Coalescing in Decisions

Module 3 — Operators & Control Flow

52/250

coalesce.php

```
<?php
$username = $_GET["user"] ?? "Guest";
$message = "Welcome, " . $username;
?>
<h1>Home</h1>
<p><?php echo htmlspecialchars($message); ?></p>
```

Sample Output

GET user=Ayse

Home

Welcome, Ayse

for Loop Numbers

Module 3 — Operators & Control Flow

53/250

for-loop.php

```
<?php
$limit = 5;
?>
<h1>Numbers</h1>
<ul>
<?php for ($i = 1; $i <= $limit; $i++): ?>
  <li><?php echo $i; ?></li>
<?php endfor; ?>
</ul>
```

Sample Output

Numbers

- 1
- 2
- 3
- 4
- 5

while Countdown

Module 3 — Operators & Control Flow

54/250

while.php

```
<?php
$counter = 3;
?>
<h1>Countdown</h1>
<ul>
<?php while ($counter > 0): ?>
  <li><?php echo $counter; ?></li>
  <?php $counter--; ?>
<?php endwhile; ?>
</ul>
```

Sample Output

Countdown

- 3
- 2
- 1

foreach Task List

Module 3 — Operators & Control Flow

55/250

tasks.php

```
<?php
$tasks = ["Read form", "Validate", "Show result"];
?>
<h1>Tasks</h1>
<ul>
<?php foreach ($tasks as $task): ?>
  <li><?php echo $task; ?></li>
<?php endforeach; ?>
</ul>
```

Sample Output

Tasks

- Read form
- Validate
- Show result

Nested Loop Table

Module 3 — Operators & Control Flow

56/250

table.php

```
<?php
$rows = 2;
$cols = 3;
?>
<table border="1">
<?php for ($r = 1; $r <= $rows; $r++): ?>
  <tr>
  <?php for ($c = 1; $c <= $cols; $c++): ?>
    <td><?php echo "$r,$c"; ?></td>
  <?php endfor; ?>
  </tr>
<?php endfor; ?>
</table>
```

Sample Output

```
1,1 | 1,2 | 1,3
2,1 | 2,2 | 2,3
```

continue in a Loop

Module 3 — Operators & Control Flow

57/250

continue.php

```
<?php
$numbers = [1, 2, 3, 4, 5];
?>
<h1>Odd Numbers</h1>
<ul>
<?php foreach ($numbers as $n): ?>
  <?php if ($n % 2 === 0) continue; ?>
  <li><?php echo $n; ?></li>
<?php endforeach; ?>
</ul>
```

Sample Output

Odd Numbers

- 1
- 3
- 5

break in a Loop

Module 3 — Operators & Control Flow

58/250

break.php

```
<?php
$item = ["HTML", "CSS", "PHP", "SQL"];
?>
<h1>Stop at PHP</h1>
<ul>
<?php foreach ($item as $item): ?>
  <li><?php echo $item; ?></li>
  <?php if ($item === "PHP") break; ?>
<?php endforeach; ?>
</ul>
```

Sample Output

Stop at PHP

- HTML
- CSS
- PHP

Loop over Numbers

Module 3 — Operators & Control Flow

59/250

sum-loop.php

```
<?php
$total = 0;
for ($i = 1; $i <= 4; $i++) {
    $total += $i;
}
?>
<h1>Loop Sum</h1>
<p>Total: <?php echo $total; ?></p>
```

Sample Output

Loop Sum
Total: 10

Loop over Checkboxes

Module 3 — Operators & Control Flow

60/250

checkboxes.php

```
<?php
$selected = $_POST["skills"] ?? [];
$count = count($selected);
?>
<form method="post">
  <label><input type="checkbox" name="skills[]" value="PHP"> PHP</label>
  <label><input type="checkbox" name="skills[]" value="SQL"> SQL</label>
  <button>Send</button>
</form>
<p>Selected: <?php echo $count; ?></p>
```

Sample Output

POST skills[]=PHP&skills[]=SQL

Selected: 2

Form Calculation Branch

Module 3 — Operators & Control Flow

61/250

calc-branch.php

```
<?php
$number = (int)($_GET["number"] ?? 0);
if ($number > 0) {
    $message = "Positive number";
} else {
    $message = "Zero or negative";
}
?>
<form method="get"><input name="number" value="5"><button>Check</button></form>
<p><?php echo $message; ?></p>
```

Sample Output

GET number=5

Positive number

Validation Branch

Module 3 — Operators & Control Flow

62/250

validate-branch.php

```
<?php
$email = $_POST["email"] ?? "";
if ($email === "") {
    $message = "Email is required";
} else {
    $message = "Email received";
}
?>
<form method="post"><input name="email"><button>Send</button></form>
<p><?php echo $message; ?></p>
```

Sample Output

POST email=

Email is required

Conditional Message

Module 3 — Operators & Control Flow

63/250

message-if.php

```
<?php
$points = 120;
$message = $points >= 100 ? "Free shipping unlocked" : "Keep shopping";
?>
<h1>Cart Message</h1>
<p><?php echo $message; ?></p>
```

Sample Output

Cart Message
Free shipping unlocked

User Role Menu

Module 3 — Operators & Control Flow

64/250

role-menu.php

```
<?php
$role = "admin";
?>
<h1>Menu</h1>
<ul>
  <li>Profile</li>
  <?php if ($role === "admin"): ?>
    <li>Admin Panel</li>
  <?php endif; ?>
</ul>
```

Sample Output

Menu

- Profile
- Admin Panel

Open Hours Message

Module 3 — Operators & Control Flow

65/250

open-hours.php

```
<?php
$hour = 18;
$isOpen = $hour >= 9 && $hour < 17;
$message = $isOpen ? "Office is open" : "Office is closed";
?>
<h1>Office</h1>
<p><?php echo $message; ?></p>
```

Sample Output

Office
Office is closed

Filter Search Results

Module 3 — Operators & Control Flow

66/250

filter.php

```
<?php
$products = ["Book", "Bag", "Bottle"];
$keyword = "Bo";
?>
<h1>Results</h1>
<ul>
<?php foreach ($products as $product): ?>
  <?php if (strpos($product, $keyword) === false) continue; ?>
  <li><?php echo $product; ?></li>
<?php endforeach; ?>
</ul>
```

Sample Output

Results

- Book
- Bottle

Conditional Form Message

Module 3 — Operators & Control Flow

67/250

conditional-form.php

```
<?php
$name = trim($_POST["name"] ?? "");
$message = $name === "" ? "Write your name" : "Hello, $name";
?>
<form method="post">
  <input name="name">
  <button>Send</button>
</form>
<p><?php echo htmlspecialchars($message); ?></p>
```

Sample Output

POST name=Ela

Hello, Ela

Loops Generate Select Options

Module 3 — Operators & Control Flow

68/250

select-loop.php

```
<?php
$years = [2024, 2025, 2026];
$selectedYear = 2026;
?>
<select name="year">
<?php foreach ($years as $year): ?>
  <option <?php if ($year === $selectedYear) echo "selected"; ?>>
    <?php echo $year; ?>
  </option>
<?php endforeach; ?>
</select>
```

Sample Output

Dropdown options:

2024

2025

2026 selected

Homework List with foreach

Module 3 — Operators & Control Flow

69/250

homework.php

```
<?php
$homeworks = ["Read PHP", "Build form", "Submit lab"];
?>
<h1>Homework</h1>
<ol>
<?php foreach ($homeworks as $work): ?>
  <li><?php echo $work; ?></li>
<?php endforeach; ?>
</ol>
```

Sample Output

Homework

1. Read PHP
2. Build form
3. Submit lab

Error Messages with if

Module 3 — Operators & Control Flow

70/250

errors-array-preview.php

```
<?php
$name = "";
$errors = [];
if ($name === "") {
    $errors[] = "Name is required";
}
?>
<h1>Errors</h1>
<?php foreach ($errors as $error): ?>
    <p><?php echo $error; ?></p>
<?php endforeach; ?>
```

Sample Output

Errors
Name is required

Simple Quiz Logic

Module 3 — Operators & Control Flow

71/250

quiz.php

```
<?php
$answer = $_GET["answer"] ?? "";
$correct = "php";
$result = strtolower($answer) === $correct ? "Correct" : "Try again";
?>
<form method="get"><input name="answer"><button>Check</button></form>
<p><?php echo $result; ?></p>
```

Sample Output

GET answer=php

Correct

Reading Query with if

Module 3 — Operators & Control Flow

72/250

read-query.php

```
<?php
$page = $_GET["page"] ?? "home";
if ($page === "contact") {
    $title = "Contact Page";
} else {
    $title = "Home Page";
}
?>
<h1><?php echo $title; ?></h1>
```

Sample Output

GET page=contact

Contact Page

Purchase Discount

Module 3 — Operators & Control Flow

73/250

discount.php

```
<?php
$total = 650;
if ($total >= 500) {
    $discount = 50;
} else {
    $discount = 0;
}
$payable = $total - $discount;
?>
<h1>Payment</h1>
<p>Pay: <?php echo $payable; ?> TL</p>
```

Sample Output

Payment
Pay: 600 TL

Practice: Decision Page

Module 3 — Operators & Control Flow

74/250

decision-practice.php

```
<?php
$temperature = 28;
$message = $temperature > 25 ? "Wear a t-shirt" : "Take a jacket";
?>
<h1>Weather Advice</h1>
<p><?php echo $message; ?></p>
```

Sample Output

Weather Advice
Wear a t-shirt

Checkpoint: Conditions and Loops

Module 3 — Operators & Control Flow

75/250

control-check.php

```
<?php
$roles = ["student", "teacher", "admin"];
?>
<h1>Allowed Roles</h1>
<ul>
<?php foreach ($roles as $role): ?>
  <?php if ($role === "admin") continue; ?>
  <li><?php echo $role; ?></li>
<?php endforeach; ?>
</ul>
```

Sample Output

Allowed Roles

- student
- teacher

Module 4 — Arrays & Superglobals

Module overview

76/250

4

Goals:

- Store lists and records
- Render arrays as HTML
- Use common array functions
- Understand request arrays

Learning result

By the end of this module:

- ✓ Store lists and records
- ✓ Render arrays as HTML
- ✓ Use common array functions
- ✓ Understand request arrays

Mini result:

One .php file contains the PHP logic and the HTML output.

Indexed Arrays

Module 4 — Arrays & Superglobals

77/250

indexed-array.php

```
<?php
$courses = ["HTML", "CSS", "PHP"];
?>
<h1>Courses</h1>
<ul>
<?php foreach ($courses as $course): ?>
  <li><?php echo $course; ?></li>
<?php endforeach; ?>
</ul>
```

Sample Output

Courses

- HTML
- CSS
- PHP

Associative Arrays

Module 4 — Arrays & Superglobals

78/250

assoc-array.php

```
<?php
$student = ["name" => "Selin", "grade" => 88];
?>
<h1>Student</h1>
<p>Name: <?php echo $student["name"]; ?></p>
<p>Grade: <?php echo $student["grade"]; ?></p>
```

Sample Output

```
Student
Name: Selin
Grade: 88
```

Multidimensional Arrays

Module 4 — Arrays & Superglobals

79/250

multi-array.php

```
<?php
$students = [
    ["name" => "Ali", "score" => 80],
    ["name" => "Ece", "score" => 92]
];
?>
<h1>Scores</h1>
<?php foreach ($students as $student): ?>
    <p><?php echo $student["name"]; ?>: <?php echo $student["score"]; ?></p>
<?php endforeach; ?>
```

Sample Output

Scores

Ali: 80

Ece: 92

array_push Example

Module 4 — Arrays & Superglobals

80/250

array-push.php

```
<?php
$tasks = ["Read"];
array_push($tasks, "Code", "Test");
?>
<h1>Tasks</h1>
<?php foreach ($tasks as $task): ?>
  <p><?php echo $task; ?></p>
<?php endforeach; ?>
```

Sample Output

Tasks
Read
Code
Test

array_merge Example

Module 4 — Arrays & Superglobals

81/250

array-merge.php

```
<?php
$frontend = ["HTML", "CSS"];
$backend = ["PHP", "SQL"];
$all = array_merge($frontend, $backend);
?>
<h1>Stack</h1>
<p><?php echo implode(" ", $all); ?></p>
```

Sample Output

Stack
HTML, CSS, PHP, SQL

count Items

Module 4 — Arrays & Superglobals

82/250

count-items.php

```
<?php
$products = ["Book", "Pen", "Bag"];
$count = count($products);
?>
<h1>Products</h1>
<p>Total items: <?php echo $count; ?></p>
```

Sample Output

Products
Total items: 3

in_array Check

Module 4 — Arrays & Superglobals

83/250

in-array.php

```
<?php
$roles = ["student", "teacher"];
$role = "admin";
$message = in_array($role, $roles) ? "Allowed" : "Not allowed";
?>
<h1>Role Check</h1>
<p><?php echo $message; ?></p>
```

Sample Output

Role Check
Not allowed

array_keys and array_values

Module 4 — Arrays & Superglobals

84/250

keys-values.php

```
<?php
$user = ["name" => "Can", "city" => "Izmir"];
$keys = array_keys($user);
$values = array_values($user);
?>
<h1>Array Parts</h1>
<p>Keys: <?php echo implode(" ", $keys); ?></p>
<p>Values: <?php echo implode(" ", $values); ?></p>
```

Sample Output

Array Parts
Keys: name, city
Values: Can, Izmir

array_map Transform

Module 4 — Arrays & Superglobals

85/250

array-map.php

```
<?php
$names = ["ada", "mert", "ece"];
$upperNames = array_map("strtoupper", $names);
?>
<h1>Upper Names</h1>
<p><?php echo implode(" ", $upperNames); ?></p>
```

Sample Output

Upper Names
ADA, MERT, ECE

array_filter Passed Students

Module 4 — Arrays & Superglobals

86/250

array-filter.php

```
<?php
$scores = [45, 70, 88, 50];
$passed = array_filter($scores, fn($score) => $score >= 60);
?>
<h1>Passed Scores</h1>
<p><?php echo implode(", ", $passed); ?></p>
```

Sample Output

Passed Scores
70, 88

array_reduce Cart Total

Module 4 — Arrays & Superglobals

87/250

array-reduce.php

```
<?php
$cart = [20, 35, 15];
$total = array_reduce($cart, fn($sum, $price) => $sum + $price, 0);
?>
<h1>Cart</h1>
<p>Total: <?php echo $total; ?> TL</p>
```

Sample Output

Cart
Total: 70 TL

Sorting Arrays

Module 4 — Arrays & Superglobals

88/250

sort-array.php

```
<?php
$names = ["Zeynep", "Ali", "Can"];
sort($names);
?>
<h1>Sorted Names</h1>
<p><?php echo implode(" ", $names); ?></p>
```

Sample Output

Sorted Names
Ali, Can, Zeynep

Product List as HTML

Module 4 — Arrays & Superglobals

89/250

products-list.php

```
<?php
$products = ["Laptop", "Mouse", "Keyboard"];
?>
<h1>Products</h1>
<ul>
<?php foreach ($products as $product): ?>
  <li><?php echo $product; ?></li>
<?php endforeach; ?>
</ul>
```

Sample Output

Products

- Laptop
- Mouse
- Keyboard

Student Cards from Arrays

Module 4 — Arrays & Superglobals

90/250

student-cards.php

```
<?php
$students = [
    ["name" => "Ada", "class" => "A"],
    ["name" => "Bora", "class" => "B"]
];
?>
<h1>Students</h1>
<?php foreach ($students as $student): ?>
    <article>
        <h2><?php echo $student["name"]; ?></h2>
        <p>Class: <?php echo $student["class"]; ?></p>
    </article>
<?php endforeach; ?>
```

Sample Output

Students

Ada - Class A

Bora - Class B

Table from Arrays

Module 4 — Arrays & Superglobals

91/250

array-table.php

```
<?php
$rows = [
    ["course" => "PHP", "hour" => 3],
    ["course" => "SQL", "hour" => 2]
];
?>
<table border="1">
<?php foreach ($rows as $row): ?>
    <tr><td><?php echo $row["course"]; ?></td><td><?php echo $row["hour"]; ?></td></tr>
<?php endforeach; ?>
</table>
```

Sample Output

```
PHP | 3
SQL | 2
```

Nested Categories

Module 4 — Arrays & Superglobals

92/250

nested-categories.php

```
<?php
$categories = [
    "Web" => ["HTML", "PHP"],
    "Data" => ["SQL", "JSON"]
];
?>
<?php foreach ($categories as $category => $items): ?>
    <h2><?php echo $category; ?></h2>
    <p><?php echo implode(", ", $items); ?></p>
<?php endforeach; ?>
```

Sample Output

```
Web
HTML, PHP

Data
SQL, JSON
```

POST Arrays Overview

Module 4 — Arrays & Superglobals

93/250

post-array-overview.php

```
<?php
$tags = $_POST["tags"] ?? [];
?>
<form method="post">
  <input name="tags[]" value="php">
  <input name="tags[]" value="forms">
  <button>Send</button>
</form>
<p>Tags: <?php echo implode(" ", $tags); ?></p>
```

Sample Output

POST tags[]=php&tags[]=forms

Tags: php, forms

Checkbox Arrays

Module 4 — Arrays & Superglobals

94/250

checkbox-array.php

```
<?php
$selected = $_POST["days"] ?? [];
?>
<form method="post">
  <label><input type="checkbox" name="days[]" value="Mon"> Mon</label>
  <label><input type="checkbox" name="days[]" value="Tue"> Tue</label>
  <button>Save</button>
</form>
<p>Days: <?php echo implode(" ", $selected); ?></p>
```

Sample Output

POST days[]=Mon&days[]=Tue

Days: Mon, Tue

Select Options from Arrays

Module 4 — Arrays & Superglobals

95/250

select-options.php

```
<?php
$cities = ["Istanbul", "Ankara", "Izmir"];
$city = $_GET["city"] ?? "Ankara";
?>
<select name="city">
<?php foreach ($cities as $item): ?>
  <option <?php if ($item === $city) echo "selected"; ?><?php echo $item; ?></option>
<?php endforeach; ?>
</select>
<p>Selected: <?php echo $city; ?></p>
```

Sample Output

Dropdown:

Istanbul

Ankara selected

Izmir

Selected: Ankara

CSV-like Array

Module 4 — Arrays & Superglobals

96/250

csv-array.php

```
<?php
$line = "Ada,PHP,90";
$parts = explode(",", $line);
$name = $parts[0];
$lesson = $parts[1];
$score = $parts[2];
?>
<h1>CSV Row</h1>
<p><?php echo $name; ?> - <?php echo $lesson; ?> - <?php echo $score; ?></p>
```

Sample Output

CSV Row
Ada - PHP - 90

JSON Preview

Module 4 — Arrays & Superglobals

97/250

json-preview.php

```
<?php
$user = ["name" => "Ece", "role" => "student"];
$json = json_encode($user);
?>
<h1>JSON</h1>
<pre><?php echo $json; ?></pre>
```

Sample Output

JSON

```
{"name":"Ece","role":"student"}
```

Array to Form Options

Module 4 — Arrays & Superglobals

98/250

array-form-options.php

```
<?php
$levels = ["Beginner", "Intermediate", "Advanced"];
?>
<form>
  <select name="level">
    <?php foreach ($levels as $level): ?>
      <option><?php echo $level; ?></option>
    <?php endforeach; ?>
  </select>
</form>
```

Sample Output

Select options:
Beginner
Intermediate
Advanced

Cart Totals

Module 4 — Arrays & Superglobals

99/250

cart-total.php

```
<?php
$cart = [
    ["name" => "Book", "price" => 80],
    ["name" => "Pen", "price" => 20]
];
$total = 0;
foreach ($cart as $item) { $total += $item["price"]; }
?>
<h1>Cart</h1>
<p>Total: <?php echo $total; ?> TL</p>
```

Sample Output

Cart
Total: 100 TL

Grouping Data

Module 4 — Arrays & Superglobals

100/250

grouping.php

```
<?php
$students = [
    "A" => ["Ada", "Ali"],
    "B" => ["Bora"]
];
?>
<?php foreach ($students as $class => $names): ?>
    <h2>Class <?php echo $class; ?></h2>
    <p><?php echo implode(" ", $names); ?></p>
<?php endforeach; ?>
```

Sample Output

Class A
Ada, Ali

Class B
Bora

Array Destructuring Preview

Module 4 — Arrays & Superglobals

101/250

destructuring.php

```
<?php
$product = ["Mouse", 350];
[$name, $price] = $product;
?>
<h1>Product</h1>
<p><?php echo $name; ?></p>
<p><?php echo $price; ?> TL</p>
```

Sample Output

Product
Mouse
350 TL

\$_SERVER Intro

Module 4 — Arrays & Superglobals

102/250

server-intro.php

```
<?php
$method = $_SERVER["REQUEST_METHOD"];
$agent = $_SERVER["HTTP_USER_AGENT"] ?? "Unknown";
?>
<h1>Server Data</h1>
<p>Method: <?php echo $method; ?></p>
<p>Agent exists: <?php echo $agent !== "Unknown" ? "yes" : "no"; ?></p>
```

Sample Output

```
Server Data
Method: GET
Agent exists: yes
```

Superglobals Map

Module 4 — Arrays & Superglobals

103/250

superglobals.php

```
<?php
$sources = ["GET", "POST", "SERVER", "FILES", "SESSION"];
?>
<h1>Superglobals</h1>
<ul>
<?php foreach ($sources as $source): ?>
  <li>$ <?php echo $source; ?></li>
<?php endforeach; ?>
</ul>
```

Sample Output

Superglobals

- \$_GET
- \$_POST
- \$_SERVER
- \$_FILES
- \$_SESSION

Practice: Product Data

Module 4 — Arrays & Superglobals

104/250

product-practice.php

```
<?php
$product = ["name" => "Keyboard", "price" => 850, "stock" => 7];
?>
<h1><?php echo $product["name"]; ?></h1>
<p>Price: <?php echo $product["price"]; ?> TL</p>
<p>Stock: <?php echo $product["stock"]; ?></p>
```

Sample Output

```
Keyboard
Price: 850 TL
Stock: 7
```

Checkpoint: Arrays

Module 4 — Arrays & Superglobals

105/250

arrays-check.php

```
<?php
$menu = ["Home" => "index.php", "Contact" => "contact.php"];
?>
<nav>
<?php foreach ($menu as $label => $url): ?>
  <a href="<?php echo $url; ?>"><?php echo $label; ?></a>
<?php endforeach; ?>
</nav>
```

Sample Output

Navigation:

Home → index.php

Contact → contact.php

Module 5 — Functions, Includes & Templates

Module overview

106/250

5

Goals:

- Write reusable functions
- Return values
- Use includes and partials
- Build small template helpers

Learning result

By the end of this module:

- ✓ Write reusable functions
- ✓ Return values
- ✓ Use includes and partials
- ✓ Build small template helpers

Mini result:

One .php file contains the PHP logic and the HTML output.

Declare a Function

Module 5 — Functions, Includes & Templates

107/250

functions.php

```
<?php
function makeTitle() {
    return "PHP Functions";
}
$title = makeTitle();
?>
<h1><?php echo $title; ?></h1>
```

Sample Output

PHP Functions

Function with Parameter

Module 5 — Functions, Includes & Templates

108/250

function-parameter.php

```
<?php
function greet($name) {
    return "Hello, " . $name;
}
$message = greet("Ada");
?>
<h1>Greeting</h1>
<p><?php echo $message; ?></p>
```

Sample Output

Greeting
Hello, Ada

Return Values

Module 5 — Functions, Includes & Templates

109/250

return-values.php

```
<?php
function calculateVat($price) {
    return $price * 0.20;
}
$vat = calculateVat(300);
?>
<h1>VAT</h1>
<p><?php echo $vat; ?> TL</p>
```

Sample Output

VAT
60 TL

Default Parameters

Module 5 — Functions, Includes & Templates

110/250

default-parameter.php

```
<?php
function formatUser($name, $role = "student") {
    return $name . " (" . $role . ")";
}
$label = formatUser("Ece");
?>
<p><?php echo $label; ?></p>
```

Sample Output

Ece (student)

Type Hints

Module 5 — Functions, Includes & Templates

111/250

type-hints.php

```
<?php
function addNumbers(int $a, int $b): int {
    return $a + $b;
}
$result = addNumbers(4, 6);
?>
<h1>Typed Result</h1>
<p><?php echo $result; ?></p>
```

Sample Output

Typed Result
10

Include Function File

Module 5 — Functions, Includes & Templates

112/250

include-functions.php

```
<?php
// helpers.php contains: function e($v) { ... }
include "helpers.php";
$name = "Ada";
?>
<h1>Include Demo</h1>
<p><?php echo e($name); ?></p>
```

Sample Output

Include Demo

Ada

helpers.php is reused.

Helper e() for Escaping

Module 5 — Functions, Includes & Templates

113/250

helper-e.php

```
<?php
function e($value) {
    return htmlspecialchars($value, ENT_QUOTES, "UTF-8");
}
$userInput = "<script>alert(1)</script>";
?>
<h1>Safe Text</h1>
<p><?php echo e($userInput); ?></p>
```

Sample Output

Safe Text

```
<script>alert(1)</script>
```

Shown as text, not executed.

Render List Item Function

Module 5 — Functions, Includes & Templates

114/250

render-li.php

```
<?php
function renderItem($text) {
    return "<li>" . htmlspecialchars($text) . "</li>";
}
$items = ["Plan", "Code", "Test"];
?>
<ul>
<?php foreach ($items as $item): ?>
    <?php echo renderItem($item); ?>
<?php endforeach; ?>
</ul>
```

Sample Output

- Plan
- Code
- Test

Form Component Function

Module 5 — Functions, Includes & Templates

115/250

form-component.php

```
<?php
function inputField($name, $label) {
    return '<label>' . $label . '<input name="' . $name . '"></label>';
}
?>
<form method="post">
    <?php echo inputField("email", "Email"); ?>
    <button>Send</button>
</form>
```

Sample Output

Form:
Email [input]
Send button

Named Arguments Preview

Module 5 — Functions, Includes & Templates

116/250

named-arguments.php

```
<?php
function badge($text, $prefix = "Status") {
    return $prefix . ": " . $text;
}
$message = badge(text: "Ready", prefix: "Server");
?>
<p><?php echo $message; ?></p>
```

Sample Output

Server: Ready

Variable Scope in Functions

Module 5 — Functions, Includes & Templates

117/250

function-scope.php

```
<?php
$title = "Page title";
function sectionTitle() {
    $title = "Section title";
    return $title;
}
?>
<h1><?php echo $title; ?></h1>
<h2><?php echo sectionTitle(); ?></h2>
```

Sample Output

Page title
Section title

Avoiding Global Variables

Module 5 — Functions, Includes & Templates

118/250

avoid-global.php

```
<?php
function priceWithTax($price, $rate) {
    return $price + ($price * $rate);
}
$total = priceWithTax(100, 0.20);
?>
<h1>Total</h1>
<p><?php echo $total; ?> TL</p>
```

Sample Output

Total
120 TL

Anonymous Function

Module 5 — Functions, Includes & Templates

119/250

anonymous-function.php

```
<?php
$format = function ($name) {
    return strtoupper($name);
};
$result = $format("php lab");
?>
<h1><?php echo $result; ?></h1>
```

Sample Output

PHP LAB

Arrow Function

Module 5 — Functions, Includes & Templates

120/250

arrow-function.php

```
<?php
$prices = [10, 20, 30];
$withVat = array_map(fn($p) => $p * 1.20, $prices);
?>
<h1>VAT Prices</h1>
<p><?php echo implode(", ", $withVat); ?></p>
```

Sample Output

VAT Prices
12, 24, 36

Reusable Page Template

Module 5 — Functions, Includes & Templates

121/250

page-template.php

```
<?php
function pageTitle($title) {
    return "<h1>" . htmlspecialchars($title) . "</h1>";
}
$title = "Reusable Template";
?>
<?php echo pageTitle($title); ?>
<p>Content changes, template stays.</p>
```

Sample Output

Reusable Template
Content changes, template stays.

Partials: Header and Footer

Module 5 — Functions, Includes & Templates

122/250

partials.php

```
<?php
$pageTitle = "Products";
include "partials/header.php";
?>
<h1><?php echo $pageTitle; ?></h1>
<p>Product content goes here.</p>
<?php include "partials/footer.php"; ?>
```

Sample Output

```
[Header]
Products
Product content goes here.
[Footer]
```

Simple Router Function

Module 5 — Functions, Includes & Templates

123/250

router.php

```
<?php
function currentPage() {
    return $_GET["page"] ?? "home";
}
$page = currentPage();
$title = $page === "about" ? "About" : "Home";
?>
<h1>?php echo $title; ?></h1>
```

Sample Output

GET page=about

About

Test a Small Function

Module 5 — Functions, Includes & Templates

124/250

test-function.php

```
<?php
function isPassing($score) {
    return $score >= 60;
}
$result = isPassing(75) ? "test passed" : "test failed";
?>
<h1>Function Test</h1>
<p><?php echo $result; ?></p>
```

Sample Output

Function Test
test passed

Checkpoint: Functions

Module 5 — Functions, Includes & Templates

125/250

functions-check.php

```
<?php
function totalPrice($price, $quantity) {
    return $price * $quantity;
}
$total = totalPrice(25, 4);
?>
<h1>Checkpoint</h1>
<p>Total: <?php echo $total; ?> TL</p>
```

Sample Output

Checkpoint
Total: 100 TL

Module 6 — HTML Forms with GET

Module overview

126/250

6

Goals:

- Build GET forms
- Read query strings
- Keep values in inputs
- Create searchable and shareable URLs

Learning result

By the end of this module:

- ✓ Build GET forms
- ✓ Read query strings
- ✓ Keep values in inputs
- ✓ Create searchable and shareable URLs

Mini result:

One .php file contains the PHP logic and the HTML output.

GET Form Basic

Module 6 — HTML Forms with GET

127/250

get-basic.php

```
<?php
$name = $_GET["name"] ?? "";
?>
<form method="get">
  <input name="name" placeholder="Name">
  <button>Send</button>
</form>
<p>Hello <?php echo htmlspecialchars($name); ?></p>
```

Sample Output

GET name=Ada

Hello Ada

Read q Parameter

Module 6 — HTML Forms with GET

128/250

read-q.php

```
<?php
$q = $_GET["q"] ?? "";
$message = $q === "" ? "Type a keyword" : "Searching for $q";
?>
<form method="get"><input name="q"><button>Search</button></form>
<p><?php echo htmlspecialchars($message); ?></p>
```

Sample Output

GET q=php

Searching for php

Keep Input Value

Module 6 — HTML Forms with GET

129/250

sticky-get.php

```
<?php
$q = $_GET["q"] ?? "";
?>
<form method="get">
  <input name="q" value="<?php echo htmlspecialchars($q); ?>">
  <button>Search</button>
</form>
<p>Current value: <?php echo htmlspecialchars($q); ?></p>
```

Sample Output

Input value: php
Current value: php

Multiple GET Fields

Module 6 — HTML Forms with GET

130/250

multiple-get.php

```
<?php
$city = $_GET["city"] ?? "";
$level = $_GET["level"] ?? "";
?>
<form method="get">
  <input name="city" placeholder="City">
  <input name="level" placeholder="Level">
  <button>Filter</button>
</form>
<p><?php echo htmlspecialchars("$city - $level"); ?></p>
```

Sample Output

GET city=Istanbul&level=Beginner

Istanbul - Beginner

Filter by Category

Module 6 — HTML Forms with GET

131/250

category-filter.php

```
<?php
$category = $_GET["category"] ?? "all";
$products = ["book" => "PHP Book", "pen" => "Blue Pen"];
?>
<h1>Products</h1>
<?php foreach ($products as $type => $name): ?>
    <?php if ($category !== "all" && $category !== $type) continue; ?>
    <p><?php echo $name; ?></p>
<?php endforeach; ?>
```

Sample Output

GET category=book

Products

PHP Book

Pagination Page Parameter

Module 6 — HTML Forms with GET

132/250

pagination.php

```
<?php
$page = max(1, (int)($_GET["page"] ?? 1));
$nextPage = $page + 1;
?>
<h1>Page <?php echo $page; ?></h1>
<a href="?page=?php echo $nextPage; ?>">Next</a>
```

Sample Output

Page 2
Next link: ?page=3

Sort Order Radio Buttons

Module 6 — HTML Forms with GET

133/250

sort-radio.php

```
<?php
$sort = $_GET["sort"] ?? "asc";
?>
<form method="get">
  <label><input type="radio" name="sort" value="asc"> Asc</label>
  <label><input type="radio" name="sort" value="desc"> Desc</label>
  <button>Apply</button>
</form>
<p>Sort: <?php echo htmlspecialchars($sort); ?></p>
```

Sample Output

GET sort=desc

Sort: desc

GET Checkbox

Module 6 — HTML Forms with GET

134/250

get-checkbox.php

```
<?php
$onlyStock = isset($_GET["stock"]);
$message = $onlyStock ? "Only in-stock items" : "All items";
?>
<form method="get">
  <label><input type="checkbox" name="stock" value="1"> In stock</label>
  <button>Filter</button>
</form>
<p><?php echo $message; ?></p>
```

Sample Output

GET stock=1

Only in-stock items

GET Radio Buttons

Module 6 — HTML Forms with GET

135/250

get-radio.php

```
<?php
$view = $_GET["view"] ?? "grid";
?>
<form method="get">
  <label><input type="radio" name="view" value="grid"> Grid</label>
  <label><input type="radio" name="view" value="list"> List</label>
  <button>Change</button>
</form>
<p>View: <?php echo htmlspecialchars($view); ?></p>
```

Sample Output

GET view=list

View: list

GET Select List

Module 6 — HTML Forms with GET

136/250

get-select.php

```
<?php
$level = $_GET["level"] ?? "beginner";
?>
<form method="get">
  <select name="level">
    <option value="beginner">Beginner</option>
    <option value="advanced">Advanced</option>
  </select>
  <button>Choose</button>
</form>
<p>Level: <?php echo htmlspecialchars($level); ?></p>
```

Sample Output

GET level=advanced

Level: advanced

GET Date Field

Module 6 — HTML Forms with GET

137/250

get-date.php

```
<?php
$date = $_GET["date"] ?? date("Y-m-d");
?>
<form method="get">
  <input type="date" name="date" value="<?php echo htmlspecialchars($date); ?>">
  <button>Show</button>
</form>
<p>Date: <?php echo htmlspecialchars($date); ?></p>
```

Sample Output

GET date=2026-04-26

Date: 2026-04-26

GET Number Fields

Module 6 — HTML Forms with GET

138/250

get-number.php

```
<?php
$min = (int)($_GET["min"] ?? 0);
$max = (int)($_GET["max"] ?? 100);
?>
<form method="get">
  <input type="number" name="min" value="<?php echo $min; ?>">
  <input type="number" name="max" value="<?php echo $max; ?>">
  <button>Filter</button>
</form>
<p>Range: <?php echo $min; ?> - <?php echo $max; ?></p>
```

Sample Output

Range: 10 - 50

Search Results List

Module 6 — HTML Forms with GET

139/250

search-results.php

```
<?php
$q = $_GET["q"] ?? "";
$items = ["PHP form", "CSS grid", "PHP file"];
?>
<h1>Results</h1>
<?php foreach ($items as $item): ?>
  <?php if ($q && stripos($item, $q) === false) continue; ?>
  <p><?php echo $item; ?></p>
<?php endforeach; ?>
```

Sample Output

GET q=php

Results

PHP form

PHP file

URL Encoding

Module 6 — HTML Forms with GET

140/250

url-encoding.php

```
<?php
$keyword = "web design";
$url = "search.php?q=" . urlencode($keyword);
?>
<h1>Encoded Link</h1>
<a href="<?php echo $url; ?>"><?php echo $url; ?></a>
```

Sample Output

Encoded Link
search.php?q=web+design

Links as GET Requests

Module 6 — HTML Forms with GET

141/250

links-get.php

```
<?php
$category = $_GET["category"] ?? "all";
?>
<a href="?category=books">Books</a>
<a href="?category=tech">Tech</a>
<p>Current: <?php echo htmlspecialchars($category); ?></p>
```

Sample Output

Clicked: ?category=books

Current: books

Hidden Field in GET

Module 6 — HTML Forms with GET

142/250

hidden-get.php

```
<?php
$source = $_GET["source"] ?? "unknown";
$q = $_GET["q"] ?? "";
?>
<form method="get">
  <input type="hidden" name="source" value="navbar">
  <input name="q">
  <button>Search</button>
</form>
<p>Source: <?php echo $source; ?></p>
```

Sample Output

GET source=navbar&q=php

Source: navbar

Validate GET Parameters

Module 6 — HTML Forms with GET

143/250

validate-get.php

```
<?php
$page = (int)($_GET["page"] ?? 1);
if ($page < 1) { $page = 1; }
?>
<h1>Validated Page</h1>
<p>Page: <?php echo $page; ?></p>
```

Sample Output

GET page=-5

Validated Page

Page: 1

Search with No Results

Module 6 — HTML Forms with GET

144/250

no-results.php

```
<?php
$q = $_GET["q"] ?? "";
$matches = [];
$message = $q && count($matches) === 0 ? "No results found" : "";
?>
<form method="get"><input name="q"><button>Search</button></form>
<p><?php echo $message; ?></p>
```

Sample Output

GET q=xyz

No results found

Bookmarkable GET URL

Module 6 — HTML Forms with GET

145/250

bookmarkable.php

```
<?php
$topic = $_GET["topic"] ?? "php";
$link = "lesson.php?topic=" . urlencode($topic);
?>
<h1>Bookmark</h1>
<p>Save this URL: <?php echo $link; ?></p>
```

Sample Output

Bookmark
Save this URL:
lesson.php?topic=php

GET vs Sensitive Data

Module 6 — HTML Forms with GET

146/250

get-sensitive.php

```
<?php
$username = $_GET["username"] ?? "";
$message = "GET values are visible in the URL.";
?>
<form method="get"><input name="username"><button>Send</button></form>
<p>User: <?php echo htmlspecialchars($username); ?></p>
<p><?php echo $message; ?></p>
```

Sample Output

URL contains username=ada

GET values are visible in the
URL.

Build a Query String

Module 6 — HTML Forms with GET

147/250

query-string.php

```
<?php
$params = ["q" => "php", "page" => 2];
$url = "search.php?" . http_build_query($params);
?>
<h1>Query String</h1>
<a href="<?php echo $url; ?>"><?php echo $url; ?></a>
```

Sample Output

Query String
search.php?q=php&page=2

Mini Project: Product Filter

Module 6 — HTML Forms with GET

148/250

product-filter-get.php

```
<?php
$category = $_GET["category"] ?? "all";
$products = ["book" => "PHP Book", "tech" => "Mouse"];
?>
<form method="get"><input name="category"><button>Filter</button></form>
<h1>Filtered Products</h1>
<?php foreach ($products as $type => $name): ?>
  <?php if ($category != "all" && $category != $type) continue; ?>
  <p><?php echo $name; ?></p>
<?php endforeach; ?>
```

Sample Output

GET category=tech

Filtered Products

Mouse

Practice: Search Page

Module 6 — HTML Forms with GET

149/250

practice-search.php

```
<?php
$q = trim($_GET["q"] ?? "");
$message = $q ? "You searched: $q" : "Search something";
?>
<form method="get"><input name="q"><button>Search</button></form>
<h1>Practice Search</h1>
<p><?php echo htmlspecialchars($message); ?></p>
```

Sample Output

Practice Search
You searched: php

Checkpoint: GET Forms

Module 6 — HTML Forms with GET

150/250

get-checkpoint.php

```
<?php
$q = $_GET["q"] ?? "";
$page = (int)($_GET["page"] ?? 1);
?>
<form method="get"><input name="q"><button>Search</button></form>
<p>Query: <?php echo htmlspecialchars($q); ?></p>
<p>Page: <?php echo $page; ?></p>
```

Sample Output

Query: forms
Page: 1

Module 7 — POST Forms & Validation

Module overview

151/250

7

Goals:

- Build POST forms
- Validate required data
- Sanitize and escape input
- Create server-side error messages

Learning result

By the end of this module:

- ✓ Build POST forms
- ✓ Validate required data
- ✓ Sanitize and escape input
- ✓ Create server-side error messages

Mini result:

One .php file contains the PHP logic and the HTML output.

POST Contact Form

Module 7 — POST Forms & Validation

152/250

contact.php

```
<?php
$name = trim($_POST["name"] ?? "");
$email = trim($_POST["email"] ?? "");
$message = $name ? "Thanks, $name" : "Please send the form";
?>
<form method="post">
  <input name="name" placeholder="Name">
  <input name="email" placeholder="Email">
  <button>Send</button>
</form>
<p><?php echo htmlspecialchars($message); ?></p>
```

Sample Output

POST name=Ada
&email=ada@example.com

Thanks, Ada

Required Fields

Module 7 — POST Forms & Validation

153/250

required.php

```
<?php
$title = trim($_POST["title"] ?? "");
$error = $title === "" ? "Title is required" : "";
?>
<form method="post"><input name="title"><button>Save</button></form>
<p><?php echo $error; ?></p>
```

Sample Output

POST title=

Title is required

Email Validation

Module 7 — POST Forms & Validation

154/250

email-validation.php

```
<?php
$email = $_POST["email"] ?? "";
$isValid = filter_var($email, FILTER_VALIDATE_EMAIL);
$message = $isValid ? "Valid email" : "Invalid email";
?>
<form method="post"><input name="email"><button>Check</button></form>
<p><?php echo $message; ?></p>
```

Sample Output

POST email=ada@example.com

Valid email

Select Validation

Module 7 — POST Forms & Validation

155/250

select-validation.php

```
<?php
$department = $_POST["department"] ?? "";
$message = $department == "" ? "Choose a department" : "Selected: $department";
?>
<form method="post">
  <select name="department"><option value="">Choose</option><option>Design</option></select>
  <button>Send</button>
</form>
<p><?php echo htmlspecialchars($message); ?></p>
```

Sample Output

POST department=Design

Selected: Design

Radio Validation

Module 7 — POST Forms & Validation

156/250

radio-validation.php

```
<?php
$level = $_POST["level"] ?? "";
$message = $level ? "Level: $level" : "Select a level";
?>
<form method="post">
  <label><input type="radio" name="level" value="Beginner"> Beginner</label>
  <label><input type="radio" name="level" value="Advanced"> Advanced</label>
  <button>Send</button>
</form>
<p><?php echo htmlspecialchars($message); ?></p>
```

Sample Output

POST level=Advanced

Level: Advanced

Checkbox Terms

Module 7 — POST Forms & Validation

157/250

terms.php

```
<?php
$accepted = isset($_POST["terms"]);
$message = $accepted ? "Registration can continue" : "Accept terms first";
?>
<form method="post">
  <label><input type="checkbox" name="terms"> I accept</label>
  <button>Register</button>
</form>
<p><?php echo $message; ?></p>
```

Sample Output

POST terms=on

Registration can continue

Textarea Feedback

Module 7 — POST Forms & Validation

158/250

feedback.php

```
<?php
$feedback = trim($_POST["feedback"] ?? "");
$message = $feedback ? "Feedback length: " . strlen($feedback) : "Write feedback";
?>
<form method="post"><textarea name="feedback"></textarea><button>Send</button></form>
<p><?php echo $message; ?></p>
```

Sample Output

POST feedback=Great lab

Feedback length: 9

Errors Array

Module 7 — POST Forms & Validation

159/250

errors-array.php

```
<?php
$errors = [];
$name = trim($_POST["name"] ?? "");
if ($name === "") { $errors[] = "Name is required"; }
?>
<form method="post"><input name="name"><button>Send</button></form>
<?php foreach ($errors as $error): ?>
  <p><?php echo $error; ?></p>
<?php endforeach; ?>
```

Sample Output

Name is required

Sticky Form Values

Module 7 — POST Forms & Validation

160/250

sticky-post.php

```
<?php
$name = $_POST["name"] ?? "";
?>
<form method="post">
  <input name="name" value="<?php echo htmlspecialchars($name); ?>">
  <button>Save</button>
</form>
<p>Current: <?php echo htmlspecialchars($name); ?></p>
```

Sample Output

POST name=Can

Input still shows Can

Current: Can

trim Input

Module 7 — POST Forms & Validation

161/250

trim-input.php

```
<?php
$rawName = $_POST["name"] ?? "";
$name = trim($rawName);
?>
<form method="post"><input name="name"><button>Send</button></form>
<p>Name: <?php echo htmlspecialchars($name); ?></p>
```

Sample Output

POST name=" Ada "

Name: Ada

htmlspecialchars Output

Module 7 — POST Forms & Validation

162/250

htmlspecialchars.php

```
<?php
$comment = $_POST["comment"] ?? "<b>Hi</b>";
$safeComment = htmlspecialchars($comment);
?>
<form method="post"><input name="comment"><button>Send</button></form>
<p><?php echo $safeComment; ?></p>
```

Sample Output

Output text:

Hi

Tags are not rendered.

filter_input Example

Module 7 — POST Forms & Validation

163/250

filter_input.php

```
<?php
$age = filter_input(INPUT_POST, "age", FILTER_VALIDATE_INT);
$message = $age ? "Age: $age" : "Enter a valid age";
?>
<form method="post"><input name="age"><button>Check</button></form>
<p><?php echo $message; ?></p>
```

Sample Output

POST age=22

Age: 22

Password Form

Module 7 — POST Forms & Validation

164/250

password-form.php

```
<?php
$password = $_POST["password"] ?? "";
$message = strlen($password) >= 8 ? "Strong enough" : "Use at least 8 chars";
?>
<form method="post"><input type="password" name="password"><button>Check</button></form>
<p><?php echo $message; ?></p>
```

Sample Output

POST password=abc

Use at least 8 chars

Simple Calculator POST

Module 7 — POST Forms & Validation

165/250

post-calculator.php

```
<?php
$a = (float)($_POST["a"] ?? 0);
$b = (float)($_POST["b"] ?? 0);
$result = $a + $b;
?>
<form method="post">
  <input name="a"><input name="b"><button>Add</button>
</form>
<p>Result: <?php echo $result; ?></p>
```

Sample Output

POST a=4&b=6

Result: 10

Unit Converter POST

Module 7 — POST Forms & Validation

166/250

converter.php

```
<?php
$km = (float)($_POST["km"] ?? 0);
$miles = $km * 0.621371;
?>
<form method="post"><input name="km"><button>Convert</button></form>
<p>Miles: <?php echo round($miles, 2); ?></p>
```

Sample Output

POST km=10

Miles: 6.21

Multi-Step Preview

Module 7 — POST Forms & Validation

167/250

multi-step.php

```
<?php
$step = (int)($_POST["step"] ?? 1);
$message = $step === 1 ? "Step 1: personal info" : "Step 2: confirmation";
?>
<form method="post">
  <input type="hidden" name="step" value="2">
  <button>Next</button>
</form>
<p><?php echo $message; ?></p>
```

Sample Output

Before submit:

Step 1: personal info

After submit:

Step 2: confirmation

Error Summary

Module 7 — POST Forms & Validation

168/250

error-summary.php

```
<?php
$errors = ["Name is required", "Email is invalid"];
?>
<h1>Error Summary</h1>
<ul>
<?php foreach ($errors as $error): ?>
  <li><?php echo $error; ?></li>
<?php endforeach; ?>
</ul>
```

Sample Output

Error Summary

- Name is required
- Email is invalid

Success Message

Module 7 — POST Forms & Validation

169/250

success-message.php

```
<?php
$message = $_SERVER["REQUEST_METHOD"] === "POST";
$message = $message ? "Form saved successfully" : "Fill the form";
?>
<form method="post"><input name="title"><button>Save</button></form>
<p><?php echo $message; ?></p>
```

Sample Output

After POST:
Form saved successfully

File Form Placeholder

Module 7 — POST Forms & Validation

170/250

file-placeholder.php

```
<?php
$message = "Choose a file, then upload.";
?>
<form method="post" enctype="multipart/form-data">
  <input type="file" name="document">
  <button>Upload</button>
</form>
<p><?php echo $message; ?></p>
```

Sample Output

Choose a file, then upload.

File input is ready.

CSRF Hidden Token

Module 7 — POST Forms & Validation

171/250

csrf-token.php

```
<?php
session_start();
$_SESSION["token"] = $_SESSION["token"] ?? bin2hex(random_bytes(8));
$token = $_SESSION["token"];
?>
<form method="post">
  <input type="hidden" name="token" value="<?php echo $token; ?>">
  <button>Send</button>
</form>
```

Sample Output

Hidden token is sent with POST.
Example: a8f13c09e2b1d004

Form Action Same Page

Module 7 — POST Forms & Validation

172/250

same-page.php

```
<?php
$message = "";
if ($_SERVER["REQUEST_METHOD"] === "POST") {
    $message = "Same page processed the form";
}
?>
<form method="post" action="same-page.php">
    <button>Submit</button>
</form>
<p><?php echo $message; ?></p>
```

Sample Output

After submit:
Same page processed the form

POST/Redirect/GET Pattern

Module 7 — POST Forms & Validation

173/250

prg.php

```
<?php
if ($_SERVER["REQUEST_METHOD"] === "POST") {
    header("Location: thanks.php");
    exit;
}
?>
<form method="post"><input name="message"><button>Send</button></form>
<p>POST redirects to thanks.php</p>
```

Sample Output

Submit form
→ POST /prg.php
→ redirect
→ GET /thanks.php

Sanitize Numbers

Module 7 — POST Forms & Validation

174/250

sanitize-number.php

```
<?php
$quantity = filter_input(INPUT_POST, "quantity", FILTER_SANITIZE_NUMBER_INT);
$quantity = (int)$quantity;
?>
<form method="post"><input name="quantity"><button>Save</button></form>
<p>Quantity: <?php echo $quantity; ?></p>
```

Sample Output

POST quantity=12abc

Quantity: 12

Validate Date

Module 7 — POST Forms & Validation

175/250

validate-date.php

```
<?php
$date = $_POST["date"] ?? "";
$valid = DateTime::createFromFormat("Y-m-d", $date) !== false;
$message = $valid ? "Valid date" : "Invalid date";
?>
<form method="post"><input type="date" name="date"><button>Check</button></form>
<p><?php echo $message; ?></p>
```

Sample Output

POST date=2026-04-26

Valid date

POST Array Names

Module 7 — POST Forms & Validation

176/250

post-array-names.php

```
<?php
$names = $_POST["names"] ?? [];
?>
<form method="post">
  <input name="names[]" value="Ada">
  <input name="names[]" value="Ece">
  <button>Send</button>
</form>
<p><?php echo implode(", ", $names); ?></p>
```

Sample Output

POST names[]=Ada&names[]=Ece

Ada, Ece

Registration Form

Module 7 — POST Forms & Validation

177/250

registration.php

```
<?php
$username = trim($_POST["username"] ?? "");
$email = trim($_POST["email"] ?? "");
$message = $username && $email ? "Account request created" : "Fill all fields";
?>
<form method="post"><input name="username"><input name="email"><button>Register</button></form>
<p><?php echo $message; ?></p>
```

Sample Output

POST username=ada
&email=ada@example.com

Account request created

Server-only Validation

Module 7 — POST Forms & Validation

178/250

server-validation.php

```
<?php
$price = (float)($_POST["price"] ?? 0);
$message = $price > 0 ? "Accepted" : "Server rejected invalid price";
?>
<form method="post"><input name="price"><button>Save</button></form>
<p><?php echo $message; ?></p>
```

Sample Output

POST price=-10

Server rejected invalid price

Mini Project: Feedback Form

Module 7 — POST Forms & Validation

179/250

feedback-project.php

```
<?php
$name = trim($_POST["name"] ?? "");
$feedback = trim($_POST["feedback"] ?? "");
$message = $name && $feedback ? "Thank you, $name" : "Name and feedback required";
?>
<form method="post"><input name="name"><textarea name="feedback"></textarea><button>Send</button></form>
<p><?php echo htmlspecialchars($message); ?></p>
```

Sample Output

POST name=Ece&feedback=Good

Thank you, Ece

Checkpoint: POST and Validation

Module 7 — POST Forms & Validation

180/250

post-checkpoint.php

```
<?php
$title = trim($_POST["title"] ?? "");
$errors = [];
if ($title === "") { $errors[] = "Title is required"; }
$message = count($errors) ? $errors[0] : "Saved";
?>
<form method="post"><input name="title"><button>Save</button></form>
<p><?php echo $message; ?></p>
```

Sample Output

Without title:
Title is required

With title:
Saved

Module 8 — File Writing, Reading & Uploads

Module overview

181/250

8

Goals:

- Write and append text files
- Read files safely
- Use CSV and JSON files
- Handle basic uploads

Learning result

By the end of this module:

- ✓ Write and append text files
- ✓ Read files safely
- ✓ Use CSV and JSON files
- ✓ Handle basic uploads

Mini result:

One .php file contains the PHP logic and the HTML output.

Write Text File

Module 8 — File Writing, Reading & Uploads

182/250

write-file.php

```
<?php
$text = $_POST["text"] ?? "Hello file";
file_put_contents("message.txt", $text);
$message = "Saved to message.txt";
?>
<form method="post"><input name="text"><button>Save</button></form>
<p><?php echo $message; ?></p>
```

Sample Output

POST text=Hello file

Saved to message.txt

Append Text File

Module 8 — File Writing, Reading & Uploads

183/250

append-file.php

```
<?php
$line = $_POST["line"] ?? "New note";
file_put_contents("notes.txt", $line . PHP_EOL, FILE_APPEND);
$message = "Line appended";
?>
<form method="post"><input name="line"><button>Add</button></form>
<p><?php echo $message; ?></p>
```

Sample Output

Line appended
notes.txt grows by one line.

Read File Content

Module 8 — File Writing, Reading & Uploads

184/250

read-file.php

```
<?php
$content = file_exists("message.txt") ? file_get_contents("message.txt") : "No file yet";
?>
<h1>File Content</h1>
<pre><?php echo htmlspecialchars($content); ?></pre>
```

Sample Output

File Content
Hello file

file_exists Check

Module 8 — File Writing, Reading & Uploads

185/250

file-exists.php

```
<?php
$path = "data/notes.txt";
$message = file_exists($path) ? "File found" : "File missing";
?>
<h1>File Check</h1>
<p><?php echo $message; ?></p>
```

Sample Output

File Check
File found

filesize Example

Module 8 — File Writing, Reading & Uploads

186/250

filesize.php

```
<?php
$path = "message.txt";
$size = file_exists($path) ? filesize($path) : 0;
?>
<h1>File Size</h1>
<p><?php echo $size; ?> bytes</p>
```

Sample Output

File Size
24 bytes

filemtime Example

Module 8 — File Writing, Reading & Uploads

187/250

filemtime.php

```
<?php
$path = "message.txt";
$time = file_exists($path) ? date("H:i", filemtime($path)) : "unknown";
?>
<h1>Last Update</h1>
<p><?php echo $time; ?></p>
```

Sample Output

Last Update
14:30

Read Lines with file()

Module 8 — File Writing, Reading & Uploads

188/250

read-lines.php

```
<?php
$lines = file_exists("notes.txt") ? file("notes.txt", FILE_IGNORE_NEW_LINES) : [];
?>
<h1>Notes</h1>
<?php foreach ($lines as $line): ?>
  <p><?php echo htmlspecialchars($line); ?></p>
<?php endforeach; ?>
```

Sample Output

Notes
First note
Second note

CSV Write

Module 8 — File Writing, Reading & Uploads

189/250

csv-write.php

```
<?php
$row = ["Ada", "PHP", 90];
$file = fopen("grades.csv", "a");
fputcsv($file, $row);
fclose($file);
$message = "CSV row saved";
?>
<p><?php echo $message; ?></p>
```

Sample Output

CSV row saved

New row: Ada,PHP,90

CSV Read

Module 8 — File Writing, Reading & Uploads

190/250

csv-read.php

```
<?php
$rows = [];
if (($file = fopen("grades.csv", "r")) !== false) {
    while (($row = fgetcsv($file)) !== false) { $rows[] = $row; }
    fclose($file);
}
?>
<h1>Grades</h1>
<?php foreach ($rows as $row): ?>
    <p><?php echo $row[0]; ?>: <?php echo $row[2]; ?></p>
<?php endforeach; ?>
```

Sample Output

Grades

Ada: 90

JSON Write

Module 8 — File Writing, Reading & Uploads

191/250

json-write.php

```
<?php
$user = ["name" => "Mert", "role" => "student"];
file_put_contents("user.json", json_encode($user));
$message = "JSON saved";
?>
<p><?php echo $message; ?></p>
```

Sample Output

JSON saved
user.json contains user data.

JSON Read

Module 8 — File Writing, Reading & Uploads

192/250

json-read.php

```
<?php
$json = file_exists("user.json") ? file_get_contents("user.json") : "{}";
$user = json_decode($json, true);
$name = $user["name"] ?? "Unknown";
?>
<h1>User</h1>
<p><?php echo htmlspecialchars($name); ?></p>
```

Sample Output

User
Mert

Guestbook Append

Module 8 — File Writing, Reading & Uploads

193/250

guestbook.php

```
<?php
$name = trim($_POST["name"] ?? "");
if ($name) { file_put_contents("guestbook.txt", $name . PHP_EOL, FILE_APPEND); }
$message = $name ? "Guest added" : "Write a name";
?>
<form method="post"><input name="name"><button>Add</button></form>
<p><?php echo $message; ?></p>
```

Sample Output

POST name=Ayse

Guest added

Simple Log File

Module 8 — File Writing, Reading & Uploads

194/250

log-file.php

```
<?php
$event = "Page opened at " . date("H:i");
file_put_contents("app.log", $event . PHP_EOL, FILE_APPEND);
$message = "Log written";
?>
<p><?php echo $message; ?></p>
```

Sample Output

Log written
app.log: Page opened at 14:30

Delete File

Module 8 — File Writing, Reading & Uploads

195/250

delete-file.php

```
<?php
$path = "old.txt";
if (file_exists($path)) {
    unlink($path);
    $message = "File deleted";
} else {
    $message = "File not found";
}
?>
<p><?php echo $message; ?></p>
```

Sample Output

File deleted

Directory List

Module 8 — File Writing, Reading & Uploads

196/250

directory-list.php

```
<?php
$files = is_dir("uploads") ? scandir("uploads") : [];
?>
<h1>Upload Folder</h1>
<?php foreach ($files as $file): ?>
  <?php if ($file === "." || $file === "..") continue; ?>
  <p><?php echo htmlspecialchars($file); ?></p>
<?php endforeach; ?>
```

Sample Output

Upload Folder
avatar.png
notes.txt

Create Folder

Module 8 — File Writing, Reading & Uploads

197/250

create-folder.php

```
<?php
$folder = "uploads";
if (!is_dir($folder)) { mkdir($folder); }
$message = "Folder ready: " . $folder;
?>
<p><?php echo $message; ?></p>
```

Sample Output

Folder ready: uploads

Upload Form Basic

Module 8 — File Writing, Reading & Uploads

198/250

upload-basic.php

```
<?php
$message = "Select a file";
if (isset($_FILES["photo"])) {
    $message = "Received: " . $_FILES["photo"]["name"];
}
?>
<form method="post" enctype="multipart/form-data">
  <input type="file" name="photo">
  <button>Upload</button>
</form>
<p><?php echo htmlspecialchars($message); ?></p>
```

Sample Output

POST file=avatar.png

Received: avatar.png

Validate Upload Extension

Module 8 — File Writing, Reading & Uploads

199/250

upload-extension.php

```
<?php
$name = $_FILES["photo"]["name"] ?? "";
$ext = strtolower(pathinfo($name, PATHINFO_EXTENSION));
$message = in_array($ext, ["jpg", "png"]) ? "Allowed file" : "Only jpg/png";
?>
<form method="post" enctype="multipart/form-data"><input type="file" name="photo"><button>Upload</button></form>
<p><?php echo $message; ?></p>
```

Sample Output

file=avatar.png

Allowed file

Validate Upload Size

Module 8 — File Writing, Reading & Uploads

200/250

upload-size.php

```
<?php
$size = $_FILES["document"]["size"] ?? 0;
$message = $size <= 1000000 ? "Size accepted" : "File too large";
?>
<form method="post" enctype="multipart/form-data"><input type="file" name="document"><button>Upload</button></form>
<p><?php echo $message; ?></p>
```

Sample Output

file size=540 KB

Size accepted

Rename Uploaded File

Module 8 — File Writing, Reading & Uploads

201/250

upload-rename.php

```
<?php
$message = "No file";
if (isset($_FILES["photo"])) {
    $ext = pathinfo($_FILES["photo"]["name"], PATHINFO_EXTENSION);
    $newName = "upload_" . time() . "." . $ext;
    $message = "New name: " . $newName;
}
?>
<form method="post" enctype="multipart/form-data"><input type="file" name="photo"><button>Upload</button></form>
<p><?php echo htmlspecialchars($message); ?></p>
```

Sample Output

file=avatar.png

New name: upload_1714140000.png

Read Uploaded Text File

Module 8 — File Writing, Reading & Uploads

202/250

upload-read-text.php

```
<?php
$content = "";
if (isset($_FILES["note"])) {
    $content = file_get_contents($_FILES["note"]["tmp_name"]);
}
?>
<form method="post" enctype="multipart/form-data"><input type="file" name="note"><button>Read</button></form>
<pre><?php echo htmlspecialchars($content); ?></pre>
```

Sample Output

Uploaded note.txt

First line

Second line

Download Link

Module 8 — File Writing, Reading & Uploads

203/250

download-link.php

```
<?php
$file = "report.txt";
$message = file_exists($file) ? "Download ready" : "No report yet";
?>
<h1><?php echo $message; ?></h1>
<a href="<?php echo $file; ?>" download>Download report</a>
```

Sample Output

Download ready
Link: Download report

File Permissions

Module 8 — File Writing, Reading & Uploads

204/250

permissions.php

```
<?php
$path = "data";
$message = is_writable($path) ? "Folder is writable" : "Check permissions";
?>
<h1>Permissions</h1>
<p><?php echo $message; ?></p>
```

Sample Output

Permissions
Folder is writable

Safe File Names

Module 8 — File Writing, Reading & Uploads

205/250

safe-file-name.php

```
<?php
$original = "my photo.png";
$safe = preg_replace("/[^a-zA-Z0-9._-]/", "_", $original);
?>
<h1>Safe Name</h1>
<p><?php echo $safe; ?></p>
```

Sample Output

Safe Name
my_photo.png

Handling File Errors

Module 8 — File Writing, Reading & Uploads

206/250

file-errors.php

```
<?php
$error = $_FILES["photo"]["error"] ?? UPLOAD_ERR_NO_FILE;
$message = $error == UPLOAD_ERR_OK ? "Upload success" : "Upload error: $error";
?>
<form method="post" enctype="multipart/form-data"><input type="file" name="photo"><button>Upload</button></form>
<p><?php echo $message; ?></p>
```

Sample Output

No file selected

Upload error: 4

Build Notes App with Files

Module 8 — File Writing, Reading & Uploads

207/250

notes-app.php

```
<?php
$note = trim($_POST["note"] ?? "");
if ($note) { file_put_contents("notes.txt", $note . PHP_EOL, FILE_APPEND); }
$notes = file_exists("notes.txt") ? file("notes.txt", FILE_IGNORE_NEW_LINES) : [];
?>
<form method="post"><input name="note"><button>Add</button></form>
<?php foreach ($notes as $item): ?>
  <p><?php echo htmlspecialchars($item); ?></p>
<?php endforeach; ?>
```

Sample Output

After adding:
Remember validation
Finish lab

Export Report

Module 8 — File Writing, Reading & Uploads

208/250

export-report.php

```
<?php
$report = "Students: 24" . PHP_EOL . "Passed: 20";
file_put_contents("report.txt", $report);
$message = "Report exported";
?>
<p><?php echo $message; ?></p>
<a href="report.txt" download>Download</a>
```

Sample Output

Report exported
Download link is visible.

Import CSV

Module 8 — File Writing, Reading & Uploads

209/250

import-csv.php

```
<?php
$students = [];
foreach (file("students.csv", FILE_IGNORE_NEW_LINES) as $line) {
    $students[] = str_getcsv($line);
}
?>
<h1>Imported Students</h1>
<?php foreach ($students as $student): ?>
    <p><?php echo $student[0]; ?> - <?php echo $student[1]; ?></p>
<?php endforeach; ?>
```

Sample Output

Imported Students

Ada - 90

Mert - 75

Checkpoint: Files

Module 8 — File Writing, Reading & Uploads

210/250

files-checkpoint.php

```
<?php
$path = "notes.txt";
$count = file_exists($path) ? count(file($path)) : 0;
?>
<h1>File Checkpoint</h1>
<p>Note count: <?php echo $count; ?></p>
```

Sample Output

File Checkpoint
Note count: 3

Module 9 — Sessions, Cookies & Login Flow

Module overview

211/250

9

Goals:

- Remember users with sessions
- Use cookies for preferences
- Hash and verify passwords
- Protect pages

Learning result

By the end of this module:

- ✓ Remember users with sessions
- ✓ Use cookies for preferences
- ✓ Hash and verify passwords
- ✓ Protect pages

Mini result:

One .php file contains the PHP logic and the HTML output.

Start a Session

Module 9 — Sessions, Cookies & Login Flow

212/250

session-start.php

```
<?php
session_start();
$_SESSION["started"] = "yes";
$message = "Session started";
?>
<h1>Session</h1>
<p><?php echo $message; ?></p>
```

Sample Output

Session
Session started

Session Counter

Module 9 — Sessions, Cookies & Login Flow

213/250

session-counter.php

```
<?php
session_start();
$_SESSION["visits"] = ($_SESSION["visits"] ?? 0) + 1;
$visits = $_SESSION["visits"];
?>
<h1>Counter</h1>
<p>Visits: <?php echo $visits; ?></p>
```

Sample Output

Counter
Visits: 3

Store Username in Session

Module 9 — Sessions, Cookies & Login Flow

214/250

session-user.php

```
<?php
session_start();
if (isset($_POST["username"])) { $_SESSION["username"] = $_POST["username"]; }
$username = $_SESSION["username"] ?? "Guest";
?>
<form method="post"><input name="username"><button>Save</button></form>
<p>User: <?php echo htmlspecialchars($username); ?></p>
```

Sample Output

POST username=Ada

User: Ada

Flash Message

Module 9 — Sessions, Cookies & Login Flow

215/250

flash.php

```
<?php
session_start();
$_SESSION["flash"] = $_SESSION["flash"] ?? "Profile saved";
$message = $_SESSION["flash"];
unset($_SESSION["flash"]);
?>
<p><?php echo $message; ?></p>
```

Sample Output

First request:
Profile saved

Next request:
message disappears

Logout with session_destroy

Module 9 — Sessions, Cookies & Login Flow

216/250

logout.php

```
<?php
session_start();
session_destroy();
$message = "You are logged out";
?>
<h1>Logout</h1>
<p><?php echo $message; ?></p>
```

Sample Output

Logout
You are logged out

Set a Cookie

Module 9 — Sessions, Cookies & Login Flow

217/250

set-cookie.php

```
<?php
setcookie("theme", "dark", time() + 86400);
$message = "Theme cookie is being set";
?>
<h1>Cookie</h1>
<p><?php echo $message; ?></p>
```

Sample Output

Cookie
Theme cookie is being set

Read a Cookie

Module 9 — Sessions, Cookies & Login Flow

218/250

read-cookie.php

```
<?php
$theme = $_COOKIE["theme"] ?? "light";
?>
<h1>Theme</h1>
<p>Current: <?php echo htmlspecialchars($theme); ?></p>
```

Sample Output

Theme
Current: dark

Expire a Cookie

Module 9 — Sessions, Cookies & Login Flow

219/250

expire-cookie.php

```
<?php  
setcookie("theme", "", time() - 3600);  
$message = "Theme cookie expired";  
?>  
<p><?php echo $message; ?></p>
```

Sample Output

Theme cookie expired

Remember Theme with Cookie

Module 9 — Sessions, Cookies & Login Flow

220/250

theme-cookie.php

```
<?php
$theme = $_POST["theme"] ?? ($_COOKIE["theme"] ?? "light");
setcookie("theme", $theme, time() + 86400);
?>
<form method="post"><input name="theme" value="dark"><button>Save</button></form>
<p>Theme: <?php echo htmlspecialchars($theme); ?></p>
```

Sample Output

POST theme=dark

Theme: dark

Login Form POST

Module 9 — Sessions, Cookies & Login Flow

221/250

login.php

```
<?php
session_start();
$email = $_POST["email"] ?? "";
if ($email === "ada@example.com") { $_SESSION["user"] = $email; }
$message = isset($_SESSION["user"]) ? "Logged in" : "Enter email";
?>
<form method="post"><input name="email"><button>Login</button></form>
<p><?php echo $message; ?></p>
```

Sample Output

POST email=ada@example.com

Logged in

password_hash Example

Module 9 — Sessions, Cookies & Login Flow

222/250

password-hash.php

```
<?php
$password = "secret123";
$hash = password_hash($password, PASSWORD_DEFAULT);
$message = "Password is hashed";
?>
<h1>Hash</h1>
<p><?php echo $message; ?></p>
```

Sample Output

Hash

Password is hashed

Stored value is a long hash.

password_verify Example

Module 9 — Sessions, Cookies & Login Flow

223/250

password-verify.php

```
<?php
$hash = password_hash("secret123", PASSWORD_DEFAULT);
$valid = password_verify("secret123", $hash);
$message = $valid ? "Password correct" : "Wrong password";
?>
<p><?php echo $message; ?></p>
```

Sample Output

Password correct

Protect a Page

Module 9 — Sessions, Cookies & Login Flow

224/250

protect-page.php

```
<?php
session_start();
$isLoggedIn = isset($_SESSION["user"]);
$message = $isLoggedIn ? "Dashboard content" : "Please log in first";
?>
<h1>Protected Page</h1>
<p><?php echo $message; ?></p>
```

Sample Output

Protected Page
Please log in first

Authentication Message

Module 9 — Sessions, Cookies & Login Flow

225/250

auth-message.php

```
<?php
session_start();
$user = $_SESSION["user"] ?? null;
$message = $user ? "Welcome $user" : "Guest view";
?>
<h1>Auth</h1>
<p><?php echo htmlspecialchars($message); ?></p>
```

Sample Output

Auth
Welcome ada@example.com

Role Check

Module 9 — Sessions, Cookies & Login Flow

226/250

role-check.php

```
<?php
session_start();
$_SESSION["role"] = "editor";
$role = $_SESSION["role"];
$message = $role === "admin" ? "Admin tools" : "Standard tools";
?>
<p><?php echo $message; ?></p>
```

Sample Output

Standard tools

Cart in Session

Module 9 — Sessions, Cookies & Login Flow

227/250

session-cart.php

```
<?php
session_start();
$_SESSION["cart"] = $_SESSION["cart"] ?? [];
$_SESSION["cart"][] = "Book";
$count = count($_SESSION["cart"]);
?>
<h1>Cart</h1>
<p>Items: <?php echo $count; ?></p>
```

Sample Output

Cart
Items: 1

CSRF Token with Session

Module 9 — Sessions, Cookies & Login Flow

228/250

session-csrf.php

```
<?php
session_start();
$_SESSION["csrf"] = $_SESSION["csrf"] ?? bin2hex(random_bytes(8));
$token = $_SESSION["csrf"];
?>
<form method="post">
  <input type="hidden" name="csrf" value="<?php echo $token; ?>">
  <button>Submit</button>
</form>
```

Sample Output

Form sends token with POST.
Server compares it with session.

Session Security Basics

Module 9 — Sessions, Cookies & Login Flow

229/250

session-security.php

```
<?php
session_start();
session_regenerate_id(true);
$message = "Session id regenerated after login";
?>
<p><?php echo $message; ?></p>
```

Sample Output

Session id regenerated after
login

Checkpoint: Sessions and Cookies

Module 9 — Sessions, Cookies & Login Flow

230/250

session-checkpoint.php

```
<?php
session_start();
$_SESSION["name"] = $_SESSION["name"] ?? "Guest";
$name = $_SESSION["name"];
$theme = $_COOKIE["theme"] ?? "light";
?>
<h1>Checkpoint</h1>
<p>User: <?php echo htmlspecialchars($name); ?></p>
<p>Theme: <?php echo htmlspecialchars($theme); ?></p>
```

Sample Output

Checkpoint
User: Guest
Theme: light

Module 10 — PDO, CRUD & Final Project

Module overview

231/250

10

Goals:

- Connect with PDO
- Use prepared statements
- Render database records
- Combine PHP skills in a CRUD app

Learning result

By the end of this module:

- ✓ Connect with PDO
- ✓ Use prepared statements
- ✓ Render database records
- ✓ Combine PHP skills in a CRUD app

Mini result:

One .php file contains the PHP logic and the HTML output.

Connect to Database with PDO

Module 10 — PDO, CRUD & Final Project

232/250

db-connect.php

```
<?php
$dns = "mysql:host=localhost;dbname=school;charset=utf8mb4";
$pdo = new PDO($dns, "root", "");
$message = "PDO connection created";
?>
<h1>Database</h1>
<p><?php echo $message; ?></p>
```

Sample Output

Database
PDO connection created

Create Table Concept

Module 10 — PDO, CRUD & Final Project

233/250

create-table.php

```
<?php
$sql = "CREATE TABLE posts (id INT, title VARCHAR(100))";
?>
<h1>Create Table</h1>
<pre><?php echo htmlspecialchars($sql); ?></pre>
```

Sample Output

Create Table
CREATE TABLE posts (...)

Insert with Prepared Statement

Module 10 — PDO, CRUD & Final Project

234/250

insert-pdo.php

```
<?php
$title = $_POST["title"] ?? "First post";
$stmt = $pdo->prepare("INSERT INTO posts (title) VALUES (?)");
$stmt->execute([$title]);
$message = "Post inserted";
?>
<form method="post"><input name="title"><button>Add</button></form>
<p><?php echo $message; ?></p>
```

Sample Output

POST title=First post

Post inserted

Prepared SELECT

Module 10 — PDO, CRUD & Final Project

235/250

select-pdo.php

```
<?php
$id = (int)($_GET["id"] ?? 1);
$stmt = $pdo->prepare("SELECT * FROM posts WHERE id = ?");
$stmt->execute([$id]);
$post = ["title" => "First post"];
?>
<h1><?php echo htmlspecialchars($post["title"]); ?></h1>
```

Sample Output

GET id=1

First post

fetchAll Records

Module 10 — PDO, CRUD & Final Project

236/250

fetchall.php

```
<?php
$stmt = $pdo->query("SELECT title FROM posts");
$post = ["title" => "PHP intro"], ["title" => "Forms"];
?>
<h1>Posts</h1>
<?php foreach ($posts as $post): ?>
  <p><?php echo htmlspecialchars($post["title"]); ?></p>
<?php endforeach; ?>
```

Sample Output

Posts
PHP intro
Forms

Render Database Table

Module 10 — PDO, CRUD & Final Project

237/250

db-table.php

```
<?php
$users = [{"name" => "Ada", "email" => "ada@example.com"}];
?>
<table border="1">
<?php foreach ($users as $user): ?>
  <tr><td><?php echo htmlspecialchars($user["name"]); ?></td><td><?php echo htmlspecialchars($user["email"]); ?></td></tr>
<?php endforeach; ?>
</table>
```

Sample Output

Ada | ada@example.com

Edit Form

Module 10 — PDO, CRUD & Final Project

238/250

edit-form.php

```
<?php
$post = ["id" => 5, "title" => "Old title"];
?>
<form method="post" action="update.php">
  <input type="hidden" name="id" value="<?php echo $post["id"]; ?>">
  <input name="title" value="<?php echo htmlspecialchars($post["title"]); ?>">
  <button>Update</button>
</form>
```

Sample Output

Edit form
id hidden value: 5
title value: Old title

Update Record

Module 10 — PDO, CRUD & Final Project

239/250

update-record.php

```
<?php
$id = (int)($_POST["id"] ?? 0);
$title = $_POST["title"] ?? "";
$stmt = $pdo->prepare("UPDATE posts SET title=? WHERE id=?");
$stmt->execute([$title, $id]);
$message = "Post updated";
?>
<p><?php echo $message; ?></p>
```

Sample Output

POST id=5&title=New title

Post updated

Delete Record

Module 10 — PDO, CRUD & Final Project

240/250

delete-record.php

```
<?php
$id = (int)($_POST["id"] ?? 0);
$stmt = $pdo->prepare("DELETE FROM posts WHERE id=?");
$stmt->execute([$id]);
$message = "Post deleted";
?>
<form method="post"><input type="hidden" name="id" value="5"><button>Delete</button></form>
<p><?php echo $message; ?></p>
```

Sample Output

POST id=5

Post deleted

Search with LIKE

Module 10 — PDO, CRUD & Final Project

241/250

search-like.php

```
<?php
$q = $_GET["q"] ?? "";
$stmt = $pdo->prepare("SELECT * FROM posts WHERE title LIKE ?");
$stmt->execute(["%$q%"]);
$results = [{"title" => "PHP forms"}];
?>
<form method="get"><input name="q"><button>Search</button></form>
<?php foreach ($results as $row): ?>
    <p><?php echo htmlspecialchars($row["title"]); ?></p>
<?php endforeach; ?>
```

Sample Output

GET q=forms

PHP forms

Transactions

Module 10 — PDO, CRUD & Final Project

242/250

transactions.php

```
<?php
pdo->beginTransaction();
pdo->prepare("INSERT INTO orders (total) VALUES (?)")->execute([250]);
pdo->prepare("INSERT INTO logs (text) VALUES (?)")->execute(["order created"]);
pdo->commit();
$message = "Transaction completed";
?>
<p><?php echo $message; ?></p>
```

Sample Output

Transaction completed
Both inserts are saved together.

Count Rows

Module 10 — PDO, CRUD & Final Project

243/250

count-rows.php

```
<?php
$stmt = $pdo->query("SELECT COUNT(*) FROM posts");
$count = 12;
?>
<h1>Post Count</h1>
<p><?php echo $count; ?></p>
```

Sample Output

Post Count
12

PDO Error Handling

Module 10 — PDO, CRUD & Final Project

244/250

pdo-errors.php

```
<?php
try {
    $pdo = new PDO($dsn, "root", "");
    $message = "Connected";
} catch (PDOException $e) {
    $message = "Database error";
}
?>
<p><?php echo $message; ?></p>
```

Sample Output

Connected

or

Database error

Database Pagination

Module 10 — PDO, CRUD & Final Project

245/250

db-pagination.php

```
<?php
$page = max(1, (int)($_GET["page"] ?? 1));
$limit = 5;
$offset = ($page - 1) * $limit;
$sql = "SELECT * FROM posts LIMIT $limit OFFSET $offset";
?>
<h1>Page <?php echo $page; ?></h1>
<pre><?php echo $sql; ?></pre>
```

Sample Output

```
Page 2
SELECT * FROM posts LIMIT 5
OFFSET 5
```

Join Users and Posts

Module 10 — PDO, CRUD & Final Project

246/250

join.php

```
<?php
$sql = "SELECT users.name, posts.title FROM posts JOIN users ON users.id = posts.user_id";
$rows = [{"name" => "Ada", "title" => "PHP notes"}];
?>
<h1>User Posts</h1>
<?php foreach ($rows as $row): ?>
  <p><?php echo $row["name"]; ?>: <?php echo $row["title"]; ?></p>
<?php endforeach; ?>
```

Sample Output

User Posts
Ada: PHP notes

Safe Output from Database

Module 10 — PDO, CRUD & Final Project

247/250

safe-db-output.php

```
<?php
$post = ["title" => "<b>PHP</b> Notes"];
$safeTitle = htmlspecialchars($post["title"]);
?>
<h1>Post</h1>
<p><?php echo $safeTitle; ?></p>
```

Sample Output

Post

PHP Notes

HTML is displayed as text.

Configuration File

Module 10 — PDO, CRUD & Final Project

248/250

config-file.php

```
<?php
// config.php
$dbName = "school";
$dsn = "mysql:host=localhost;dbname=" . $dbName;
?>
<h1>Config</h1>
<p>Database: <?php echo $dbName; ?></p>
```

Sample Output

Config
Database: school

Mini CRUD Overview

Module 10 — PDO, CRUD & Final Project

249/250

crud-overview.php

```
<?php
$action = ["Create", "Read", "Update", "Delete"];
?>
<h1>CRUD</h1>
<ul>
<?php foreach ($actions as $action): ?>
  <li><?php echo $action; ?></li>
<?php endforeach; ?>
</ul>
```

Sample Output

CRUD

- Create
- Read
- Update
- Delete

Final Project Checklist

Module 10 — PDO, CRUD & Final Project

250/250

final-checklist.php

```
<?php
$steps = ["Build form", "Validate", "Save with PDO", "List records"];
?>
<h1>Final Project</h1>
<ol>
<?php foreach ($steps as $step): ?>
  <li><?php echo $step; ?></li>
<?php endforeach; ?>
</ol>
```

Sample Output

Final Project

1. Build form
2. Validate
3. Save with PDO
4. List records