(15 Batch)

1 A) What type of pages require what type of servers? Explain the servers briefly when rendering pages

Types of Pages and Required Servers

Type of Page	Required Server	Explanation
Static Pages	Web Server	A simple web server (like Apache or Nginx) is enough because static pages (HTML, CSS, images) don't change based on user input. They are delivered as-is.
Dynamic Pages	Web Server + Application Server	Dynamic pages (like PHP, JSP, or ASP.NET) change based on user input, database interaction, or backend logic. These need both a web server and an application server (like Apache with PHP, Tomcat for JSP, or IIS for ASP.NET).
Database-Driven Pages	Web Server + Application Server + Database Server	These pages fetch or store data from a database (like MySQL, PostgreSQL). The application server interacts with the database server to build the page dynamically.
Single Page Applications (SPA)	Web Server + API Server	SPAs (like those built in React, Angular) use a web server for the initial HTML/CSS/JS and communicate with an API server (backend) via AJAX or fetch for data updates.

Explanation of Key Servers

Web Server: Serves static content (HTML, CSS, JS). Example: Apache, Nginx.

Application Server: Handles business logic and dynamic content. Example: Tomcat (Java), Node.js (JS), PHP (scripting).

Database Server: Stores data and processes queries. Example: MySQL, MongoDB.

API Server: Offers data services to front-end apps (especially for SPAs and mobile apps). Example: Express (Node.js), Django REST, Flask.

Rendering Flow Example (Dynamic Page with PHP + MySQL):

Client requests a page (e.g., `profile.php`).

Web server (Apache) forwards it to PHP interpreter (application server).

PHP script fetches data from MySQL (database server).

PHP generates an HTML page and sends it back through the web server.

Client receives and views the rendered HTML page.

1 B) Why HTTP/1.0 is faster than HTTP/1.1?Explain briefly?

Why HTTP/1.0 is faster than HTTP/1.1? Explain briefly.

Note: This is a **misleading question. HTTP/1.1 is actually faster and more efficient than HTTP/1.0** due to several enhancements. Below is a brief comparison explaining why:

---

1. Connection Handling

HTTP/1.0: Opens and closes a **new TCP connection for each request** (non-persistent). This causes delay due to repeated connection setups and teardowns.

HTTP/1.1: Supports **persistent connections** (default) using **Keep-Alive**, allowing multiple requests and responses over a single TCP connection. This **reduces latency and improves speed**.

---

2. Request Pipelining

Not available in HTTP/1.0.

HTTP/1.1 allows **pipelining**, where multiple HTTP requests can be sent **without waiting** for the previous responses. This boosts performance over high-latency networks.

3. Chunked Transfer Encoding

HTTP/1.1 introduces **chunked transfer**, allowing a server to start sending data before knowing the total size.

Not supported in HTTP/1.0. This helps in faster rendering of large dynamic content.

---

4. Improved Caching Mechanisms

HTTP/1.1 adds headers like `Cache-Control`, `ETag`, and `If-Modified-Since`, enabling **more efficient caching** and reducing unnecessary data transfers.

5. Host Header Support

In **HTTP/1.0**, virtual hosting is not well supported; each IP typically hosts one site.

HTTP/1.1 makes the `Host:` header mandatory, allowing **multiple websites to be hosted on a single IP address** (virtual hosting).

Conclusion

The question should ideally ask **why HTTP/1.1 is faster.**
HTTP/1.1 provides better speed, reduced latency, and network efficiency through persistent connections, pipelining, chunked transfer, and caching—making it significantly faster and more reliable than HTTP/1.0.

2 A) What is DHTML? How can you write HTML comments? Give an example?

What is DHTML?

DHTML (Dynamic HTML) is not a language, but a combination of technologies used to create interactive and animated web pages. It includes:

HTML (structure)

CSS (styling)

JavaScript (behavior)

DOM (document access and manipulation)

Purpose: To change the content and style of a webpage **dynamically without reloading** the whole page.

How to Write HTML Comments?

HTML comments are written using:

```
<!-- This is a comment -->
```

✅ Example:

```
<p>This is visible text.</p>
<!-- This text is a comment and will not be displayed in the browser
-->
```

2 B) What is hypertext? Why does HTML call mark-up languages?

What is Hypertext?

Hypertext is text that contains **links (hyperlinks)** to other texts or documents. Clicking a hyperlink allows users to **navigate from one page or section to another** on the web.

✅ **Example:** A clickable word or phrase that takes you to another web page.

Why is HTML called a Markup Language?

HTML (HyperText Markup Language) is called a **markup language** because:

> It uses **tags (markups)** to define the **structure and layout** of a web document (like headings, paragraphs, images, links).
>
> The tags do **not perform any logic** like programming; they only **"mark up"** content to tell the browser how to display it.
>
> These tags wrap content and **"label"** it for browsers to understand its role.

✅ **Example:**

```
<p>This is a paragraph.</p>
```

2 C) What do you know about events in DHTML? Discuss onclick, onmouseover, and onmouseout event with example?

What are Events in DHTML?

In **DHTML**, an **event** is an action or occurrence that happens in the browser, such as clicking a button, moving the mouse, or typing on the keyboard. These events can trigger **JavaScript functions** to create dynamic behavior on a webpage.

Common events include: `onclick`, `onmouseover`, `onmouseout`, `onload`, `onkeydown`, etc.

---

1. onclick

Triggered when: The user clicks an HTML element.

Example:

```
<button onclick="alert('Button Clicked!')">Click Me</button>
```

---

2. onmouseover

Triggered when: The mouse pointer moves over an element.

Example:

```
<p onmouseover="this.style.color='blue'">Hover over me!</p>
```

---

3. onmouseout

Triggered when: The mouse pointer moves out of an element.

Example:

```
<p onmouseout="this.style.color='black'">Move mouse out from me!</p>
```

✅ Summary

Event	Description
onclick	Triggers when an element is clicked
onmouseover	Triggers when mouse enters an element
onmouseout	Triggers when mouse leaves an element

2 D) What is <img> tag HTML? Write HTML code so that you can show six images in two rows in a web page?

What is `<img>` Tag in HTML?

The `<img>` tag in HTML is used to **embed images** in a web page. It is an **empty tag** (no closing tag) and requires the `src` attribute to specify the image source.

✅ **Basic Syntax:**

```
<img src="image.jpg" alt="Image description">
```

HTML Code to Show 6 Images in 2 Rows:

```
<!DOCTYPE html>
<html>
<head>
  <title>Image Grid</title>
</head>
<body>
  <!-- First Row -->
  <img src="img1.jpg" width="150">
  <img src="img2.jpg" width="150">
  <img src="img3.jpg" width="150">
  <br><br>
  <!-- Second Row -->
  <img src="img4.jpg" width="150">
  <img src="img5.jpg" width="150">
  <img src="img6.jpg" width="150">
</body>
</html>
```

This will display **3 images per row** across **2 rows** on the web page.

3 A) What is HTML form? Can you design an HTML form for taking input the following data? Also, design an HTML table to show the data:(The Data is given in a table)

Roll--Gender--Session--Department
100--Male--2020-2021--CSE
200--Female--2020-2021--EEE
300--Male--2020-2021-STAT

3 A) What is an HTML Form?

An **HTML form** is used to **collect input from users** on a webpage. It can include elements like text boxes, radio buttons, checkboxes, dropdowns, and submit buttons.

✅ **Syntax:**

```
<form action="submit.php" method="post">
  <!-- form elements go here --></form>
```

✅ HTML Form to Take Input:

```
<h3>Student Information Form</h3>
<form>
  Roll:
      <input type="text" name="roll">
      <br><br>

  Gender:
  <input type="radio" name="gender" value="Male"> Male
  <input type="radio" name="gender" value="Female"> Female
      <br><br>

  Session:
  <input type="text" name="session" placeholder="e.g., 2020-2021">
      <br><br>

  Department:
  <select name="department">
    <option value="CSE">CSE</option>
    <option value="EEE">EEE</option>
    <option value="STAT">STAT</option>
```

```
  </select><br><br>

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

✅ HTML Table to Display Data:

```
<h3>Student Data Table</h3>
<table border="1">
  <tr>
    <th>Roll</th>
    <th>Gender</th>
    <th>Session</th>
    <th>Department</th>
  </tr>
  <tr>
    <td>100</td>
    <td>Male</td>
    <td>2020-2021</td>
    <td>CSE</td>
  </tr>
  <tr>
    <td>200</td>
    <td>Female</td>
    <td>2020-2021</td>
    <td>EEE</td>
  </tr>
  <tr>
    <td>300</td>
    <td>Male</td>
    <td>2020-2021</td>
    <td>STAT</td>
  </tr>
</table>
```

3 B) What do you know about Checkbox? How does it work in HTML form? Explain with proper example?

What is a Checkbox?

A **checkbox** is an input element in an HTML form that allows users to **select one or more options** from a set of choices. Each checkbox works independently and can be either **checked** or **unchecked**.

How Does It Work in an HTML Form?

When a checkbox is checked, its **value** is sent to the server upon form submission.

Multiple checkboxes with the same `name` attribute can be used to allow multiple selections.

Example of Checkbox in HTML Form:

```
<form>
  <p>Select your favorite fruits:</p>

  <input type="checkbox" name="fruits" value="Apple"> Apple<br>
  <input type="checkbox" name="fruits" value="Banana"> Banana<br>
  <input type="checkbox" name="fruits" value="Mango"> Mango<br><br>

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

If the user selects **Apple** and **Mango**, both values will be sent with the form.

3 C) What do you mean by block-level and inline-level elements in HTML? Give two examples of each of them?

Block-level Elements

Definition: Block-level elements are HTML elements that **take up the full width** of their parent container by default.

Layout Behavior: They **start on a new line**, pushing subsequent content to the next line.

Content: They can contain both **block-level and inline elements**.

Use: Mainly used for structuring the layout, sections, or larger parts of the page.

More Examples:

`<div>`, `<p>`, `<h1>`, `<ul>`, `<ol>`, `<li>`, `<section>`, `<header>`

Inline-level Elements

Definition: Inline elements take only as much width as their content needs.

Layout Behavior: They **do not start on a new line**; they flow within the current line alongside other inline elements.

Content: Usually contain text or smaller pieces of content.

Use: Used for formatting or styling parts of text or small pieces inside block-level elements.

More Examples:

`<span>`, `<a>`, `<img>`, `<strong>`, `<em>`, `<input>`, `<label>`

Additional Notes

Block-level elements can contain other block-level and inline elements, but **inline elements** cannot contain block-level elements.

You can **change the default behavior** of these elements using CSS `display` property:

e.g., `display: inline;` on a `<div>` makes it behave like an inline element.

`display: block;` on a `<span>` makes it behave like a block element.

Summary Table

Element Type	Default Behavior	Can Contain	Examples
Block-level	Starts on new line, full width	Block & inline elements	, <p>, <h1>,
Inline-level	Flows in line, width of content	Only inline elements	, <a>, <img>,

4 A) What do you know about PHP? What are the different ways to add PHP code in HTML?

What is PHP?

PHP (Hypertext Preprocessor) is a **server-side scripting language** used to create **dynamic and interactive web pages**.

It can **interact with databases**, manage sessions, handle forms, and generate HTML dynamically.

PHP code is executed on the **server**, and the output (usually HTML) is sent to the client's browser.

Ways to Add PHP Code in HTML:

Standard PHP Tags (Most Common):

```
<?php
  echo "Hello World!";?>
```

Short-open Tags (Must be enabled in `php.ini`):

```
<? echo "Hello World!"; ?>
```

Script Tag (Older style, rarely used):

```
<script language="php">
  echo "Hello World!";
</script>
```

ASP-style Tags (Deprecated in new versions):

```
<?php
  echo "Hello World!";
?>
```

4 B) Write the use of $_SESSION variable during sign-in of a website using HTML/PHP code.

Use of `$_SESSION` (3 marks):

`$_SESSION` stores **logged-in user data across pages** after `session_start()`.

After validating credentials, **set a session variable** (e.g., `$_SESSION['user_id']`) and redirect to protected pages.

On protected pages, **check** `isset($_SESSION['user_id'])`; for logout, call `session_destroy()`.

Minimal HTML/PHP example

```
/* login.php */
<?php
session_start();
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    if ($_POST['user'] === 'admin' && $_POST['pass'] === '1234') {
        $_SESSION['user'] = 'admin';   // store login state
        header('Location: dashboard.php');
        exit;
    } else {
    $err = "Invalid!"; }
}
?>
<form method="post">
```

```
  <input name="user">
<input type="password" name="pass">
  <button>Login</button>
</form><?= isset($err) ? $err : '' ?>

/* dashboard.php (protected) */
<?php
session_start();
if (!isset($_SESSION['user'])) {
      header('Location: login.php');
       exit; }
echo "Welcome, " . htmlspecialchars($_SESSION['user']);
echo '<a href="logout.php">Logout</a>';
?>


/* logout.php */
<?php
session_start();
session_unset();
session_destroy();
header('Location: login.php');
```

Key words to write: `session_start()`, store user info in `$_SESSION[...]`, check it to protect pages, `session_destroy()` on logout.

4 C) Write the functions of the following array functions with example.
i) preg_replace()
ii) array_key_exist()
iii) htmlspecialschars()

i) `preg_replace()`

Function: Performs a **search and replace** in a string using **regular expressions**.

Example:

```
$text = "Hello 123";
echo preg_replace("/[0-9]+/", "", $text);//
Output: Hello
```

---

ii) **`array_key_exists()`**

Function: Checks if a **specified key exists** in an array. Returns `true` if the key is found.

Example:

```
$arr = ["name" => "John", "age" => 25];
if (array_key_exists("name", $arr)) {
    echo "Key exists!";  // Output: Key exists!
}
```

---

iii) **`htmlspecialchars()`**

Function: Converts **special characters** (<, >, &, ") into **HTML entities** to prevent HTML injection.

Example:

```
echo htmlspecialchars("<b>Hello</b>");

// Output: &lt;b&gt;Hello&lt;/b&gt;
```

4 D) What is associate array? How can you put and print the values from the following table?

Emp-ID--Name--Designation--Salary
11--Shanon Rahman--AA--30,000
12--Rahima Khatun--BB--40,000

What is an Associative Array?

An **associative array** in PHP is an array that uses **named keys** instead of numeric indexes.

Each key maps to a specific value, like a dictionary.

Code to Put and Print Values from the Table:

```php
<?php
$employees = [
  11 => ["Name" => "Shanon Rahman", "Designation" => "AA", "Salary" =>
30000],

  12 => ["Name" => "Rahima Khatun", "Designation" => "BB", "Salary" =>
40000]
];
foreach ($employees as $id => $info) {
  echo "Emp-ID: $id, Name: {$info['Name']},
Designation: {$info['Designation']},
Salary: {$info['Salary']}<br>";
}
?>
```

5 A) What is MySQL database? How can you connect PHP code with MySQL database?

What is MySQL Database?

> **MySQL** is an **open-source relational database management system (RDBMS)** that stores data in **tables** using rows and columns.
>
> It uses **SQL (Structured Query Language)** for managing and retrieving data.

How to Connect PHP with MySQL?

You can connect PHP to a MySQL database using `mysqli_connect()` or **PDO**.

Example using `mysqli_connect()`:

```php
<?Php
$conn = mysqli_connect("localhost", "root", "", "mydb");
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}
echo "Connected successfully";
?>
```

Parameters: `host`, `username`, `password`, `database_name`.

5 B) How can we get the ID of the last inserted data in a MySQL database? Give an example?

How to get the last inserted ID (AUTO_INCREMENT) in MySQL

PHP (mysqli):

```
<?Php
$conn = mysqli_connect("localhost","root","","test");
$sql = "INSERT INTO users(name,email)
VALUES('Alice','a@example.com')";mysqli_query($conn, $sql);
$lastId = mysqli_insert_id($conn);   // or: $conn->insert_idecho
"Last ID = ".$lastId;?>
```

PHP (PDO):

```
$pdo = new PDO("mysql:host=localhost;dbname=test","root","");
$pdo->exec("INSERT INTO users(name,email)
VALUES('Bob','b@example.com')");echo "Last ID = " . $pdo-
>lastInsertId();
```

MySQL-side alternative:

```
SELECT LAST_INSERT_ID();
```

> Note: LAST_INSERT_ID() / mysqli_insert_id() / lastInsertId() return the last AUTO_INCREMENT value for the current connection, which makes it safe to use right after your INSERT.

5 C) Write a PHP program to upload an image file with proper validation.

Exam focus (4 marks): validate **size, type/MIME, upload errors, unique name, secure move**.

PHP + HTML (single file) — secure image upload

```
<?Php
// upload.phpif ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $maxSize  = 2 * 1024 * 1024; // 2 MB
    $allowed  = ['image/jpeg' => 'jpg', 'image/png' => 'png',
'image/gif' => 'gif'];

    if (!isset($_FILES['photo']) || $_FILES['photo']['error'] !==
UPLOAD_ERR_OK) {
        die("Upload error!");
    }

    // Size check
    if ($_FILES['photo']['size'] > $maxSize) {
        die("File too large. Max 2MB allowed.");
    }

    // MIME check (safer than extension)
    $finfo = new finfo(FILEINFO_MIME_TYPE);
    $mime  = $finfo->file($_FILES['photo']['tmp_name']);
    if (!array_key_exists($mime, $allowed)) {
        die("Invalid file type. Only JPG/PNG/GIF allowed.");
    }

    // Optional: extra check it's really an image
    if (@getimagesize($_FILES['photo']['tmp_name']) === false) {
        die("Not a valid image file.");
    }

    // Ensure upload dir exists
    $uploadDir = __DIR__ . '/uploads';
    if (!is_dir($uploadDir)) {
        mkdir($uploadDir, 0755, true);
```

```
    }

    // Generate a safe unique filename
    $newName = bin2hex(random_bytes(8)) . '.' . $allowed[$mime];
    $dest    = $uploadDir . '/' . $newName;

    if (!move_uploaded_file($_FILES['photo']['tmp_name'], $dest)) {
        die("Failed to move uploaded file.");
    }

    echo "Upload successful! Saved as: " . htmlspecialchars($newName);
}?>

<!DOCTYPE html>
<html>
<body>
<form method="post" enctype="multipart/form-data">
  <label>Select image (JPG/PNG/GIF, max 2MB):</label><br>
  <input type="file" name="photo" accept="image/*" required>
  <button type="submit">Upload</button>
</form>
</body>
</html>
```

Key points to write:

Use `enctype="multipart/form-data"` in the form.

Check `$_FILES['photo']['error']`, size, and MIME (`finfo`, `getimagesize`).

Generate a unique filename and use `move_uploaded_file()` to store it safely.

4 D) From the following code in PHP, find out the errors (if any) and write the reason. Write the corrected code.

<? php

$conn = mysqli_connect($server, $user, $password);

$sql = “select * from MyGuests;

$result = mysqli_query($conn, $sql);

If($result){

**echo “Query runs successfully
”;**

}else{

**Echo “Error: ”. $sql. “
”. mysqli_error($conn);**

}

?>

Errors and Reasons:

`<? php` → Should be `<?php` (no space allowed).

Quotes: “ ” (curly quotes) are invalid in PHP. Must use straight quotes `"` or `'`.

Missing closing quote: `“select * from MyGuests;` → Missing ending quote.

`If` **and** `Echo` → PHP keywords are **case-insensitive**, but best practice is lowercase (`if`, `echo`).

Semicolon in query: In SQL queries inside PHP, **semicolon `;` at the end is optional** but not needed inside the string.

Corrected Code:

```
<?Php
$conn = mysqli_connect($server, $user, $password);$sql = ″SELECT *
FROM MyGuests″;$result = mysqli_query($conn, $sql);
if ($result) {
    echo ″Query runs successfully<br>″;
} else {
    echo ″Error: ″ . $sql . ″<br>″ . mysqli_error($conn);
}
?>
```

6 A) What do you know about JavaScript? How do ‘for in’ and ‘for of’ loop work in JavaScript-Explain with proper examples.

What is JavaScript? (1 mark)

JavaScript is a **client-side scripting language** used to make web pages **dynamic, interactive, and functional**.

It can manipulate the **HTML DOM**, handle **events**, validate forms, and interact with servers via AJAX.

How `for...in` and `for...of` Loops Work (3 marks)

---

1. `for...in` Loop

Purpose: Iterates over the **keys (property names)** of an object or array.

Syntax:

```
for (let key in object) {
    // code
}
```

Example:

```
let person = { name: "John", age: 25 };for (let key in person)
{
  console.log(key + ": " + person[key]);
}// Output:// name: John// age: 25
```

---

2. `for...of` Loop

Purpose: Iterates over the **values** of an iterable (like an array, string, or Set).

Syntax:

```
for (let value of iterable) {
    // code
}
```

Example:

```
let colors = ["red", "green", "blue"];for (let color of colors)
{
  console.log(color);
}// Output:// red// green// blue
```

Key Difference:

`for...in` → Iterates over **keys** (or indexes in arrays).

`for...of` → Iterates over **values** directly.

6 B) What is client-side validation? Give an example program in DHTML to validate a form with the following fields using JavaScript?

Roll--Name--Dept

11--Shanon Rahman--AA

12--Rahima Khatun--BB

What is Client-side Validation? (1 mark)

Client-side validation is the process of **checking form data in the user's browser using JavaScript** before it is sent to the server. It helps **reduce server load** and provides **instant feedback** to users.

Example: DHTML Form Validation (Roll, Name, Dept) (2 marks)

```
<!DOCTYPE html><html><head>
  <script>
    function validateForm() {
      let roll = document.forms["myForm"]["roll"].value.trim();
      let name = document.forms["myForm"]["name"].value.trim();
      let dept = document.forms["myForm"]["dept"].value.trim();

      if (roll === "" || isNaN(roll)) {
        alert("Please enter a valid Roll number");
        return false;
```

```
    }
    if (name === "") {
      alert("Please enter the Name");
      return false;
    }
    if (dept === "") {
      alert("Please enter the Department");
      return false;
    }
    return true;
  }
</script></head><body>
<form name="myForm" onsubmit="return validateForm()">
  Roll: <input type="text" name="roll"><br><br>
  Name: <input type="text" name="name"><br><br>
  Dept: <input type="text" name="dept"><br><br>
  <input type="submit" value="Submit">
</form></body></html>
```

6 C) Write several ways of adding JavaScript code to DHTML web-page with example.

Several Ways to Add JavaScript to a DHTML Web Page

---

1. Inline JavaScript

Add JavaScript directly inside an HTML element's attribute (e.g., `onclick`).

```
<button onclick="alert('Clicked!')">Click Me</button>
```

---

2. Internal JavaScript

Write JavaScript inside a `<script>` tag within the HTML file.

```
<script>
  function greet() {
    alert('Hello from internal script!');
  }</script><button onclick="greet()">Greet</button>
```

---

3. External JavaScript File

Link an external `.js` file using the `<script src="filename.js"></script>` tag.

```
<script src="script.js"></script><button
onclick="externalGreet()">Greet</button>
```

script.js:

```
function externalGreet() {
  alert('Hello from external file!');
}
```

6 D) Write down the output of the following code and its reason.

console.log(100==''100'')

console.log(100===''100'')

Output:

```
console.log(100 == '100');   // true
console.log(100 === '100');  // false
```

Reason:

`==` (loose equality) compares **values after type conversion**. Here, string `'100'` is converted to number `100`, so they are equal → `true`.

`===` (strict equality) compares **both value and type without conversion**. Number `100` is **not** equal to string `'100'` → `false`.

7 A) What do you know about CSS? Can you write the basic syntax of CSS? Explain the syntax with proper example?

What is CSS?

CSS (Cascading Style Sheets) is a language used to **style and format** the appearance of HTML elements on a web page.

It controls layout, colors, fonts, spacing, and more to make webpages visually appealing and responsive.

Basic Syntax of CSS

```
selector {
  property: value;
}
```

Selector: Selects the HTML element(s) to style (e.g., `p`, `.class`, `#id`).

Property: The style attribute you want to change (e.g., `color`, `font-size`).

Value: The setting for the property (e.g., `red`, `16px`).

Example:

```
p {
  color: blue;
  font-size: 16px;
}
```

This CSS styles all `<p>` elements with blue text and font size 16 pixels.

7 B) How can you show blue colored text 'I am a student of 3rd year' in HTML inside a box placed in the middle of a page with a university logo as background?

HTML + CSS to show blue text inside a centered box with background logo

```
<!DOCTYPE html>
<html>
<head>
  <style>
    body, html {
      height: 100%;
      margin: 0;
      display: flex;
      justify-content: center;  /* center horizontally */
      align-items: center;      /* center vertically */
      background: url('university-logo.png') no-repeat center center;
      background-size: contain;
    }
    .box {
      background-color: white;
```

```
      padding: 20px 40px;
      border: 2px solid black;
      color: blue;
      font-size: 20px;
      font-weight: bold;
      text-align: center;
      box-shadow: 2px 2px 8px rgba(0,0,0,0.3);
    }
  </style>
</head>
<body>
  <div class="box">I am a student of 3rd year</div></body>
</html>
```

Explanation:

The **body** uses flexbox to center the `.box` both vertically and horizontally.

The **background** is the university logo centered and scaled.

The `.box` has white background with blue text and a border, forming a visible box.

7 C) Describe CSS box model with its required diagram and proper example program.

CSS Box Model

Every HTML element is considered as a rectangular box composed of **four parts**:

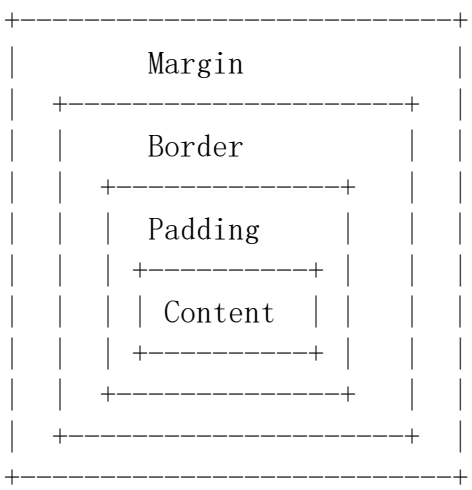
Content:
The actual content like text or images.

Padding:
Space between the content and the border. Increases the clickable or visible area.

Border:
A border surrounding the padding and content.

Margin:
The outermost space that separates the element from other elements.

Diagram of CSS Box Model:



Example Program:

```
<!DOCTYPE html>
<html>
<head>
<style>.box {
  width: 200px;
  padding: 20px;          /* space inside */
  border: 5px solid blue; /* border thickness & color */
  margin: 15px;           /* space outside */
  background-color: lightgray;
}
</style>
</head>
<body>
<div class="box">This is a CSS Box Model example.</div>
</body>
</html>
```

Here, the total width =
content width (200px) + padding (20px × 2) + border (5px × 2) + margin (15px × 2 outside spacing).

7 D) What do you know about ASP and ASP.NET?

ASP (Active Server Pages):

ASP is a **server-side scripting technology** by Microsoft to create **dynamic web pages** using scripting languages like VBScript or JScript.

It runs on the server and generates HTML sent to the client browser.

ASP.NET:

-

ASP.NET is a **modern web application framework** by Microsoft, part of the .NET platform, used to build **dynamic, scalable, and secure websites and web apps**.

It supports multiple languages (C#, VB.NET), offers better performance, and includes features like Web Forms, MVC, and Web API.

8 A) Compare between Web 2.0 and Web 3.0

Aspect	Web 2.0	Web 3.0
Definition	Known as the **social web**; emphasizes user-generated content, collaboration, and sharing.	Called the **semantic web** or **decentralized web**; uses AI and blockchain to create intelligent, autonomous, and user-centric experiences.
Data Ownership	Data is mostly stored and controlled by **centralized companies** (Facebook, Google, Amazon).	Data ownership is **distributed**, with users controlling their data through decentralized networks.
Technology Stack	Technologies like AJAX, RSS, blogs, social media platforms, wikis, and APIs.	Uses **Artificial Intelligence (AI), Machine Learning (ML), Blockchain, Smart Contracts, and Semantic Web** standards like RDF and OWL.
User Interaction	Web 2.0 focuses on **collaborative content creation** and **social**	Web 3.0 offers **personalized, context-aware, and smarter applications** that understand

Aspect	Web 2.0	Web 3.0
	networking.	user intent and provide autonomous services.
Monetization Model	Often based on advertising and centralized platforms controlling user data.	Emphasizes token-based economies, cryptocurrencies, and decentralized finance (DeFi) models.
Security and Privacy	Limited privacy; users often trust third parties to secure data.	Enhanced security through cryptography and decentralized networks, reducing reliance on intermediaries.
Examples	Facebook, YouTube, Wikipedia, Twitter.	Decentralized apps (dApps), Ethereum, IPFS, NFTs, AI-powered assistants.

8 B) Suppose you want to develop a web application? You want to give user specific view according to the user's choice. What type of page you will use?Describe that page with figure diagram?

1 Mark: Type of Page

Use a **Dynamic Web Page** to provide user-specific views based on their choices or data.

---

5 Marks: Detailed Description of Dynamic Web Page

Definition:
A dynamic web page is one whose content is **generated dynamically at runtime** rather than being fixed like a static page.

How it works:
When a user interacts (e.g., selects options, logs in), the request is sent to a **server** which runs scripts (like PHP, ASP.NET, Node.js).

The server **retrieves data** from a **database** or other sources relevant to the user.

Based on this data and user input, the server **generates HTML dynamically**, which is sent back to the browser for display.

This allows personalized views, such as showing different menus, content, or options tailored to the user's profile or choice.

Technologies used:

Server-side languages: PHP, ASP.NET, Python (Django/Flask), Java (JSP/Servlets)

Client-side scripts: JavaScript (AJAX, React, Angular) to update parts of the page dynamically without reload.

Advantages:

Personalized user experience

Easier content management

Real-time data display

Interaction with databases and APIs

Diagram: Dynamic Web Page Workflow

---

2 Marks: Summary

Dynamic pages **create customized content in response to user input or data**.

They improve user engagement by showing **personalized views**, e.g., dashboards, shopping carts, user profiles.

14 Batch

1 A) What do you know about front-end and back-end programming languages? Why are they necessary in present world?

Front-end Programming Languages

These are languages/technologies used to **design and build the user interface (UI)** of a website or web application (what users see and interact with).

Examples: **HTML**, **CSS**, **JavaScript**, React, Angular.

They handle **layout, styling, and interactivity** in the browser.

Back-end Programming Languages

These are languages used to **manage the server-side logic**, databases, and data processing.

Examples: **PHP**, **Python (Django/Flask)**, **Java**, **C#**, **Node.js**.

They handle tasks like **authentication, data storage, and business logic**.

Why Necessary in the Present World?

Front-end + Back-end = Full Web Functionality: Both are required to make modern websites interactive, dynamic, and user-friendly.

Back-end ensures **data security, performance, and logic**, while front-end ensures **design and user experience**.

Without them, we cannot build **e-commerce sites, social media, online services**, or any dynamic applications that are essential today.

Front-end Languages

Used for **UI design and user interaction** (e.g., **HTML, CSS, JavaScript**).

Back-end Languages

Used for **server logic, database handling** (e.g., **PHP, Python, Java, Node.js**).

Necessity

Both are required to build **dynamic, interactive, and secure modern websites/apps**.

1 B) What do you understand by HTTP, HTML and FTP ?

HTTP (HyperText Transfer Protocol):

HTTP is a **stateless application-level protocol** used for transferring hypertext (web pages) between a client (browser) and a web server.

It defines how requests (GET, POST) and responses are structured.

Example: When you type `https://example.com`, your browser sends an HTTP request to fetch the page.

HTML (HyperText Markup Language):

HTML is the **markup language** used to **design the structure** of web pages.

It consists of **tags and elements** like `<h1>`, `<p>`, `<a>` to create headings, paragraphs, and links.

HTML defines **what content appears**, but not how it looks (CSS handles styling).

FTP (File Transfer Protocol):

FTP is a **standard protocol for transferring files** between client and server over the Internet.

It supports **uploading, downloading, and managing files** on remote servers.

Tools like **FileZilla** or command-line FTP are used to connect to a web host.

Here's an **extended difference table with 4 key points** between **HTTP and FTP**:

Aspect	HTTP	FTP
Purpose	Transfers **web pages, images, and other web content.**	Transfers **files** (upload/download) between systems.
Port Used	Default **port 80** (or **443 for HTTPS**).	Uses **port 21 (control)** and **port 20 (data).**
Authentication	Typically does **not require login** (public access).	Usually requires **username and password.**
Usage Example	Accessing websites through a browser.	Uploading website files via FileZilla or command line.

1 C) Write the general format of URL and explain every part briefly ?

General Format of URL:

`protocol://hostname:port/path/filename?query#fragment`

Brief Explanation (1 line each):

protocol: Communication method (e.g., `http`, `https`).

hostname: Server or domain name (e.g., `www.example.com`).

port (optional): Server port (default: 80 for HTTP).

path/filename: Location of the resource (e.g., `/images/pic.jpg`).

query (optional): Data passed after `?` (e.g., `?id=5`).

fragment (optional): Page section after `#` (e.g., `#top`).

Example:

`https://www.example.com/products/item.html?id=10#reviews`

1 D) Who sets CSS standard? Write the differences between HTTP and HTTPS.

Who sets CSS standard? (1 mark)

W3C (World Wide Web Consortium) sets and maintains the **CSS standards** to ensure consistent styling across web browsers.

Differences between HTTP and HTTPS: (2 marks)

Aspect	HTTP	HTTPS
Security	Data is sent in **plain text.**	Data is **encrypted** using SSL/TLS.
Port	Uses **port 80** by default.	Uses **port 443** by default.
URL Format	http://example.com	https://example.com
Use Case	Suitable for non-sensitive data.	Ideal for secure transactions (e.g., banking).

2 A) What do you mean by HTML and DHTML ? Why do you use <!DOCTYPE> declaration in HTML pages?

What is HTML and DHTML? (1 mark)

HTML (HyperText Markup Language): A markup language used to create the **structure and content** of web pages using tags.

DHTML (Dynamic HTML): A combination of **HTML, CSS, JavaScript, and DOM** that allows web pages to be **dynamic and interactive** (e.g., animations, real-time updates).

Why use `<!DOCTYPE>` in HTML pages? (2 marks)

The `<!DOCTYPE>` declaration is placed at the very beginning of an HTML document to **inform the web browser about the HTML version and type** used in the page.

It triggers the browser's **standards mode**, ensuring that the page is rendered according to the official HTML specifications rather than in quirks mode, which emulates older, non-standard behaviors.

This helps in **maintaining consistent layout and behavior** across different browsers and devices, preventing rendering issues and improving compatibility.

2 B) What is markup language? Why does HTML call mark-up language?

What is a Markup Language?

A **markup language** is a set of rules and syntax using **tags or codes** to annotate or "mark up" text documents.

These tags provide **instructions to browsers or processors** about the structure, format, and presentation of the content, such as headings, paragraphs, links, lists, and images.

It does **not perform calculations or logic** but focuses on describing the content and how it should appear.

Why is HTML Called a Markup Language?

HTML (HyperText Markup Language) uses a predefined set of **tags enclosed in angle brackets**, like `<h1>`, `<p>`, `<a>`, to **markup or label the content**.

These tags **tell browsers how to display** different parts of a webpage — for example:

`<h1>` marks a heading,

`<p>` marks a paragraph,

`<a>` marks a hyperlink.

Because HTML's primary function is to **structure and present content by marking it up**, it is classified as a markup language.

Example:

```
<h1>This is a heading</h1>
<p>This is a paragraph.</p>
<a href="https://example.com">This is a link</a>
```

The tags `<h1>`, `<p>`, and `<a>` mark up the text to give it meaning and presentation instructions to the browser.

2 C) What do you know about events in DHTML? Disucss key-related events with example?

Events in DHTML

Events are actions or occurrences that happen in the browser, such as a user clicking a button, moving the mouse, or pressing a key.

DHTML uses **JavaScript to respond to these events**, making web pages interactive and dynamic.

Common event types include **mouse events**, **keyboard events**, **form events**, and **window events**.

Key-related Events in DHTML

`onkeydown`

Triggered when a key is **pressed down**.

Example:

```
<input type="text" onkeydown="console.log('Key down: ' + event.key)">
```

`onkeyup`

Triggered when a key is **released**.

Example:

```
<input type="text" onkeyup="console.log('Key up: ' + event.key)">
```

`onkeypress`

Triggered when a key is **pressed and produces a character value**. (Note: Deprecated in modern browsers, use `onkeydown` and `onkeyup` instead.)

Example:

```
<input type="text" onkeypress="console.log('Key pressed: ' + event.key)">
```

Summary:

Key events allow detecting and responding to keyboard actions in real-time, useful for form validation, shortcuts, or games.

These events improve user interaction by allowing dynamic responses to key presses.

2 D) Discuss different value used for TYPE attribute of <INPUT>tag

Different Values for `type` Attribute of `<input>` Tag

The `type` attribute specifies the **kind of input control** to display. Common values include:

text

For single-line text input.

Example: `<input type="text">`

password

For entering passwords; input characters are masked.

Example: `<input type="password">`

radio

For selecting **one option** from a set of choices (radio buttons).

Example: `<input type="radio" name="gender" value="male">`

checkbox

For selecting **multiple options** independently (checkboxes).

Example: `<input type="checkbox" name="subscribe">`

submit

A button to **submit** the form.

Example: `<input type="submit" value="Send">`

button

A clickable button that can trigger JavaScript actions.

Example: `<input type="button" value="Click Me">`

email

For email input with basic validation.

Example: `<input type="email">`

number

For numeric input with up/down controls.

Example: `<input type="number" min="1" max="10">`

Summary:

The `type` attribute controls the behavior and appearance of input fields, enabling different types of user input.

3 A) What is HTML Table? Write a DHTML code to design the following table:

Dept. Name--No. of Students--No. of Teachers

CSE--200--15

EEE--160--12

STAT--160--10

What is an HTML Table?

An **HTML table** is a structured set of rows and columns used to display data in a grid format on a webpage.

It is created using `<table>`, `<tr>` (table row), `<th>` (table header), and `<td>` (table data) tags.

DHTML Code to Design the Given Table:

```
<!DOCTYPE html>
<html>
<head>
  <title>Department Table</title>
  <script>
```

```
    // DHTML part could be adding some dynamic effect if needed
    function highlightRow(row) {
      row.style.backgroundColor = 'lightyellow';
    }
    function resetRow(row) {
      row.style.backgroundColor = '';
    }
  </script>
</head>
<body>
<table border="1" cellpadding="8" cellspacing="0" style="border-
collapse: collapse; width: 50%;">
  <tr>
    <th>Dept. Name</th>
    <th>No. of Students</th>
    <th>No. of Teachers</th>
  </tr>
  <tr onmouseover="highlightRow(this)" onmouseout="resetRow(this)">
    <td>CSE</td>
    <td>200</td>
    <td>15</td>
  </tr>
  <tr onmouseover="highlightRow(this)" onmouseout="resetRow(this)">
    <td>EEE</td>
    <td>160</td>
    <td>12</td>
  </tr>
  <tr onmouseover="highlightRow(this)" onmouseout="resetRow(this)">
    <td>STAT</td>
    <td>160</td>
    <td>10</td>
  </tr>
</table>
</body>
</html>
```

Explanation:

The table headers are created with `<th>`.

Data rows are created with `<tr>` and data cells with `<td>`.

The JavaScript functions `highlightRow` and `resetRow` add simple DHTML effects to highlight rows on mouse hover.

3 B) What are CSS selectors? How can you write a text “I am a student of CSE” in red color inside a box placed in the middle of a page with dotted 2px width border?

What are CSS Selectors?

CSS selectors are **patterns used to select HTML elements** you want to style.

They tell the browser **which elements to apply CSS rules to** (e.g., by tag name, class, id, attribute).

Example Code to Show Text in Red Inside a Centered Box with Dotted Border:

```
<!DOCTYPE html>
<html>
<head>
  <style>
    .center-box {
      width: 300px;
      margin: 100px auto;          /* centers the box horizontally */
      border: 2px dotted black;      /* dotted border */
      padding: 20px;
      text-align: center;
      color: red;                    /* red text */
      font-size: 20px;
      font-weight: bold;
    }
  </style>
</head>
<body>
<div class="center-box">I am a student of CSE</div>
</body>
</html>
```

Explanation:

`.center-box` is a **class selector** targeting the `<div>`.

`margin: 100px auto;` centers the box horizontally with some vertical spacing.

`border: 2px dotted black;` creates a dotted border of 2px width.

`color: red;` colors the text red.

3 C) Discuss different ways of adding CSS in a DHTML page?

Different Ways to Add CSS in a DHTML Page:

Inline CSS:

CSS is added directly inside an HTML element using the `style` attribute.

Example: `<p style="color: red;">Text</p>`

Internal CSS:

CSS is written inside a `<style>` tag within the `<head>` section of the HTML page.

Example:

```
<style>
  p { color: blue; }</style>
```

External CSS:

CSS is written in a separate `.css` file and linked using the `<link>` tag inside the `<head>`.

Example:

```
<link rel="stylesheet" href="styles.css">
```

3 D) Narrate CSS box model with its required diagram and proper example program?

CSS Box Model

Every HTML element is rendered as a **rectangular box** consisting of four parts:

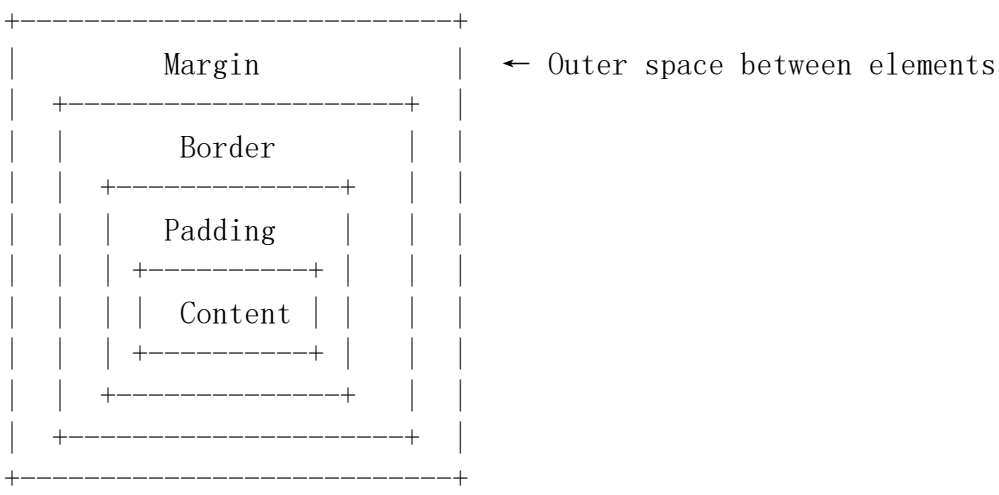
Content: The actual text, image, or media inside the box.

Padding: Space between the content and the border, inside the box.

Border: The edge surrounding the padding and content.

Margin: The outer space that separates the box from other elements.

Diagram:



Example Program:

```
<!DOCTYPE html>
<html>
<head>
<style>
.box {
  width: 200px;                /* content width */
  padding: 20px;               /* inside space */
  border: 5px solid blue;      /* border */
  margin: 15px;                /* outside space */
  background-color: lightgray; /* background color */
}
</style>
</head>
<body>
<div class="box">This is a CSS Box Model example.</div>
</body>
</html>
```

Explanation:

The total width of `.box` =
content width (200px) + padding (20px × 2) + border (5px × 2) + margin (15px × 2 outside spacing).

This model helps control spacing and layout precisely on a webpage.

4 A) What is PHP? “PHP is one of the popular web programming languages”- Explain ?

What is PHP?

PHP (Hypertext Preprocessor) is a **server-side scripting language** designed to create **dynamic and interactive web pages**.

It is embedded within HTML and runs on the server, generating HTML code sent to the client browser.

“PHP is one of the popular web programming languages” - Explanation

PHP is widely used because it is **easy to learn, open-source, and platform-independent**.

It supports **database connectivity**, enabling dynamic content like user logins, forums, and e-commerce sites.

Many popular websites and platforms (like WordPress) are built using PHP, making it very popular among web developers.

4 B) Write the difference between following in PHP:

1) include() and require()

2) Session and Cookie

1) Difference between `include()` and `require()`

Aspect	include()	require()
Error handling	Generates a **warning** if the file is missing, but the script **continues** running.	Generates a **fatal error** if the file is missing, and the script **stops execution**.
Use case	Used when the file is **optional** or non-critical.	Used when the file is **mandatory** for the application to work.
Behavior in loops	Can be used multiple times to include the file multiple times.	Same as include, but script halts on failure.

Aspect	include()	require()
Performance	Slightly slower than require() because it checks for file presence and issues warning.	Slightly faster since failure halts immediately.

2) Difference between Session and Cookie

Aspect	Session	Cookie
Storage Location	Stored on the **server** in temporary files or memory.	Stored on the **client's browser** as small text files.
Security	More secure because data is stored server-side.	Less secure; users can see, modify, or delete cookies.
Lifetime	Lasts until browser is closed or session expires.	Can persist for a specified time (days, months).
Data Size	Can store larger data on the server.	Limited size (~4KB) per cookie.
Use Case	Storing sensitive data like login status, user preferences during a session.	Storing preferences or tracking info over time (e.g., language, cart items).

4 C) What is the function of the following array function with example:

A) preg_replace()

B) array_keys()

C) count()

D) in_array()

E) krsort()

A) `preg_replace()`

Function: Performs a **regular expression search and replace** on a string.

Example:

```
$text = "Hello World!";
$new_text = preg_replace("/World/", "PHP", $text);
echo $new_text;
// Output: Hello PHP!
```

B) **`array_keys()`**

Function: Returns all the **keys** from an array.

Example:

```
$arr = ['a' => 1, 'b' => 2];
$keys = array_keys($arr);print_r($keys);
// Output: Array ( [0] => a [1] => b )
```

C) **`count()`**

Function: Returns the **number of elements** in an array (or properties in an object).

Example:

```
$arr = [10, 20, 30];echo count($arr);
// Output: 3
```

D) **`in_array()`**

Function: Checks if a **value exists** in an array, returns `true` or `false`.

Example:

```
$arr = [1, 2, 3];if (in_array(2, $arr)) {
    echo "Found";  // Output: Found
}
```

E) **`krsort()`**

Function: Sorts an array by **keys in descending order**, maintaining key-to-value associations.

Example:

```
$arr = ['b' => 2, 'a' => 1, 'c' => 3];
krsort($arr);print_r($arr);
// Output:// Array ( [c] => 3 [b] => 2 [a] => 1 )
```

5 A) Write the function of $_SERVER[‘DOCUMENT_ROOT’] and $_SERVER[‘HTTP_HOST’] and $_SERVER[‘PHP_SELF’] in PHP?

Functions of PHP `$_SERVER` Variables

`$_SERVER['DOCUMENT_ROOT']`

Returns the **root directory path** of the server where the current script is executed.

Example:

```
echo $_SERVER['DOCUMENT_ROOT'];
// Output: /var/www/html (depends on server configuration)
```

`$_SERVER['HTTP_HOST']`

Returns the **host name** from the current request header (e.g., the domain name).

Example:

```
echo $_SERVER['HTTP_HOST'];// Output: www.example.com
```

`$_SERVER['PHP_SELF']`

Returns the **file name of the currently executing script** relative to the root directory.

Example:

```
echo $_SERVER['PHP_SELF'];// Output: /folder/index.php
```

5 B) What are superglobal variables in PHP? Write the names and functions of them?

What are Superglobal Variables in PHP?

Superglobals are **built-in global arrays** in PHP that are **accessible from anywhere** in a script (inside or outside functions) **without using** `global` **keyword**.

They store information about **HTTP requests, server environment, sessions, forms, etc.**

Names and Functions of Superglobal Variables:

`$_GET` – Stores data sent via **URL query string** (HTTP GET method).

`$_POST` – Stores data sent via **HTML form (POST method)**.

`$_REQUEST` – Contains data from **both GET and POST** requests.

`$_SERVER` – Contains **server and execution environment information**.

`$_FILES` – Manages **file uploads**.

`$_SESSION` – Stores **session variables** on the server.

`$_COOKIE` – Stores **cookie values** set in the client browser.

`$_ENV` – Contains **environment variables**.

`$GLOBALS` – Holds all global variables in a script.

C) What is an array in PHP? Discuss different types of arrays in PHP with proper example?

What is an Array in PHP?

An **array** in PHP is a special variable that can **store multiple values** under a single name, and the values can be accessed using **keys or indexes**.

Types of Arrays in PHP:

1) Indexed Array

Stores values with numeric indexes (starting from 0).

Example:

```
$colors = ["Red", "Green", "Blue"];echo $colors[0];  // Output: Red
```

---

2) Associative Array

Stores values with **named keys** instead of numeric indexes.

Example:

```
$student = ["Name" => "Shanon", "Dept" => "CSE"];echo $student["Dept"];  // Output: CSE
```

---

3) Multidimensional Array

An array containing **one or more arrays** (like a table or matrix).

Example:

```
$students = [
    ["Name" => "Shanon", "Dept" => "CSE"],
    ["Name" => "Rahima", "Dept" => "EEE"]
];
echo $students[1]["Dept"];
 // Output: EEE
```

5 D) What is meant by the code "<?=$err;"? Can you differentiate POST and GET method?

What is meant by the code `<?=$err;?>`?

`<?=$err;?>` is a **short form of** `<?php echo $err; ?>` in PHP.

It is used to **directly output the value of the variable** `$err` to the webpage.

Difference between POST and GET Methods:

GET	POST
Sends data as part of the URL (visible).	Sends data in the HTTP request body (hidden).

GET	POST
Limited data size (about 2048 characters).	No size limitation (can send large data).
Less secure (data visible in address bar).	More secure (data not shown in URL).
Example: page.php?name=John.	Used for form submissions (login, etc.).

6 A) What do you know about scripting language? Write an example DHTML program with a button to show current date/time.

What is a Scripting Language? (1 mark)

A scripting language is a **lightweight programming language** used to automate tasks and control the behavior of web pages or applications.

It is usually **interpreted, not compiled** (e.g., JavaScript, PHP).

In web development, scripting languages add **interactivity and dynamic features** to web pages.

Example DHTML Program with a Button to Show Current Date/Time: (2 marks)

```
<!DOCTYPE html>
<html>
<head>
  <title>Show Date/Time</title>
  <script>
    function showDateTime() {
      document.getElementById("result").innerHTML = new Date();
    }
  </script>
</head>
<body>
  <button onclick="showDateTime()">Show Current Date/Time</button>
  <p id="result"></p></body></html>
```

6 B) What is client-side validation? Give an example program in DHTML to validate a form with the following fields using JavaScript? Emp-Id--Name--Designation--Salary

11-Shanon Rahman--AA--30,000

12--Rahima Khatun--BB--40,000

What is Client-Side Validation? (1 mark)

> **Client-side validation** is the process of **checking user input in the browser (using JavaScript or DHTML)** before sending it to the server.
>
> It improves **user experience** by providing instant feedback and reducing server load.

Example DHTML Program for Form Validation: (4 marks)

```
<!DOCTYPE html>
<html>
<head>
  <title>Employee Form Validation</title>
  <script>
    function validateForm() {
      let empId = document.getElementById("empId").value;
      let name = document.getElementById("name").value;
      let designation = document.getElementById("designation").value;
      let salary = document.getElementById("salary").value;

      if (empId == "" || name == "" || designation == "" || salary ==
"") {
        alert("All fields must be filled out");
        return false;
      }

      if (isNaN(empId)) {
        alert("Emp-Id must be a number");
        return false;
      }

      if (isNaN(salary)) {
        alert("Salary must be a number");
        return false;
```

```
      }

      return true; // Form is valid
    }
  </script></head><body>
  <h2>Employee Details Form</h2>
  <form onsubmit="return validateForm()">
    Emp-Id: <input type="text" id="empId"><br><br>
    Name: <input type="text" id="name"><br><br>
    Designation: <input type="text" id="designation"><br><br>
    Salary: <input type="text" id="salary"><br><br>
    <input type="submit" value="Submit">
  </form>
</body>
</html>
```

6 C) Write sevaral ways of adding JavaScript code to a DHTML webpage with example?

Ways of Adding JavaScript to a DHTML Webpage:

There are **three main ways** to add JavaScript to a webpage:

---

1) Inline JavaScript (Using HTML attributes)

Code is directly added inside HTML tags using attributes like `onclick`, `onmouseover`, etc.
Example:

```
<button onclick="alert('Hello!')">Click Me</button>
```

---

2) Internal JavaScript (Inside `<script>` tag)

Code is written inside `<script>` tags in the **same HTML file**, usually inside the `<head>` or at the end of `<body>`.
Example:

```
<script>
  function sayHello() {
    alert("Hello, World!");
  }</script><button onclick="sayHello()">Click Me</button>
```

---

3) External JavaScript (Using `.js` file)

JavaScript code is placed in a **separate** `.js` **file** and linked to the HTML file using `<script src="filename.js"></script>`.
Example:

```
<script src="script.js"></script>
```

script.js

```
alert("Hello from external file!");
```

---

4) Inside Event Handlers (Alternative Inline Method)

JavaScript can also be added directly to event attributes in HTML elements.
Example:

```
<p onmouseover="alert('Mouse Over Event!')">Hover over me</p>
```

7 A) What do you know about firewall? Why is it necessary in network communication?

What is a Firewall?

A **firewall** is a network security system (hardware or software) that **monitors and controls incoming and outgoing network traffic** based on predefined security rules.

It acts as a **barrier between a trusted internal network and untrusted external networks (like the internet).**

Why is it Necessary in Network Communication?

Protects against **unauthorized access, hackers, and malware**.

Ensures **secure data transmission** by filtering harmful traffic.

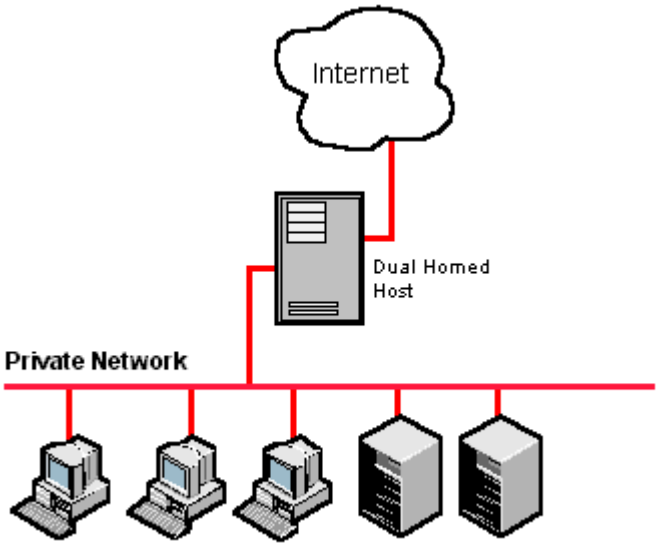
Prevents attacks such as **DDoS, viruses, or phishing attempts** by blocking suspicious connections.

7 B) Write the differences between Bastion host and Dual-homed host computer?

Differences between Bastion Host and Dual-Homed Host:

Bastion Host	Dual-Homed Host
A **hardened computer** specifically designed to withstand attacks and provide **secure services (e.g., proxy, firewall)**.	A host that has **two network interfaces (NICs)**, usually one connected to the internal network and the other to the external network (Internet).
Typically used as a **public-facing server** in a DMZ (Demilitarized Zone).	Used as a **router or gateway** to separate internal and external networks.
Focuses on **application-level security**.	Focuses on **network-level isolation**.

7 C) Discuss dual-homed host architecture with the required diagram?



Dual-Homed Host Architecture

A **dual-homed host** is a computer with **two network interfaces (NICs)** — one connected to the **internal network (LAN)** and the other to the **external network (Internet)**.

It acts as a **gateway or firewall**, ensuring that traffic between the networks is controlled and secure.

The host typically runs firewall software to filter unauthorized access.

Direct packet forwarding between the two interfaces is **disabled**, which adds a layer of security.

Key Features:

Two Network Interfaces: One for LAN and one for the Internet.

Security Control: Filters traffic using firewall rules.

Isolation: Prevents direct access to the internal network.

Used in DMZ setups to protect internal servers.

Example Use Case:

A company server that connects the corporate network to the internet but controls access using firewall software like **iptables**.

7 D) What do you know about web page transition? Classify and describe each of them with example?

What is Web Page Transition?

Web page transition refers to the **effect or animation that occurs when a user moves from one web page to another**.

It enhances user experience by making navigation smoother and visually appealing.

Classification of Web Page Transitions:

None (No Transition):

The page changes instantly without any animation.

Example: Clicking a link that loads a new page immediately.

Fade Transition:

The current page fades out and the new page fades in smoothly.

Example: Using CSS or JavaScript to gradually change page opacity during navigation.

Slide Transition:

The new page slides into view from left, right, top, or bottom.

Example: Mobile apps or SPA frameworks like React use slide effects for page changes.

Example:

Using CSS to create a fade effect on page load:

```
css
CopyEdit
body {
  animation: fadeIn 1s ease-in;
}
@keyframes fadeIn {
  from { opacity: 0; }
  to { opacity: 1; }
}
```

8 A) What do you know about web server? Narrate common features of a web server?

What is a Web Server? (1 mark)

A **web server** is a computer or software that **stores, processes, and delivers web pages** to users via the Internet using the HTTP/HTTPS protocol.

Common Features of a Web Server: (3 marks)

HTTP/HTTPS Support:

Handles requests and responses using HTTP/HTTPS protocols securely.

Static and Dynamic Content Delivery:

Serves static files (HTML, CSS, images) and runs server-side scripts (PHP, ASP.NET) to generate dynamic content.

Logging and Monitoring:

Keeps logs of requests, errors, and user activities for analysis and troubleshooting.

Security Features:

Supports authentication, SSL/TLS encryption, and access control to protect data and restrict unauthorized access.

8 B) What do you know about cross site scripting and DoS attacks? Discuss them with example?

Cross-Site Scripting (XSS):

XSS is a **web security vulnerability** where attackers inject malicious scripts into trusted websites.

These scripts run in the browsers of users visiting the affected page, stealing cookies, session tokens, or manipulating content.

Example: An attacker injects `<script>alert('Hacked!');</script>` into a comment box; when other users view the comment, the script runs.

Denial of Service (DoS) Attack:

A DoS attack **aims to make a website or service unavailable** by overwhelming it with excessive traffic or resource requests.

This causes legitimate users to be denied access due to server overload or crash.

Example: Sending thousands of fake requests per second to a website, causing it to slow down or crash.

8 C) What is load balancer in web server? How can you manage network traffic for anti-overload techniques in web server?

What is a Load Balancer in a Web Server?

A **load balancer** is a device or software that **distributes incoming network traffic across multiple servers** to ensure no single server becomes overwhelmed.

It helps improve **performance, reliability, and availability** of web services by balancing the load.

Managing Network Traffic for Anti-Overload Techniques:

Traffic Distribution: Distribute requests evenly among servers using algorithms like Round Robin or Least Connections.

Rate Limiting: Limit the number of requests from a single IP to prevent overload or abuse.

Caching: Use caching to serve repeated requests quickly without hitting the server repeatedly.

Connection Throttling: Control the number of simultaneous connections to protect server resources.

8 D) What do you know about CGI script? Why is it necessary?

What is a CGI Script?

CGI (Common Gateway Interface) script is a program that runs on a web server to **generate dynamic content** by processing user input from web forms or URLs.

It acts as an interface between the web server and external applications or databases.

Why is it Necessary?

Enables web servers to **handle user requests dynamically** instead of just serving static files.

Allows creation of **interactive websites**, such as processing form data, generating customized pages, or running server-side programs.

13 Batch

1 A) What is web programming? “Web programming is a blessing in the modern world”-explain?

What is Web Programming?

Web programming is the process of **writing code to create websites and web applications** that run on web browsers or servers.

It involves languages like **HTML, CSS, JavaScript, PHP, Python, and others** to build interactive, dynamic web pages.

“Web programming is a blessing in the modern world” – Explanation

It **connects people globally**, enabling communication, education, shopping, and entertainment online.

Facilitates **automation and digitization** of services, making daily tasks faster and easier.

Supports **business growth and innovation** by allowing companies to reach customers worldwide.

Enables development of **useful tools and platforms** like social media, online banking, and e-learning.

1 B) What do you know about Telnet, FTP and E-commerce.

Telnet:

Network protocol for **remote command-line access** to another computer over a network.

Provides a **command-line interface** for users to log in and execute commands remotely.

Useful for **remote system administration and troubleshooting**.

Less secure as it transmits data, including passwords, in **plain text**.

FTP (File Transfer Protocol):

Standard protocol to **transfer files between client and server** over the Internet or network.

Supports **uploading, downloading, renaming, deleting, and managing files remotely**.

Used for **website maintenance, data backup, and file sharing**.

Not encrypted by default; secure alternatives like **SFTP** are preferred.

E-commerce:

Buying and selling **goods and services online** through websites, apps, or digital platforms.

Includes **online shopping, electronic payments, online auctions, and digital marketing**.

Enables **global reach, 24/7 availability, personalized experiences, and fast transactions**.

Examples: Amazon, eBay, Alibaba.

1 C) Define URL? Write the general format of URL and explain every part briefly ?

What is a URL? (1 mark)

URL (Uniform Resource Locator) is the **address used to locate a resource (like a web page) on the Internet**.

General Format of a URL: (3 marks)

```
protocol://username:password@host:port/path?query_string#fragment_id
```

Explanation of Each Part:

Protocol: Specifies the communication protocol to be used (e.g., `http`, `https`, `ftp`).

Username:Password (optional): User credentials for accessing restricted resources.

Host: The domain name or IP address of the server hosting the resource (e.g., `www.example.com`).

Port (optional): The network port number to connect to (default for HTTP is 80).

Path: The specific location or file on the server (e.g., `/folder/page.html`).

Query String (optional): Contains parameters passed to the resource, starting with `?` (e.g., `?id=10&sort=asc`).

Fragment ID (optional): Refers to a specific section within the resource, starting with `#` (e.g., `#section2`).

1 D) What do you know about Internet Telephony? Write the name of some well-known tools used for Internet Telephony?

What is Internet Telephony?

Internet Telephony, also called **Voice over Internet Protocol (VoIP)**, is the technology that allows **voice communication and multimedia sessions over the Internet** instead of traditional phone lines.

It converts voice signals into digital data packets and transmits them over IP networks, enabling cheaper and flexible communication.

Well-known Tools for Internet Telephony:

Skype

Zoom

Google Meet

WhatsApp Voice Call

Microsoft Teams

2 A) Which HTML tag is used to display the data in tabular form?

HTML Tag Used to Display Data in Tabular Form

The `<table>` tag is used to create a table in HTML.

Common Tags Used Inside `<table>`:

`<tr>` **(Table Row):** Defines a row in the table.

`<th>` **(Table Header):** Defines a header cell in the table, usually bold and centered.

`<td>` **(Table Data):** Defines a standard cell containing data.

`<caption>` **(Optional):** Adds a title or caption to the table.

Basic Example:

```
<table border="1">
  <caption>Student Details</caption>
  <tr>
    <th>Roll</th>
    <th>Name</th>
    <th>Department</th>
  </tr>
  <tr>
    <td>11</td>
    <td>Shanon Rahman</td>
    <td>CSE</td>
  </tr>
  <tr>
    <td>12</td>
    <td>Rahima Khatun</td>
    <td>EEE</td>
  </tr>
</table>
```

2 B) What are some common lists that are used when designing a page? Explain with example?

Common Types of Lists Used in Web Page Design

Ordered List (`<ol>`)

Displays a list with **numbered items**.

Used when the order or sequence matters.

Example:

```
<ol>
  <li>First item</li>
  <li>Second item</li>
  <li>Third item</li></ol>
```

Unordered List (`<ul>`)

Displays a list with **bulleted items**.

Used when the order is not important.

Example:

```
<ul>
  <li>Apple</li>
  <li>Banana</li>
  <li>Cherry</li></ul>
```

Definition List (`<dl>`)

Displays a list of terms and their definitions.

Contains `<dt>` (definition term) and `<dd>` (definition description).

Example:

```
<dl>
  <dt>HTML</dt>
  <dd>HyperText Markup Language</dd>
  <dt>CSS</dt>
  <dd>Cascading Style Sheets</dd></dl>
```

2 C) Explain the layout of HTML?

Layout of HTML

The layout of an HTML document is structured in a **hierarchical way** and includes several key parts:

DOCTYPE Declaration

Declares the HTML version being used (e.g., `<!DOCTYPE html>` for HTML5).

Helps browsers render the page correctly.

`<html>` **Tag**

The root element that wraps the entire HTML document.

`<head>` **Section**

Contains meta-information about the document such as:

`<title>`: Title of the webpage shown in the browser tab.

`<meta>` tags: Metadata like character set, author, viewport settings.

Links to CSS files or scripts.

`<body>` **Section**

Contains the **visible content** of the webpage such as text, images, links, tables, forms, etc.

Basic HTML Layout Example:

```
<!DOCTYPE html>
<html>
<head>
  <title>Sample Page</title>
  <meta charset="UTF-8"></head><body>
  <h1>Welcome to My Website</h1>
  <p>This is a sample paragraph.</p>
</body>
</html>
```

3 A) Write the differences between the following terms:

1) HTML and DHTML

2) Checkboxes and RadioButtons

3) Transparency filter and Wipe filter

1) HTML vs DHTML

HTML (HyperText Markup Language)	DHTML (Dynamic HTML)
Static markup language to create web page structure.	Combination of HTML, CSS, and JavaScript for dynamic pages.
Pages are **fixed** and do not change without reload.	Enables **interactive and animated content** without page reload.
No scripting or dynamic effects by itself.	Uses scripts to change content, style, and layout **on the fly**.

---

2) Checkboxes vs Radio Buttons

Checkbox	Radio Button
Allows **multiple selections** from a set of options.	Allows **only one selection** from a group of options.
Each box operates independently.	Radio buttons are **mutually exclusive** in a group.
Used when more than one option can be chosen.	Used when only one option is allowed.

---

3) Transparency Filter vs Wipe Filter

Transparency Filter	Wipe Filter
Makes an element or image **partially or fully transparent**.	Creates a **wiping effect** that reveals or hides elements gradually.
Controls **opacity** levels of objects.	Controls animation of content sliding in or out.
Used for fading effects.	Used for transition or page change animations.

3 B) What are HTML attributes? Discuss different types of attribute used in HTML <img> tag with example.

What are HTML Attributes?

HTML attributes provide **additional information** about HTML elements.

They are placed inside the **opening tag** and usually come in **name="value"** pairs.

Attributes help define **element behavior, appearance, or metadata**.

Different Types of Attributes in `<img>` Tag with Examples:

`src` **(Source):**

Specifies the **path or URL of the image** to be displayed.

Example: `<img src="flower.jpg" alt="Flower">`

`alt` **(Alternative Text):**

Provides **text description** of the image if it cannot be displayed.

Important for **accessibility and SEO**.

Example: `<img src="flower.jpg" alt="A red flower">`

`width` **and** `height`**:**

Specify the **dimensions** of the image in pixels or percentage.

Example: `<img src="flower.jpg" width="200" height="150">`

`title`**:**

Shows a **tooltip text** when the mouse hovers over the image.

Example: `<img src="flower.jpg" title="Beautiful flower">`

3 C) What is a selected box in HTML? Write an example program to draw a select box for inserting five department names from your university.

What is a Select Box in HTML?

A **select box** (also called a dropdown list) allows users to **choose one option from a list**.

It is created using the `<select>` tag, with multiple `<option>` tags inside representing choices.

Example Program:

```
<!DOCTYPE html>
<html>
<head>
  <title>Select Box Example</title>
</head>
<body>
  <form>
    <label for="department">Select Department:</label>
    <select id="department" name="department">
      <option value="cse">Computer Science and Engineering</option>
      <option value="eee">Electrical and Electronics
Engineering</option>
      <option value="stat">Statistics</option>
      <option value="mech">Mechanical Engineering</option>
      <option value="civil">Civil Engineering</option>
    </select>
  </form>
</body>
</html>
```

4 A) What do you know about front-end and back-end programming languages? Explain “PHP is one of the popular web programming language”

Front-end and Back-end Programming Languages (2 marks)

> **Front-end programming languages** are used to create the **user interface and experience** of a website or web application. They run **in the browser** and handle layout, design, and interaction.
>
> Examples: **HTML, CSS, JavaScript**.
>
> **Back-end programming languages** run **on the server** and handle the **business logic, database interactions, and server-side processing**.
>
> Examples: **PHP, Python, Java, Node.js**.

Both are necessary to build fully functional web applications where front-end handles what users see and interact with, while back-end manages data and application logic behind the scenes.

“PHP is one of the popular web programming languages” - Explanation (2 marks)

PHP is a **server-side scripting language** widely used for **creating dynamic web pages and web applications**.

It is **open-source, easy to learn, and well-supported**, making it popular among developers.

PHP can **connect with databases like MySQL**, making it ideal for data-driven websites.

Many popular platforms like **WordPress, Facebook, and Wikipedia** use PHP, proving its reliability and widespread adoption.

4 B) Find out the errors from the following code in PHP(if exist):

<?php

$age=array(“Peter”=>”35”, “Ben”=>”37”,”Joe”=>”43”);

foreach($age as $idx>$val){

echo”$id is $val years old.
”;

}?>

Write the correct code.

Errors in the Given PHP Code:

Wrong arrow operator in `foreach` loop:

`foreach($age as $idx > $val)` should be `foreach($age as $id => $val)`.

Use `=>` (double arrow) to separate key and value, not `>`.

Variable name mismatch:

Inside the loop, `echo "$id is $val years old.<br>";` but the key variable is `$idx` in the loop header (though it should be `$id`). Both names must be consistent.

Quotes used:

The code uses **smart quotes** (“ ”), but PHP requires **straight quotes** (" or ').

Corrected PHP Code:

```
<?php$age = array("Peter" => "35", "Ben" => "37", "Joe" => "43");
foreach ($age as $id => $val) {
    echo "$id is $val years old.<br>";
}?>
```

4 C) Write the difference between the following:

1) include_once() and require_once()

2) mysqli_fetch_array() and mysqli_fetch_assosc()

1) include_once() vs require_once()

include_once()	require_once()
Includes and evaluates the specified file once.	Includes and evaluates the specified file once.
If the file is not found, it throws a warning but the script continues execution.	If the file is not found, it throws a fatal error and stops script execution.

2) mysqli_fetch_array() vs mysqli_fetch_assoc()

mysqli_fetch_array()	mysqli_fetch_assoc()
Fetches a result row as an associative array, numeric array, or both (default both).	Fetches a result row as an associative array only.
Uses more memory because of duplicate keys (numeric + associative).	More efficient, fetches only associative keys (column names).

4 D) What do you know about multi-dimensional array in PHP? How can you print all the values in tabular format as mentioned below from the following array in PHP

Dept. Name--No. of Studens--No. of Teachers

$cars=array(array(“CSE”,250,18), array(“EEE”,200,13), array(“ICE”, 190,2), array(“ETE”,80,15));

What is a Multi-Dimensional Array in PHP? (1 mark)

A **multi-dimensional array** is an array that contains one or more arrays inside it as elements.

It allows storing data in a **table-like structure** with rows and columns.

Example PHP Code to Print the Given Array in Tabular Format: (3 marks)

```
<?php$cars = array(
  array("CSE", 250, 18),
  array("EEE", 200, 13),
  array("ICE", 190, 2),
  array("ETE", 80, 15)
);
echo "<table border='1'>";echo "<tr><th>Dept. Name</th><th>No. of Students</th><th>No. of Teachers</th></tr>";
foreach ($cars as $row) {
  echo "<tr>";
  foreach ($row as $value) {
    echo "<td>$value</td>";
  }
  echo "</tr>";
}
echo "</table>";?>
```

5 A) Discuss the function of mysql_insert_id() in PHP with example. What are superglobal variables inPHP? Write the output of the following code if it run using the as http://www.example.com/test/variables.php

<?php

echo $_SERVER[‘PHP_SELF’];

**echo”
”;**

echo $_SERVER[‘SERVER_NAME’];

?>

Function of `mysql_insert_id()` in PHP (2 marks)

The `mysql_insert_id()` function (or `mysqli_insert_id()` in MySQLi) returns the **ID of the last record inserted into an AUTO_INCREMENT column** in the current database connection.

It is useful when you need to get the **primary key of the newly inserted record**.

Example:

```
<?php$conn = mysqli_connect("localhost", "root", "", "testdb");$sql = "INSERT INTO students (name, age) VALUES ('Rahim', 22)";
mysqli_query($conn, $sql);
$last_id = mysqli_insert_id($conn);
echo "Last inserted ID is: " . $last_id;?>
```

What are Superglobal Variables in PHP?

Superglobal variables are **built-in PHP variables** that are always accessible, regardless of scope (inside functions, classes, or scripts).

Examples: `$_GET`, `$_POST`, `$_SESSION`, `$_COOKIE`, `$_SERVER`, `$_FILES`.

Output of the Given Code: (2 marks)

```
<?phpecho $_SERVER['PHP_SELF'];echo "<br>";echo $_SERVER['SERVER_NAME'];?>
```

If run at:
`http://www.example.com/test/variables.php`

Output:

```
/test/variables.php
www.example.com
```

5 B) Write the different ways of receiving 'name' using PHP from the following HTML FORM after submission:

<form method="post" action="<?php echo $_SERVER['PHP_SELF']:?>">

First Name:<input type="text" name="fname">

Last name: <input type="text" name="lname">

<input type="submit">

</form>

Different Ways of Receiving 'name' in PHP from the Given Form:

Using `$_POST` **(as form method is POST):**

```
$first_name = $_POST['fname'];$last_name  = $_POST['lname'];
```

Using `$_REQUEST`**:**

```
$first_name = $_REQUEST['fname'];$last_name  =
$_REQUEST['lname'];
```

Using `filter_input()`**:** (More secure)

```
$first_name = filter_input(INPUT_POST, 'fname');$last_name  =
filter_input(INPUT_POST, 'lname');
```

5 C) What do you know about session and cookie in PHP? Give an example program to create session and destroy session in a webpage?

Session and Cookie in PHP (2 marks)

Session:

A session stores **user information on the server** (e.g., login data) for use across multiple pages.

Data is stored temporarily until the user logs out or the session expires.

Cookie:

A cookie stores **small pieces of data on the client's browser**.

It is often used for remembering user preferences or login credentials.

Key Difference:

Session data is stored on the **server**, while cookie data is stored on the **client's browser**.

Example Program to Create and Destroy a Session

```
<?php

// Start the session

session_start();


// Create session variables

$_SESSION['username'] = "Rahim";

$_SESSION['email'] = "rahim@example.com";


echo "Session variables are set.<br>";


// Destroy the session

session_unset();   // Unset all session variables

session_destroy(); // Destroy the session


echo "Session destroyed.";

?>
```

5 D) What is the function of 'in_array()' in PHP? Give an example to find the name "Glenn" from the following:
<?php $people = array("Peter","Joe", "Glenn", "Cleverland");?>

Function of `in_array()` in PHP

The `in_array()` function checks **if a specific value exists in an array**.

It returns **TRUE (1)** if the value is found, otherwise **FALSE (0)**.

Example to Find "Glenn":

```
<?php$people = array("Peter", "Joe", "Glenn", "Cleverland");
if (in_array("Glenn", $people)) {
    echo "Glenn is found in the array.";
} else {
    echo "Glenn is not in the array.";
}?>
```

6 A) What do you know about scripting languages? Write an example DHTML program to an alert"you have pressed a wrong key" using a keystroke related event.

What do you know about Scripting Languages? (1 mark)

Scripting languages are **interpreted languages** used to automate tasks or add functionality to web pages.

They run either **on the client-side (e.g., JavaScript)** or **server-side (e.g., PHP)**.

In web development, scripting languages make pages **dynamic and interactive**.

Example DHTML Program using Keystroke Event: (2 marks)

```
<!DOCTYPE html>
<html>
<head>
  <title>Keystroke Event</title>
  <script>
    function checkKey(event) {
      alert("You have pressed a wrong key");
    }
  </script>
</head>
```

```
<body onkeypress="checkKey(event)">
  <h2>Press any key to see the alert.</h2>
</body>
</html>
```

6 B) What is client-side validation? Give an example program in DHTML to validate a form with the following fields using JavaScript?

Roll--Name--Year--Department

11--Shaon--1--CSE

12--Rahim--2-EEE

What is Client-Side Validation? (1 mark)

Client-side validation is the process of **checking user input in a form using scripts like JavaScript before sending data to the server**.

It helps to **reduce server load** and provide **immediate feedback** to users.

Example DHTML Program for Form Validation: (4 marks)

```
<!DOCTYPE html>
<html>
<head>
  <title>Form Validation</title>
  <script>
    function validateForm() {
      var roll = document.getElementById("roll").value;
      var name = document.getElementById("name").value;
      var year = document.getElementById("year").value;
      var dept = document.getElementById("dept").value;

      if (roll == "" || name == "" || year == "" || dept == "") {
        alert("All fields must be filled!");
        return false;
      }

      if (isNaN(roll)) {
        alert("Roll must be a number!");
        return false;
      }
```

```
    return true;
  }
</script></head><body>
<h2>Student Form</h2>
<form onsubmit="return validateForm()">
  Roll: <input type="text" id="roll"><br><br>
  Name: <input type="text" id="name"><br><br>
  Year: <input type="text" id="year"><br><br>
  Department: <input type="text" id="dept"><br><br>
  <input type="submit" value="Submit">
</form>
</body>
</html>
```

6 C) What is CSS box model? Discuss different ways of adding CSS in a DHTML page?

What is CSS Box Model?

The **CSS box model** describes how every HTML element is treated as a rectangular box, consisting of **4 main areas**:

Content: The actual text or image inside the box.

Padding: Space between content and border (transparent area).

Border: The line that wraps around the padding and content.

Margin: Space between the element's border and surrounding elements (outer space).

Example:

```
div {
  width: 200px;
  padding: 10px;
  border: 2px solid black;
  margin: 20px;
}
```

Here, the total width = content (200px) + padding (10px + 10px) + border (2px + 2px) + margin (20px + 20px).

Different Ways of Adding CSS in a DHTML Page:

Inline CSS:

CSS written directly inside an HTML element using the `style` attribute.

```
<p style="color:blue;">This is blue text.</p>
```

Internal CSS:

CSS is placed inside `<style>` tags in the `<head>` section of an HTML file.

```
<style>
  p { color: red; }
</style>
```

External CSS:

A separate `.css` file is linked using `<link>` tag.

```
<link rel="stylesheet" href="style.css">
```

7 A) How can you apply JavaScript to a web page?

Ways to Apply JavaScript to a Web Page

JavaScript can be added to an HTML page in **three main ways**:

---

1. Inline JavaScript

JavaScript code is written directly inside an HTML tag using the `onclick`, `onmouseover`, etc., attributes.
Example:

```
<button onclick="alert('Hello!')">Click Me</button>
```

---

2. Internal JavaScript

JavaScript code is written inside `<script>` tags within the HTML file (usually inside `<head>` or at the end of `<body>`).
Example:

```
<script>
  alert("Welcome to my website!");</script>
```

---

3. External JavaScript

JavaScript code is kept in a separate `.js` file, and the file is linked using the `<script src="file.js"></script>` tag.
Example:

```
<script src="script.js"></script>
```

---

4. Using Event Handlers

JavaScript functions can be triggered using HTML events like `onload`, `onclick`, `onchange`.
Example:

```
<body onload="alert('Page Loaded!')">
```

7 B) Mention the environments where VBScript could be run. What are the variable naming conventions in the VBScript?

Environments where VBScript could be run: (2 marks)

VBScript (Visual Basic Scripting Edition) can run in the following environments:

Web Browsers:

Primarily supported by **Internet Explorer (IE)** as a client-side scripting language.

Windows Script Host (WSH):

Can execute `.vbs` files directly on Windows operating systems.

Microsoft Office Applications:

Can be embedded and run in applications like Excel or Word for automation (similar to VBA).

ASP (Active Server Pages):

Used as a server-side scripting language for dynamic web pages.

Variable Naming Conventions in VBScript: (2 marks)

Variable names **must begin with a letter** (A-Z or a-z).

They **cannot contain spaces or special characters** (except underscore _).

They must be **unique within their scope**.

The maximum length is **255 characters**.

VBScript is **not case-sensitive**, so `Name` and `name` are the same.

Example:

```
Dim studentName
studentName = "Rahim"
```

7 C) What is the technology that is used by VBScript? What is a dictionary object in VBScript?

Technology used by VBScript: (2 marks)

VBScript (Visual Basic Scripting Edition) is based on **COM (Component Object Model)** technology.

It is a **lightweight scripting language** developed by Microsoft, primarily used for:

Client-side scripting in **Internet Explorer**.

Server-side scripting with **ASP (Active Server Pages)**.

Automation tasks using **Windows Script Host (WSH)**.

What is a Dictionary Object in VBScript? (2 marks)

A **Dictionary object** in VBScript is similar to an **associative array or collection**.

It stores data as **key-value pairs**, where each key is unique.

It provides methods to **add, remove, or check** values using keys.

Example:

```
Set dict = CreateObject("Scripting.Dictionary")
dict.Add "CSE", "Computer Science"
dict.Add "EEE", "Electrical Engineering"
MsgBox dict("CSE")  ' Output: Computer Science
```

8 A) Discuss the named function in JavaScript with example?

Named Function in JavaScript

A **named function** is a function that has a specific name and can be called or invoked anywhere in the script after its declaration.

Named functions improve code readability and can be reused multiple times.

They are defined using the `function` keyword followed by the function name and parentheses.

Syntax:

```
function functionName(parameters) {
  // function body
}
```

Example:

```
function greet(name) {
  console.log("Hello, " + name + "!");
}
// Calling the functiongreet("Rahim");  // Output: Hello, Rahim!
```

Key Points:

Named functions are **hoisted**, meaning they can be called before their definition in the code.

They help organize code into reusable blocks.

8 B) Explain the all possible ways to create objects in JavaScript. Give proper examples?

All Possible Ways to Create Objects in JavaScript

---

1. Using Object Literals

Simplest way to create objects using `{}` with key-value pairs.
Example:

```
let person = {
  name: "Rahim",
  age: 25
};
```

---

2. Using the `new Object()` Syntax

Create an empty object and add properties later.
Example:

```
let person = new Object();
person.name = "Rahim";
person.age = 25;
```

---

3. Using Constructor Functions

Define a function and use `new` keyword to create objects with shared structure.
Example:

```
function Person(name, age) {
  this.name = name;
  this.age = age;
}let rahim = new Person("Rahim", 25);
```

---

4. Using `Object.create()` Method

Create an object with a specified prototype object.
Example:

```
let proto = {
  greet: function() {
    console.log("Hello!");
  }
```

```
};let person = Object.create(proto);
person.name = "Rahim";
```

8 C) Write differences between Call, Apply and Bind with example.

Differences between `call()`, `apply()`, and `bind()` in JavaScript

Feature	call()	apply()	bind()
Purpose	Calls a function immediately with a given this value and arguments passed **individually.**	Calls a function immediately with a given this value and arguments passed **as an array.**	Returns a **new function** with a bound this value and optional arguments; does **not call immediately.**
Arguments	Passed as comma-separated list.	Passed as an array or array-like object.	Arguments can be passed during binding or when the returned function is called.
Execution	Executes the function immediately.	Executes the function immediately.	Does **not execute** immediately; returns a new function.

Examples:

```
function greet(greeting, punctuation) {
  console.log(greeting + ", " + this.name + punctuation);
}
const person = { name: "Rahim" };
// Using call()
greet.call(person, "Hello", "!");    // Output: Hello, Rahim!
// Using apply()
greet.apply(person, ["Hi", "!!"]);   // Output: Hi, Rahim!!
// Using bind()const greetRahim = greet.bind(person,
"Hey");greetRahim("?");                    // Output: Hey, Rahim?
```