Tishk International University
Faculty of Applied Science
Information Technology Department

Date, Time & Form Data Handling

Web Programming

Lecture 6

Fall 2025

Course Code: IT349

Grade 3

Islam Abdulazeez
islam.abdulaziz@tiu.edu.iq

November 5, 2025

Outlines

- Date & Time in PHP
- Form Handling in PHP
- Handling Different Form Input Types
- Coding Examples
- Classwork

Learning Outcomes

- At the end of today's session, you will be able to:

  - ✓ **Understand PHP date/time and form methods (GET/POST).**
  - ✓ **Apply $_GET, $_POST, and timezone settings.**
  - ✓ **Analyze how different inputs send data.**
  - ✓ **Create a multi-page form in PHP.**

Date and Time

- PHP provides built-in functions and classes to **get, display, and manipulate** date and time information.
- These functions rely on the server's clock and timezone settings.

date(format, timestamp)

- format: Specifies how to format the date/time.
- timestamp: Optional; defaults to the current time.

Common Date Format Characters

Character	Description	Example
Y	4-digit year	2025
y	2-digit year	25
m	Month (01–12)	11
d	Day (01–31)	03
H	Hour (00–23)	22
h	Hour (01–12)	10
i	Minutes	35
s	Seconds	12
A	AM or PM	PM

Date Format Example

- Example

```php
<?php
echo date("Y/m/d H:i:s A");
?>
```

2025/11/05 08:14:25 AM

Wrong Time!!

Set Timezone

- By default, PHP gets its timezone from the php.ini configuration file or the system's default timezone — and sometimes that default is UTC (Coordinated Universal Time) or another region, not your local time zone.

- **How to solve this issue?**

```
date_default_timezone_set("Asia/Baghdad");
```

Set Timezone

- Example

```php
<?php
date_default_timezone_set("Asia/Baghdad");
echo date("Y-m-d H:i:s");
?>
```

2025/11/05 09:12:43 AM

Custom Timestamp

- A custom timestamp means you create a specific date and time manually.

mktime(hour, minute, second, month, day, year)

```php
<?php
$time = mktime(14, 30, 0, 11, 4, 2025);
echo date("Y-m-d H:i:s A", $time);
?>
```

2025-11-04 14:30:00 PM

What is Form Handling?

- Form handling is the process of sending user input from one page to another.
- PHP processes the data entered into HTML forms.

➢ Usually involves two pages:

  - Form Page: Contains the HTML form.
  - Processing Page: Receives and handles the submitted data.

Action & Method

- action: Page that will receive the data.
- method: Defines how the data will be sent (GET or POST).

```
<form action="" method="">

</form>
```

Difference Between GET & POST

Feature	GET	POST
Visibility	Data visible in URL	Data hidden
Security	Less secure	More secure
Data Limit	2048 characters	No limit
Use Case	Searching, filters	Login, form submission

Handling & Receiving Input Text Data

Handling Input Text Data

- Text fields let users enter custom data (e.g., name, email).
- PHP captures and processes input for storage or display.
- Used for login, registration, or general form entries.

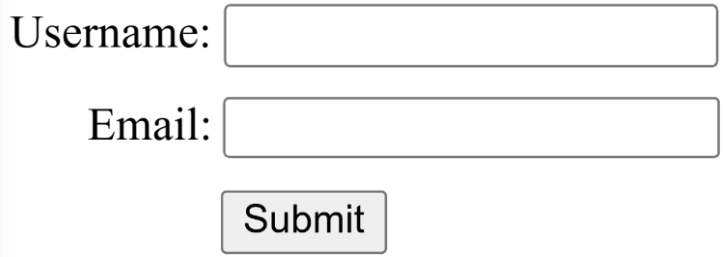
Receiving and Checking Data

- PHP retrieves submitted text input using $_POST or $_GET.
- Data is validated to ensure completeness and correct format.
- Feedback is provided based on validation results.

GET

- Example (firstPage.php)

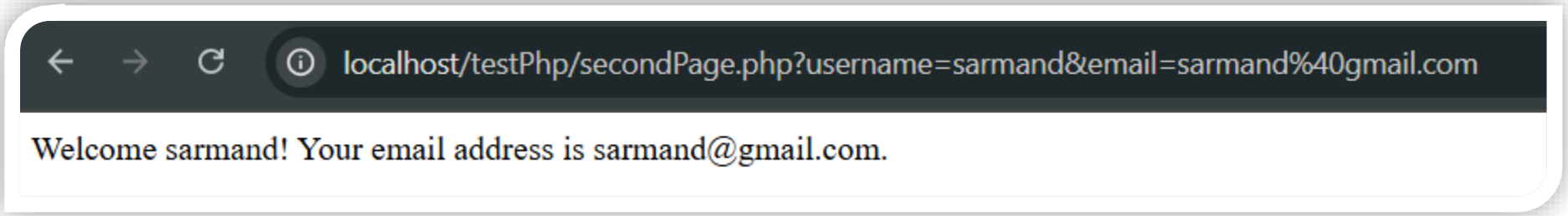
```
<form action="secondPage.php" method="GET">
  <label for="username">Username: </label>
  <input type="text" name="username"><br>
  <label for="email">Email: </label>
  <input type="email" name="email"><br>
  <input type="submit" value="Submit">
</form>
```



GET

- Example (secondPage.php)

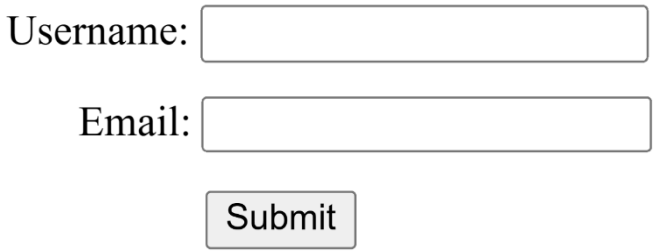
```php
<?php
$name = $_GET['username'];
$email = $_GET['email'];
echo "Welcome $name! Your email address is $email.";
?>
```



POST

- Example (firstPage.php)

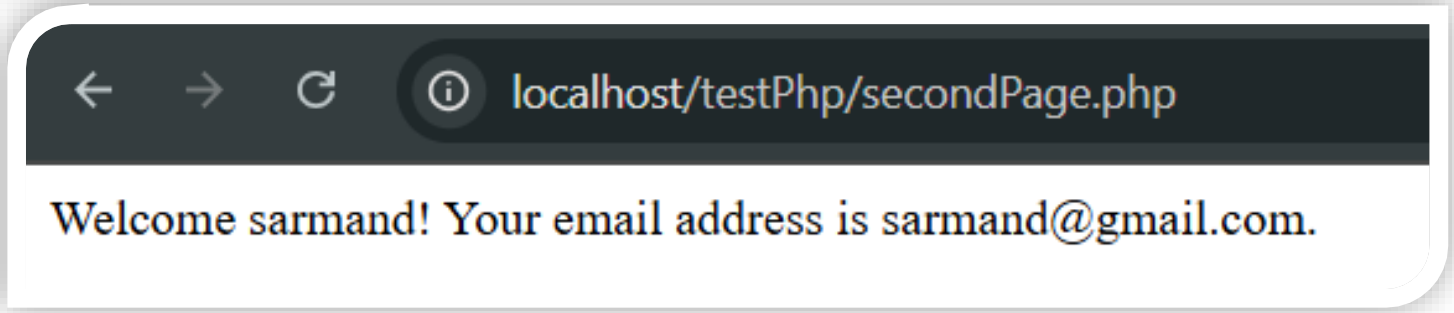
```
<form action="secondPage.php" method="POST">
  <label for="username">Username: </label>
  <input type="text" name="username"><br>
  <label for="email">Email: </label>
  <input type="email" name="email"><br>
  <input type="submit" value="Submit">
</form>
```



POST

- Example (secondPage.php)

```php
<?php
$name = $_POST['username'];
$email = $_POST['email'];
echo "Welcome $name! Your email is $email.";
?>
```



POST

- Example (login.php)

```
<h2>Login</h2>
<form action="check_login.php" method="post">
    <label>Username:</label>
    <input type="text" name="username"><br><br>

    <label>Password:</label>
    <input type="password" name="password"><br><br>

    <input type="submit" value="Login">
</form>
```

POST

- Example (check_login.php)

```php
<?php
 $username = $_POST['username'];
 $password = $_POST['password'];

 if ($username == "user123" && $password == "pass123") {
     echo "<p style='color:green;'>Login successful! Welcome, $username.</p>";
   } else {
     echo "<p style='color:red;'>Invalid username or password.</p>";
   }
?>
```

Handling & Receiving Checkboxes Data

Handling Checkboxes Data

- Checkboxes let users select multiple options.
- Selected values are sent to the server as an array.
- PHP can store or display the chosen options.

Receiving Checkboxes Data

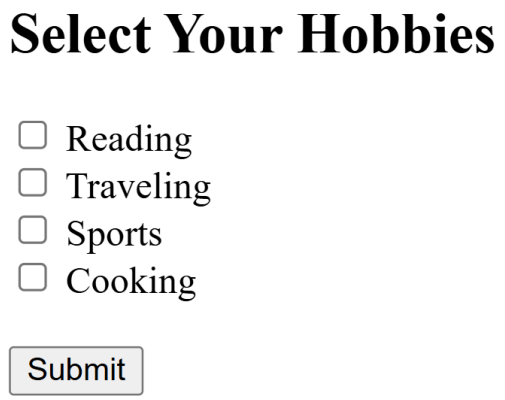
- PHP retrieves checkbox data using $_POST or $_GET.
- The values can be displayed or saved to a database.

Handling Checkboxes Data

- Example (hobbies.php)

```
<h2>Select Your Hobbies</h2>
<form action="result.php" method="post">
    <input type="checkbox" name="hobbies[]" value="Reading"> Reading<br>
    <input type="checkbox" name="hobbies[]" value="Traveling"> Traveling<br>
    <input type="checkbox" name="hobbies[]" value="Sports"> Sports<br>
    <input type="checkbox" name="hobbies[]" value="Cooking"> Cooking<br><br>

    <input type="submit" value="Submit">
</form>
```



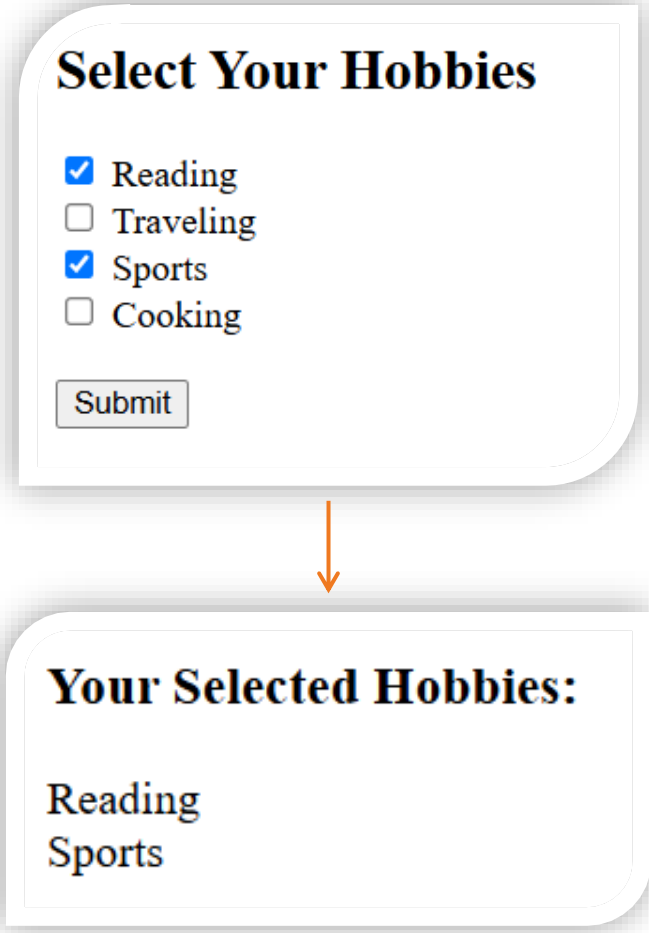
Receiving Checkboxes Data

- Example (result.php)

```php
<?php
$getData = $_POST['hobbies'];

echo "<h3>Your Selected Hobbies:</h3>";

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



Handling & Receiving Radio Buttons Data

Handling Radio Buttons Data

- Radio buttons let users choose only one option.
- Only one value is sent when the form is submitted.
- PHP processes the selected option from the form.

Receiving Radio Buttons Data

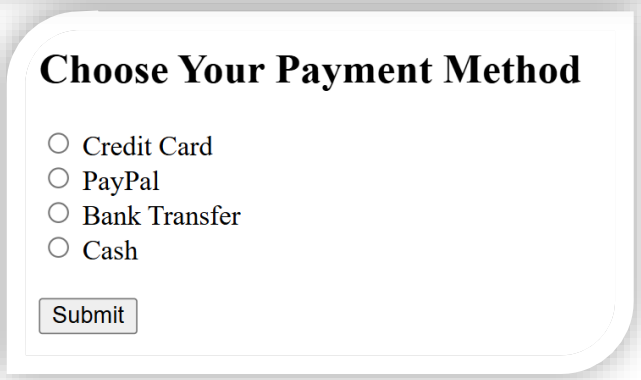
- PHP retrieves the selected radio value using $_POST or $_GET.
- The value can be displayed, validated, or used in decisions.

Handling Radio Buttons Data

- Example (payment.php)

```php
<h2>Choose Your Payment Method</h2>
<form action="result.php" method="post">
    <input type="radio" name="payment" value="Credit Card"> Credit Card<br>
    <input type="radio" name="payment" value="PayPal"> PayPal<br>
    <input type="radio" name="payment" value="Bank Transfer"> Bank Transfer<br>
    <input type="radio" name="payment" value="Cash"> Cash<br><br>

    <input type="submit" value="Submit">
</form>
```

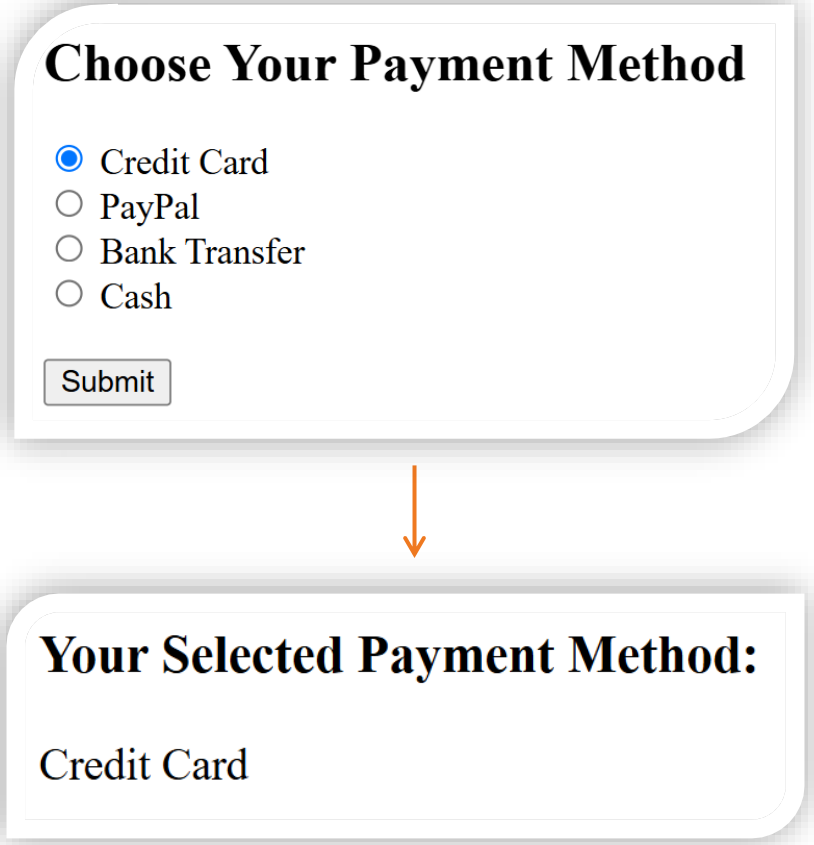


Receiving Radio Buttons Data

- Example (result.php)

```php
<h3>Your Selected Payment Method:</h3>

<?php
$paymentType = $_POST['payment'];
echo "$paymentType";
?>
```



Handling & Receiving Selection Data

Handling Selection Data

- Dropdowns let users choose one or more options.
- Commonly used for categories or lists.
- PHP handles the selected value when the form is submitted.

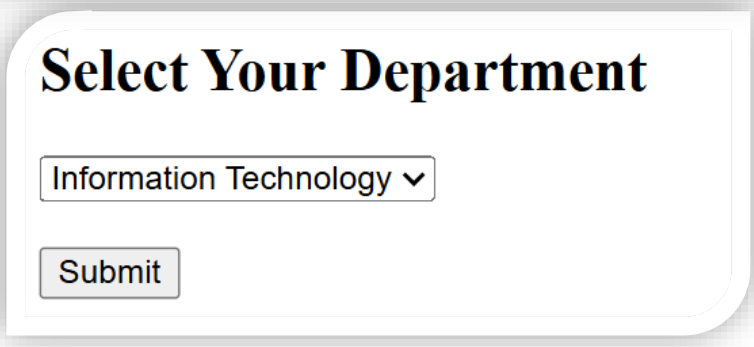
Receiving Selection Data

- PHP retrieves the selected option using $_POST or $_GET.
- The value can be displayed or processed as needed.

Handling Selection Data

- Example (firstPage.php)

```
<h2>Select Your Department</h2>
<form action="secondPage.php" method="POST">
    <select name="department">
        <option value="Information Technology">Information Technology</option>
        <option value="Cybersecurity">Cybersecurity</option>
        <option value="Artificial Intelligence">Artificial Intelligence</option>
        <option value="Finance">Finance</option>
    </select><br><br>
    <input type="submit" value="Submit">
</form>
```

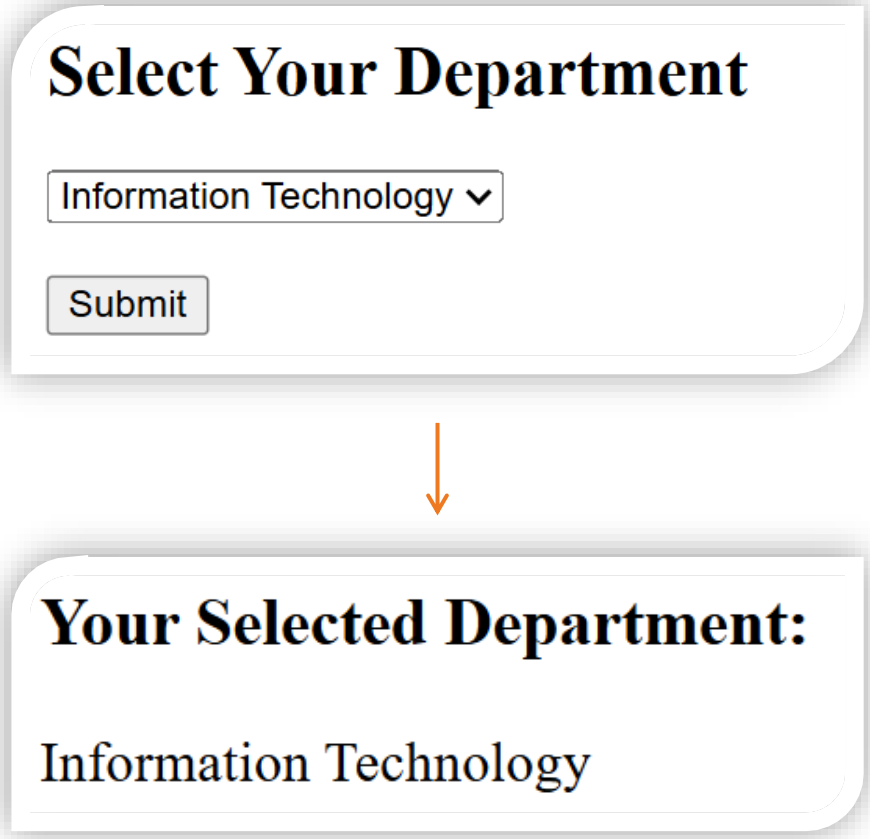


Receiving Selection Data

- Example (secondPage.php)

```
<h3>Your Selected Department:</h3>

<?php
$dep = $_POST['department'];
echo "$dep";
?>
```



Classwork

Simple Smartphone Review

Select a Smartphone :

Iphone Xs

Write your review :

it's a nice iPhone.

What are the best features :

☐ Camera ☐ Display ☐ Battery ☑ Performace ☑ Software

What are the worst features :

☑ Camera ☐ Display ☑ Battery ☐ Performace ☐ Software

Submit

Thank you for submiting your review

You have selected : Iphone Xs.

Your review text is : it's a nice.

Best Feature are :

Performace

Software

Worst Feature are :

Camera

Battery

3 stars

Best	5 star
Good	4 star
Nice	3 star
Fine	2 star
Bad	1 star

Lab Assesments and Next Session's Topic

Lab Assessments

- Lab Exercises

Next Session's Topic

- Form Validation, Cookies, and Sessions

References

- Tatroe, K., & MacIntyre, P. (2020). Programming PHP: Creating dynamic web pages (4th ed.). O'Reilly Media.
- Ullman, L. (2016). PHP for the web: Visual QuickStart guide (5th ed.). Peachpit Press.
- PHP Group. (n.d.). PHP: Hypertext Preprocessor — official documentation. Retrieved October 19, 2025, from https://www.php.net

Thank You!