

Web technology II 2081 solution

1) what is XML ? explain basic structure of XML explain AJAX with its components.

Ans- XML is a markup language used to store, share, and transport data in a structured format. It uses user-defined tags and is both human- and machine-readable.

✓ Basic Structure of XML

1. Declaration (Optional):

<?xml version="1.0" encoding="UTF-8"?>

2. Root Element:

One main outer tag (e.g., <data>...</data>)

3. Elements:

Data stored in tags

Example: <name>John</name>

4. Attributes:

Info inside opening tag

Example: <book id="101">

5. Text Content:

Data inside tags

6. Comments:

<!-- This is a comment -->

• Example:

<book>

<title>XML</title>

<author>John</author>

</book>

Explain AJAX with its components

AJAX is a web development technique used to create fast and dynamic web pages. It allows web pages to be updated asynchronously by exchanging small amounts of data with the server behind the scenes, without reloading the entire page.

The main components of an AJAX application:-

HTML (or XHTML): HTML is used for structuring the content and elem AC of a web page. It provides the foundation for presenting the data and user interface in an AJAX application

CSS (Cascading Style Sheets): CSS is used for styling and formatting the HTML elements, ensuring a visually appealing and consistent look for the application.

JavaScript: JavaScript is a programming language that enables interactivity and dynamic behavior in web pages. It is a key component of AJAX as it allows for making asynchronous requests to the server, handling responses, and updating the web page without a full reload.

XMLHttpRequest (XHR) object: The XMLHttpRequest object is a core component of AJAX. It provides the ability to send HTTP requests asynchronously to the server and receive responses. It serves as the bridge between the JavaScript code running in the browser and the server-side technologies

Data Formats: AJAX applications commonly use data interchange formats like JSON (JavaScript Object Notation) or XML (eXtensible Markup Language) to send and receive data between the client and server. JSON has become more popular due to its lightweight and easy-to-use nature.

DOM Manipulation: The Document Object Model (DOM) is a programming interface that represents the structure of an HTML or XML document as a tree-like structure. AJAX applications often manipulate the DOM using JavaScript to dynamically update specific parts of the web page based on the received data, providing a responsive user experience.

2) **what is XAMPP? Differentiate between server-side scripting vs client side scripting .**

Ans -XAMPP is a free and open-source software package that provides a local web server environment for web development and testing.

Different between server side scripting vs cliend sadi scripting

Client-side scripting	Server-side scripting
Source code is visible to the user. Its main function is to provide the requested output to the end user. It usually depends on the browser and its version. It runs on the user's computer. There are many advantages linked with this like faster. response times, a more interactive application. It does not provide security for data.	Source code is not visible to the user because its output of server-sideside is an HTML page. Its primary function is to manipulate and provide access to the respective database as per the request. In this any server-side technology can be used and it does not depend on the client. The primary advantage is its ability to highly customize, response requirements, access rights based on user.

It is a technique used in web development in which scripts run on the client's browser. HTML, CSS, and javascript are used. No need of interaction with the server. It reduces load on processing unit of the server	It does not provide security for data. It is a technique that uses scripts on the webserver to produce a response that is customized for each client's request. PHP, Python, Java, Ruby are used. It is all about interacting with the servers. It is surge the processing lode on the server
--	--

3) Explain looping and its types used in PHP

Ans -Looping in PHP is used to execute a block of code repeatedly as long as a certain condition is true

◆ Types of Loops in PHP:

1. while loop

Repeats code as long as condition is true

```
$i = 1;
while($i <= 5){
    echo $i;
    $i++;
}
```

2. do...while loop

Executes code at least once, then checks condition

```
$i = 1;
do {
    echo $i;
    $i++;
} while($i <= 5);
```

3. for loop

Used when number of iterations is known

```
for($i = 1; $i <= 5; $i++){
    echo $i;
```

```
}
```

4. foreach loop

Used to loop through arrays

```
$colors = ["red", "green", "blue"];  
foreach($colors as $color){  
    echo $color;  
}
```

4) Define function . Explain its types

Ans-A function in PHP is a block of code that performs a specific task and can be reused whenever needed.

✓ Types of Functions in PHP:

1. Built-in Functions:

Predefined by PHP, used directly.

2. User-defined Functions:

Created by the user for specific tasks.

3. Anonymous Functions (Closures):

Functions without a name, used as variables or argument

5) Explain GET and POST with help of code .

Ans -GET and POST are two HTTP methods used to send data from a form to the server.

✓ Example (GET):

PHP (get_example.php):

```
<?php  
echo "Hello, " . $_GET['name'];  
?>
```

✓ Example (POST):

PHP (post_example.php):

```
<?php  
echo "Hello, " . $_POST['name'];  
?>
```

6) What are cookies and sessions ? Explain file inclusion .

Ans-◆ Cookies:

Small files stored on the client's browser

Used to remember user info like name or preference

◆ Sessions:

Stored on the server

Used to store user data across multiple pages (e.g., login status)

✓ File Inclusion in PHP

Used to reuse code like headers, footers, or functions across multiple pages.

◆ include and require:

Both insert contents of one PHP file into another.

7) What is database ? write the PHP code to create MYSQL database and insert data into database and table .

Ans -A database is an organized collection of data that can be easily accessed, managed, and updated. ✓ PHP Code to **Create Database and Insert Data**

◆ 1. Create Database

```
<?php
$conn = mysqli_connect("localhost", "root", "");
mysqli_query($conn, "CREATE DATABASE mydb");
mysqli_close($conn);
?>
```

◆ 2. Create Table & Insert Data

```
<?php
$conn = mysqli_connect("localhost", "root", "", "mydb");
mysqli_query($conn, "CREATE TABLE students (id INT AUTO_INCREMENT PRIMARY KEY,
name VARCHAR(50), age INT)");
mysqli_query($conn, "INSERT INTO students (name, age) VALUES ('Ram', 20)");
mysqli_close($conn);
?>
```

8) what is Laravel? Explain MVC concept .Differentiate between authentication and authorization .

Ans- **Laravel** is a popular **PHP web framework** used to build web applications easily and efficiently. It provides tools and resources for routing, database management, templating, security, and more.

MVC Concept (Model-View-Controller):

- **Model:** Manages data and business logic (e.g., database interaction).
- **View:** Handles UI presentation (HTML/CSS).
- **Controller:** Connects Model and View; processes user input and updates Model/View accordingly.

Authentication vs Authorization:

Aspect	Authentication	Authorization
Definition	Verifying who the user is	Verifying what the user can do
Purpose	Confirms identity	Grants or denies access to resources
Example	Login with username & password	Access control like admin-only pages
Comes First?	Yes	After authentication

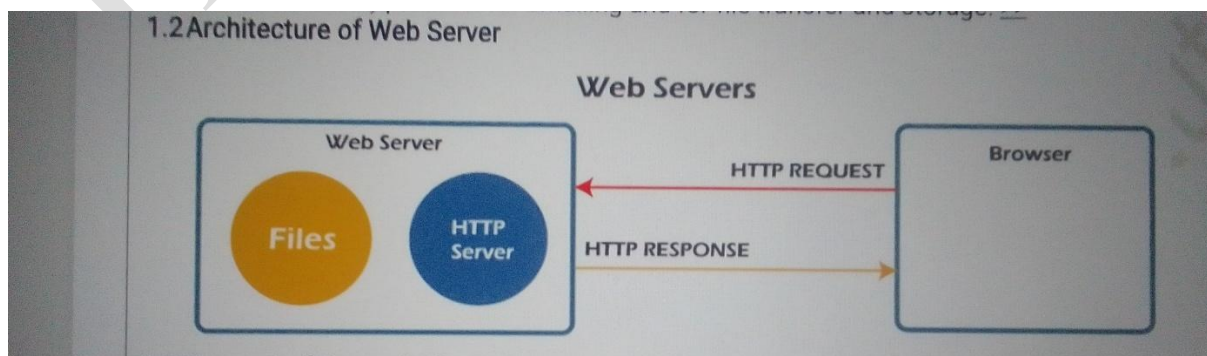
Regular/ Back 2081 solutions

1) Define web server .Explain the architecture of web server .

Ans –A web server is a computer system that stores processes

- It responds to request made by web browsers using the (HTTP) Hyper text transfer protocol

architecture of web server .



2.a) Explain the importance of database normalization in web development.

Ans-Importance of Database Normalization in Web Development

Eliminates Data Redundancy: Prevents duplicate data storage.

Ensures Data Integrity: Maintains accurate and consistent data.

Improves Query Efficiency: Speeds up data retrieval and updates.

Simplifies Maintenance: Makes the database easier to manage and scale.

2.b) Explain the use of alter ,update and delete queries wthi example

Ans- a clear explanation of the **ALTER**, **UPDATE**, and **DELETE** SQL queries with examples:

🔧 1. ALTER – Modify the structure of a table

Use: To add, delete, or modify columns in an existing table.

📝 2. UPDATE – Modify existing data in a table

Use: To change values in one or more rows.

✂ 3. DELETE – Remove records from a table

Use: To delete one or more rows based on a condition.

3.a) Write the advantages of using PHP for web development

Ans - Advantages of Using PHP for Web Development

Open Source – Free to use, reducing development costs.

Easy to Learn – Simple syntax makes it beginner-friendly.

Cross-Platform Compatibility – Runs on various operating system

3.b)Why comments are important while writing a program .

Ans- **Comments are important in PHP programming** (and all programming languages) because they make the code easier to understand, maintain, and debug. Here's why:

✓ Importance of Comments in PHP:

1. Improve Code Readability

Comments explain what specific parts of the code do, helping others (and your future self) understand it more easily.

2. **Simplify Maintenance**

When updating or fixing code later, comments act as helpful notes to quickly understand the logic.

3. **Support Team Collaboration**

In team projects, comments help developers understand each other's code without confusion.

4. **Explain Complex Logic**

Some code sections may be tricky—comments clarify why certain logic or methods are used.

5. **Temporarily Disable Code**

You can use comments to “turn off” code during testing or debugging

6. **Example in PHP:**

```
<?php
function add($a, $b) {
    return $a + $b;
}
?>
```

3.c) Write a rules to create a variables in php

Ans- Variable must start with a dollar (\$) sign, followed by the variable name.

- It can only contain alpha-numeric character and underscore (A-z, 0-9, _).
- Variable name must start with a letter or underscore (_) character.
- A PHP variable name cannot contain spaces.
- One thing to be kept in mind that the variable name cannot start with a number or special symbols.
- PHP variables are case-sensitive, so \$name and \$NAME both are treated as different variables.

Declaring string, integer, and float

```
---
--- <?PHP
```

```
--- $str="hello string";
```

```
--- $x=200;
```

```
--- $y=44.6;
```

1. Echo “ string is :\$str
”;
Echo “ integer is :\$x
”;
Echo “ float is :\$y
”;
?>

4.a) Explain error handling in PHP with example .

Ans- Error handling in PHP is the process of catching and managing errors to prevent your application from crashing and to handle issues gracefully.

◆ Explanation with example

1. **try . . . catch (Exception Handling):**

```
<?php
try {
    throw new Exception("Something went wrong!");
} catch (Exception $e) {
    echo "Error: " . $e->getMessage();
}
?>
```

2. **set_error_handler()** (Custom Error Handler):

```
<?php
function myErrorHandler($errno, $errstr) {
    echo "Error: [$errno] $errstr";
}
set_error_handler("myErrorHandler");
echo $undefinedVar;
?>
```

4.b) List out and explain the different file handling functions used in PHP

Ans—PHP provides several built-in functions to handle files — for reading, writing, creating, or closing files.

📁 Common File Handling Functions in PHP:

Function	Description
<code>fopen()</code>	Opens a file (for reading, writing, etc.)
<code>fread()</code>	Reads data from an open file
<code>fwrite()</code>	Writes data to a file
<code>fclose()</code>	Closes an open file
<code>file_exists()</code>	Checks if a file exists
<code>filesize()</code>	Returns the size of a file in bytes
<code>file_get_contents()</code>	Reads the entire file into a string (quick read)
<code>file_put_contents()</code>	Writes a string to a file (quick write)
<code>unlink()</code>	Deletes a file
<code>fgets()</code>	Reads a single line from a file
<code>feof()</code>	Checks if the end of file (EOF) has been reached

5. a. **Explain classes and objects in PHP.**

Ans- A class is a blueprint or a template for creating objects. It is a user-defined data type that bundles data (in the form of variables called properties) and functions that operate on

that data (called methods). A class defines the structure and behavior that its objects will have, but it is not a concrete entity itself.

Blueprint: Think of a car designer's blueprint. It defines all the specifications—engine size, number of doors, color options—but it isn't a physical car.

Properties: Variables defined inside a class (e.g., \$color, \$model).

Methods: Functions defined inside a class that define its behavior (e.g., startEngine(), accelerate()).

Keyword: The class keyword is used to define a class.

Object:

An object is an instance of a class. When a class is defined, no memory is allocated until an object is created from it. An object is a concrete entity that has its own state (the specific values of its properties) and can perform the behaviors defined by its class's methods. You can create multiple objects from a single class, each with its own independent properties.

Instance: If a class is the car blueprint, an object is an actual car built from that blueprint. You can build many cars (objects) from the same blueprint, and each can have a different color or owner.

Keyword: The new keyword is used to create an object (instantiate a class).

Example in PHP:

```
<?php

class Car {

    public $color;

    public $model;

    public function __construct($color, $model) {

        $this->color = $color;

        $this->model = $model;

    }

    public function showDetails() {

        return "This is a " . $this->color . " " . $this->model . " .";

    }

}
```

```

}

}

$car1 = new Car("Red", "Toyota Camry");

$car2 = new Car("Blue", "Honda Civic");

echo $car1->showDetails(); // Output: This is a Red Toyota Camry.

echo "<br>";

echo $car2->showDetails(); // Output: This is a Blue Honda Civic.

echo "<br>";

echo "The first car's color is " . $car1->color; // Output: The first car's color is Red.

?>

```

5.b) Explain method overriding.

Ans -Method overriding is an OOP concept related to inheritance. It allows a subclass (or child class) to provide its own specific implementation of a method that is already defined in its superclass (or parent class).

When a child class inherits from a parent class, it inherits all the public and protected properties and methods of the parent. However, sometimes the child class needs to perform the behavior of a method differently than the parent. In such cases, the child class can "override" the parent's method.

7.a)What is XML element? Explain the physical structure of XML.

Ans- An XML element is the fundamental building block of an XML document. It represents a piece of data within the document's hierarchy. An element usually consists of a start tag, content, and an end tag.

Physical Structure of XML

The physical structure of an XML document describes how it is composed. A well-formed XML document follows a specific structure:

Prologue:

This is the optional first part of an XML document. It includes:

XML Declaration: Specifies the XML version and character encoding. It is highly recommended.

<?xml version="1.0" encoding="UTF-8"?>

Document Type Declaration (DTD) or Schema reference: An optional declaration that defines the structure and legal elements of the document.

Comments and Processing Instructions: e.g., <!-- This is a comment -->

Root Element:

Every XML document must have exactly one root element. This element is the parent of all other elements in the document and encloses the entire content.

Nested Elements (The Body):

Inside the root element, there can be any number of nested elements (child elements). These elements form the hierarchical data structure of the document. Elements must be properly nested, meaning a child element must be closed before its parent element is closed.

7.b) Definition of AJAX Advantages of AJAX:

Ans-AJAX stands for Asynchronous JavaScript and XML. It is not a single technology but a set of web development techniques used to create fast, dynamic, and interactive web applications.

The core idea of AJAX is to allow a web page to exchange small amounts of data with a server in the background, without having to reload the entire page. By updating only specific parts of a page, AJAX provides a smoother and more responsive user experience, similar to that of a desktop application. While the name includes "XML," modern applications often use JSON (JavaScript Object Notation) for data exchange because it is lighter and easier to parse in JavaScript.

Advantages of AJAX:

Improved User Experience: The biggest advantage is the enhanced interactivity and responsiveness. Content can be updated on the fly without the jarring effect of a full page refresh, making the web application feel faster and smoother.

Reduced Bandwidth and Faster Load Times: Since only the necessary data is requested from the server (not the entire HTML, CSS, etc.), it reduces server load and bandwidth usage. This leads to quicker updates on the client-side.

Asynchronous Calls: The "A" in AJAX stands for asynchronous. This means the browser doesn't have to wait for the server's response to continue working. The user can keep interacting with the page while data is being fetched in the background.

Separation of Data and Presentation: AJAX promotes a cleaner architecture where the server-side script is only responsible for providing raw data (like JSON or XML), and the

client-side script is responsible for fetching and displaying it. This makes the code easier to manage and debug.

Platform and Architecture Independence: AJAX is supported by all major modern browsers. It relies on open standards like JavaScript and HTTP requests, making it a versatile technique that can be used with any server-side programming language (PHP, Python, Java, etc.).

8.a) Explain Laravel and its features.

Ans-Laravel is a free, open-source PHP web framework created by Taylor Otwell. It is designed for developing web applications following the Model-View-Controller (MVC) architectural pattern. Laravel aims to make the development process elegant and creative by providing a clean and expressive syntax, along with a rich set of tools and features that handle common web development tasks, such as routing, authentication, sessions, and caching. Its philosophy is to make development an enjoyable experience without sacrificing application functionality.

Key Features of Laravel:

Eloquent ORM (Object-Relational Mapper): Eloquent provides a simple, ActiveRecord implementation for working with your database. Each database table has a corresponding "Model" which is used to interact with that table. This allows developers to write database queries using PHP syntax instead of raw SQL.

Blade Templating Engine: Blade is a simple yet powerful templating engine. It allows developers to use plain PHP code in their views, but also provides its own control structures like conditionals and loops. Blade templates are compiled into plain PHP code and cached, adding no overhead to the application.

Artisan Command-Line Interface (CLI): Artisan is the built-in command-line tool for Laravel. It provides a number of helpful commands for tasks such as creating controllers, models, migrating databases, and seeding data, which significantly speeds up the development workflow.

MVC Architecture Support: Laravel is built on the MVC pattern, which ensures a clear separation between business logic (Model), presentation (View), and request handling (Controller). This makes the application more organized, scalable, and easier to maintain.

Routing System: Laravel provides a simple and expressive way to define application routes. All routes are defined in the routes/ directory, making it easy to manage application URLs and link them to specific controller actions or closures.

Migrations and Seeding: Migrations act as version control for your database schema, allowing your team to easily modify and share the application's database schema. Seeding provides a way to populate your database tables with test or default data.

8.b) Explain briefly about MVC concept in Laravel.

Ans-The Model-View-Controller (MVC) concept in Laravel is an architectural pattern that separates an application into three interconnected logical components. This separation makes the application more organized, scalable, and easier to manage by assigning specific responsibilities to each component.

Here is a brief explanation of each part and how they work together in Laravel:

Model

Responsibility: Manages the data and business logic of the application. It is responsible for interacting with the database.

In Laravel: Models are typically Eloquent ORM classes that reside in the app/Models directory. Each model corresponds to a database table (e.g., a User model interacts with the users table). The model handles all database operations like retrieving, inserting, updating, and deleting data. It also defines relationships between tables (e.g., a user has many posts).

View

Responsibility: The presentation layer. It is responsible for displaying data to the user and handling user interaction (the User Interface or UI).

In Laravel: Views are Blade template files with a .blade.php extension, located in the resources/views directory. They contain the HTML structure of the application and use Blade syntax to display the data passed to them from the Controller. Views should contain minimal PHP logic, focusing only on presentation.

Controller

Responsibility: Acts as the intermediary between the Model and the View. It handles incoming user requests, processes them, interacts with the Model to fetch or manipulate data, and then passes that data to the appropriate View to be rendered and sent back to the user.

In Laravel: Controllers are classes located in the app/Http/Controllers directory. They contain methods (actions) that are triggered by the application's Routes. The controller orchestrates the flow of the application.

2080 Back solution

1.) Explain application of XML .What are the advantages of AJAX ?

Ans - XML is used to store and transport data. It is both human-readable and machine-readable, making it ideal for data exchange across different systems and platforms.

Applications of XML:

1. **Data Exchange between Systems:**
 - XML is commonly used to exchange data between different applications, particularly over the internet (e.g., between a client and a server).
 2. **Web Services:**
 - Used in SOAP (Simple Object Access Protocol) and other web services to encode messages.
 3. **Configuration Files:**
 - Many software systems (e.g., Java, .NET) use XML files to store configuration settings.
 4. **RSS Feeds:**
 - Used in RSS and Atom to syndicate content such as news, blogs, and updates.
 5. **Document Representation:**
 - XML formats like DocBook or XHTML are used to structure documents.
 6. **Database Data Export/Import:**
 - XML is often used to export and import data between databases or between databases and applications.
 7. **Mobile and Embedded Systems:**
 - Used in Android (e.g., layout and configuration files).
-

2) Advantages of AJAX (Asynchronous JavaScript and XML):

AJAX is a set of web development techniques that allows web applications to send and retrieve data from a server asynchronously without refreshing the entire web page.

Advantages:

1. **Improved User Experience:**
 - Web pages update dynamically without needing a full reload, creating a smoother user experience.
2. **Faster Interactions:**
 - Only required data is sent and received, reducing bandwidth usage and increasing speed.
3. **Reduced Server Load:**
 - AJAX only fetches necessary data, minimizing unnecessary requests and server processing.
4. **Partial Page Updates:**
 - Enables dynamic content loading (e.g., search suggestions, live notifications) without disturbing the rest of the page.
5. **Responsive Interfaces:**
 - Supports rich, interactive interfaces similar to desktop applications.
6. **Platform and Language Independent:**
 - Can be used with any server-side technology (PHP, ASP.NET, Java, Python, etc.).
 - **2.a) Why is PHP considered a server-side scripting language?**
 - **Ans-**PHP (Hypertext Preprocessor) is considered a server-side scripting language because:

✓ 1. Executes on the Server:

- PHP code is executed on the **web server**, not in the user's browser.
- The server processes the PHP code and generates **HTML output** that is sent to the client (browser).

✓ 2. Hidden from Users:

- Since PHP runs on the server, the actual PHP code is **never visible** to the user.
- Users only see the final rendered HTML, CSS, and JavaScript in their browsers.

✓ 3. Access to Server Resources:

- PHP can interact with **databases, file systems, sessions, and server-side configurations**, which are not accessible from client-side languages like JavaScript

✓ 4. Dynamic Content Generation:

- PHP is often used to create **dynamic web pages** that change based on user input, session data, or database content.

✓ 5. Integration with Databases:

- PHP easily connects to databases like **MySQL, PostgreSQL, SQLite**, etc., making it suitable for building data-driven web applications.

✓ Example:

```
<?php
$name = "John";
echo "Hello, $name!";
?>
```

2.b) Explanation of Variable, Global Variable, and Constants in PHP

Ans- Variable: A variable in PHP is a container used to store data like strings, numbers, or arrays.

Example:

```
$name = "Alice";
$age = 25;
echo "Name: $name, Age: $age";
```

✓ Global Variable:

A **global variable** is a variable that is declared **outside a function** and is accessible **globally** in the script.

- But inside functions, you must use the `global` keyword to access it.

Example:

```
$x = 10;
$y = 20;

function add() {
    global $x, $y;
    echo $x + $y;
}

add(); // Output: 30
```

✔Constant:

A **constant** is a name or an identifier for a fixed value that **cannot change** during the execution of the script.

Example:

```
define("SITE_NAME", "MyWebsite");
echo SITE_NAME;
```

3.)What are the conditional statements used in PHP ? Explain with example

Ans-Conditional statements are used to make decisions in a program. They allow the code to execute certain blocks only if specific conditions are true.

1. if Statement

Executes a block of code *if a condition is true*.

```
php
if (condition) {
}
```

Example:

```
php
$age = 20;
if ($age >= 18) {
    echo "You are eligible to vote.";
}
```

2. if...else Statement

Executes one block of code *if the condition is true, another block **if it is false*.

```
php
if (condition) {
} else {
}
```

Example:

```
php
$age = 16;
if ($age >= 18) {
    echo "You can vote.";
} else {
    echo "You are too young to vote.";
}
```

3. if...elseif...else Statement

Checks *multiple conditions*.

```
php
if (condition1)
} elseif (condition2) {
} else
}
```

Example:

```
php
$marks = 75;
if ($marks >= 90) {
    echo "Grade A";
} elseif ($marks >= 75) {
    echo "Grade B";
} else {
```

```
echo "Grade C";  
}
```

4. switch Statement

Used to perform *different actions based on different values* of the same variable.

```
php  
switch (expression) {  
case value1:  
break;  
case value2:  
break;  
default:  
}
```

Example:

```
php  
$day = "Monday";  
switch ($day) {  
case "Monday":  
echo "Start of the week.";  
break;  
case "Friday":  
echo "Weekend is coming!";  
break;  
default:  
echo "Just another day.";  
}
```

4. a) Define function and its types in brief.

Ans -A function is a block of code that performs a specific task, increases reusability, and avoids code repetition. In PHP, functions can be predefined or user-defined.

Types of Functions in PHP:

1. Built-in Functions: Functions provided by PHP like strlen(), date(), array_merge().

2. User-defined Functions: Created by users to perform specific tasks.

```
function greet() {  
    echo "Hello, User!";  
}
```

3. Anonymous Functions (Closures): Functions without a name, often used as arguments.

```
$sayHi = function() {  
    echo "Hi!";  
};
```

4. Recursive Functions: A function that calls itself.

```
function factorial($n) {  
    if ($n <= 1) return 1;  
    else return $n * factorial($n - 1);  
}
```

4. b) Differentiate between GET and POST. [4 Marks]

Ans - differentiating between GET and POST:

Feature	GET	POST
Purpose	Used to retrieve data from a server	Used to submit data to be processed
Data Handling	Sends data in the URL (query string)	Sends data in the body of the request
Visibility	Data is visible in the URL (less secure)	Data is hidden in the request body
Usage	Ideal for read-only operations	Ideal for sending form data or updates

5. Types of Operators in PHP & PHP vs JSP

Ans-Types of operators

a) Types of Operators in PHP:

1. Arithmetic Operators (+, -, *, /, %)
2. Assignment Operators (=, +=, -=)
3. Comparison Operators (==, !=, ===, <, >)
4. Logical Operators (&&, ||, !)

5. Increment/Decrement Operators (++ , --)

6. String Operators (. for concatenation)

7. Array Operators (+, ==, !=)

8. Conditional (Ternary) Operator (? :)

b) Difference Between PHP and JSP:

• PHP	JSP
Server-side scripting language	Java-based server pages
Interpreted	Compiled into servlets
Loosely typed language	Strongly typed (Java-based)
Runs on Zend/Apache servers	Runs on Tomcat/GlassFish
Faster for small apps	Better for large enterprise apps

6. Explain XAMPP and WAP for Login Form Connected with Database

Ans-XAMPP is a free and open-source cross-platform web server solution stack package developed by Apache Friends. It includes

WAP for Login Form and Connect to Database:

HTML (login.html):

```
<form method="POST" action="login.php">
Username: <input type="text" name="username"><br>
Password: <input type="password" name="password"><br>
<input type="submit" value="Login">
</form>
```

PHP (login.php):

```
<?php
$conn = new mysqli("localhost", "root", "", "login_db");
$username = $_POST['username'];
$password = $_POST['password'];
$sql = "SELECT * FROM users WHERE username='$username' AND password='$password'";
$result = $conn->query($sql);
if ($result->num_rows > 0) {
    echo "Login successful";
} else {
```

```
echo "Invalid credentials";  
}  
$conn->close();  
?>
```

7. a) Explain MVC concept.

Ans-MVC is a design pattern used in frameworks like Laravel.

Model: Handles data (database interactions).

View: User interface (HTML + Blade).

Controller: Logic that connects Model and View.

b) What is session? Explain Blade template

Ans-A session is used to store user information across multiple pages

Blade Templating (Laravel):

Blade is Laravel's powerful templating engine that allows:

Template inheritance (@extends, @section, @yield)

Control structures (@if, @foreach)

Displaying data ({{ \$name }})

Example:

```
<!-- layout.blade.php -->  
<html>  
<body>  
@yield('content')  
</body>  
</html>  
  
<!-- home.blade.php -->  
@extends('layout')  
@section('content')  
<h1>Welcome, {{ $user }}</h1>  
@endsection
```

8. WAP in PHP to Insert, Retrieve, Update, Delete Data

Ans-Insert:

```
$conn = new mysqli("localhost", "root", "", "mydb");  
$sql = "INSERT INTO users (name, email) VALUES ('John', 'john@example.com')";  
$conn->query($sql);
```

Retrieve:

```
$result = $conn->query("SELECT * FROM users");  
while ($row = $result->fetch_assoc()) {  
    echo $row['name'] . " - " . $row['email'];  
}
```

Update:

```
$conn->query("UPDATE users SET email='new@example.com' WHERE name='John'");
```

Delete:

```
$conn->query("DELETE FROM users WHERE name='John'");  
$conn->close;
```

2080 back solution

1. What is Web Server? Explain Dynamic Content and Web Application Architecture. **[2+3+5]**

Ans-A web server is software or hardware that stores, processes, and delivers web pages to users. It uses the HTTP/HTTPS protocol to respond to client (browser) requests. Examples: Apache, Nginx.

Dynamic Content

Dynamic content is generated in real-time based on user inputs, session data, or database queries.

Examples:

User login systems

Personalized dashboards

Shopping cart updates

PHP, Node.js, or Python scripts typically generate this type of content.

Web Application Architecture

Ans -It defines the structure of a web-based application:

Client-Side: HTML, CSS, JS for UI/UX.

Server-Side: Handles logic (PHP, Node.js).

Database Layer: Stores data (MySQL, MongoDB).

Example:

Browser ↔ Web Server ↔ Application Logic ↔ Database

2. a) Importance of Database Normalization in Web Development [5]

Ans -Process of organizing data to reduce redundancy and improve data integrity.

Importance:

1. Avoids duplicate data.
2. Improves data consistency.
3. Reduces storage space.
4. Makes database scalable.
5. Simplifies queries and updates.

3. a) Advantages of PHP for Web Development [5]

Ans –Advantages

1. **Open-source** – Free to use.
2. **Cross-platform** – Runs on Windows, Linux, etc.
3. **Easy Integration** – Easily connects with MySQL, HTML, CSS.
4. **Built-in Functions** – For file handling, database, form handling.
5. **Large Community** – Abundant learning resources and support

3.b) Explain different string functions used in php with exmple

1. **strlen()** – Returns the length of a string.
2. `$text = "Hello";`
3. `echo strlen($text); // Output: 5`
4. **strtoupper()** – Converts a string to uppercase.
5. `$name = "john";`
6. `echo strtoupper($name); // Output: JOHN`
7. **strtolower()** – Converts a string to lowercase.
8. `$name = "JANE";`

```
9. echo strtolower($name); // Output: jane
```

10. **str_replace()** – Replaces all occurrences of a search string with a replacement.

```
11. $message = "Hello World";
```

```
12. echo str_replace("World", "PHP", $message); // Output: Hello PHP.
```

4. a) write the Importance of Client-Side Validation [5]

Ans Importance of Client-Side Validation

1. **Fast Feedback** – Errors are caught before sending data to server.
2. **Reduces Server Load** – Fewer invalid requests.
3. **Better User Experience** – Real-time error alerts.
4. **Security Layer** – Although not sufficient alone.
5. **Efficient Forms** – Ensures required fields are filled correctly.

4.b) ALTER and UPDATE Commands with Example [4 Marks]

Ans- ALTER and UPDATE Commands with Example

Command	ALTER	UPDATE
Purpose	Used to modify the structure of a table (e.g., add, delete, or modify columns).	Used to modify data in existing records of a table.
Affects	Table structure	Table data
Common Uses	Add/remove columns, change data types, rename columns	Change values in rows based on conditions

Examples:

⇨ ALTER Command Example:

```
ALTER TABLE students ADD email VARCHAR(100);
```

Adds a new column `email` to the `students` table.

⇨ UPDATE Command Example:

```
UPDATE students SET email = 'john@example.com' WHERE id = 1;
```

Updates the `email` field of the student whose `id` is 1.

5. Constructors and Destructors in PHP. Insert Data in MySQL using PHP [5+5]

Ans -Constructor & Destructor:

```
class User {
```

```
function __construct() {  
    echo "Object Created";  
}  
  
function __destruct() {  
    echo "Object Destroyed";  
}  
}
```

Insert Data into MySQL:

```
$conn = new mysqli("localhost", "root", "", "testdb");  
$sql = "INSERT INTO users (name, email) VALUES ('John', 'john@example.com')";  
$conn->query($sql);  
$conn->close();
```

6. What is Database? Connect Multiple Databases using PHP [2+8]

Ans-A structured collection of data stored electronically for easy access and management.

Connect Multiple Databases

```
$db1 = new mysqli("localhost", "user1", "pass1", "db1");  
$db2 = new mysqli("localhost", "user2", "pass2", "db2");  
$result1 = $db1->query("SELECT * FROM table1");  
$result2 = $db2->query("SELECT * FROM table2");  
while($row = $result1->fetch_assoc()) {  
    echo $row['name'];  
}
```

7. XML Structure and Occurrence Indicators [4 Marks]

1. What is Web Server? Explain Dynamic Content and Web Application Architecture. [2+3+5]

Ans-A web server is software or hardware that stores, processes, and delivers web pages to users. It uses the HTTP/HTTPS protocol to respond to client (browser) requests. Examples: Apache, Nginx.

Dynamic Content

Dynamic content is generated in real-time based on user inputs, session data, or database queries.

Examples:

User login systems

Personalized dashboards

Shopping cart updates

PHP, Node.js, or Python scripts typically generate this type of content.

Web Application Architecture

Ans -It defines the structure of a web-based application:

Client-Side: HTML, CSS, JS for UI/UX.

Server-Side: Handles logic (PHP, Node.js).

Database Layer: Stores data (MySQL, MongoDB).

Example:

Browser ↔ Web Server ↔ Application Logic ↔ Database

2. a) Importance of Database Normalization in Web Development [5]

Ans -Process of organizing data to reduce redundancy and improve data integrity.

Importance:

1. Avoids duplicate data.
2. Improves data consistency.
3. Reduces storage space.
4. Makes database scalable.
5. Simplifies queries and updates.

3. a) Advantages of PHP for Web Development [5]

Ans –Advantages

- 1. Open-source** – Free to use.
- 2. Cross-platform** – Runs on Windows, Linux, etc.
- 3. Easy Integration** – Easily connects with MySQL, HTML, CSS.
- 4. Built-in Functions** – For file handling, database, form handling.

5. Large Community – Abundant learning resources and support

3.b) String Functions in PHP with Example [5]

Ans -Common String Functions in PHP with Examples:

1. **strlen()** – Returns the length of a string.
2. `$text = "Hello";`
3. `echo strlen($text); // Output: 5`
4. **strtoupper()** – Converts a string to uppercase.
5. `$name = "john";`
6. `echo strtoupper($name); // Output: JOHN`
7. **strtolower()** – Converts a string to lowercase.
8. `$name = "JANE";`
9. `echo strtolower($name); // Output: jane`
10. **str_replace()** – Replaces all occurrences of a search string with a replacement.
11. `$message = "Hello World";`
12. `echo str_replace("World", "PHP", $message); // Output: Hello PHP.`

4. a) Importance of Client-Side Validation [5]

Ans Importance of Client-Side Validation

1. **Fast Feedback** – Errors are caught before sending data to server.
2. **Reduces Server Load** – Fewer invalid requests.
3. **Better User Experience** – Real-time error alerts.
4. **Security Layer** – Although not sufficient alone.
5. **Efficient Forms** – Ensures required fields are filled correctly.

4.b) ALTER and UPDATE Commands with Example [4 Marks]

Ans- ALTER and UPDATE Commands with Example

Command	ALTER	UPDATE
Purpose	Used to modify the structure of a table (e.g., add, delete, or modify columns).	Used to modify data in existing records of a table.
Affects	Table structure	Table data
Common Uses	Add/remove columns, change data types, rename columns	Change values in rows based on conditions

Examples:

⇨ ALTER Command Example:

```
ALTER TABLE students ADD email VARCHAR(100);
```

Adds a new column email to the students table.

✦ **UPDATE Command Example:**

```
UPDATE students SET email = 'john@example.com' WHERE id = 1;
```

Updates the *email* field of the student whose *id* is 1.

5. Constructors and Destructors in PHP. Insert Data in MySQL using PHP [5+5]

Ans -Constructor & Destructor:

```
class User {  
    function __construct() {  
        echo "Object Created";  
    }  
    function __destruct() {  
        echo "Object Destroyed";  
    }  
}
```

Insert Data into MySQL:

```
$conn = new mysqli("localhost", "root", "", "testdb");  
$sql = "INSERT INTO users (name, email) VALUES ('John', 'john@example.com')";  
$conn->query($sql);  
$conn->close();
```

6. What is Database? Connect Multiple Databases using PHP [2+8]

Ans-A structured collection of data stored electronically for easy access and management.

Connect Multiple Databases

```
$db1 = new mysqli("localhost", "user1", "pass1", "db1");  
$db2 = new mysqli("localhost", "user2", "pass2", "db2");  
$result1 = $db1->query("SELECT * FROM table1");  
$result2 = $db2->query("SELECT * FROM table2");  
while($row = $result1->fetch_assoc()) {
```

```
echo $row['name'];  
}
```

7. XML Structure and Occurrence Indicators [4 Marks]

```
class User {  
function __construct() {  
echo "Object Created";  
}  
function __destruct() {  
echo "Object Destroyed";  
}  
}
```

Insert Data into MySQL:

```
$conn = new mysqli("localhost", "root", "", "testdb");  
$sql = "INSERT INTO users (name, email) VALUES ('John', 'john@example.com')";  
$conn->query($sql);  
$conn->close();
```

6. What is Database? Connect Multiple Databases using PHP [2+8]

Ans-A structured collection of data stored electronically for easy access and management.

Connect Multiple Databases

```
$db1 = new mysqli("localhost", "user1", "pass1", "db1");  
$db2 = new mysqli("localhost", "user2", "pass2", "db2");  
$result1 = $db1->query("SELECT * FROM table1");  
$result2 = $db2->query("SELECT * FROM table2");  
while($row = $result1->fetch_assoc()) {  
echo $row['name'];  
}
```

7. XML Structure and Occurrence Indicators [4 Marks]

Ans-XML (eXtensible Markup Language) is used to store and transport data. It follows a hierarchical structure with **tags** similar to HTML

✓ Basic XML Structure Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<bookstore>
  <book>
    <title>XML Basics</title>
    <author>John Doe</author>
    <price>29.99</price>
  </book>
</bookstore>
```

- **Root Element:** <bookstore>
- **Child Elements:** <book>, <title>, <author>, <price>
- Every opening tag must have a closing tag.
- Tags are case-sensitive.

◆ Occurrence Indicators in XML (used in DTD or XML Schema):

Occurrence indicators define **how many times** an element can appear.

Indicator	Meaning	Example Use
?	0 or 1 occurrence (optional)	<email>? – email is optional
*	0 or more occurrences	<phone>* – phone can repeat
+	1 or more occurrences	<item>+ – at least one item

8. Advantages of AJAX and Simple Web Page [5+5]

Ans -Advantages of AJAX:

1. No page reloads – Smooth user experience.
2. Faster responses – Only updates needed content.
3. Reduced server load – Efficient data exchange.
4. Improves interactivity
5. Better performance

AJAX Web Page Example:

```
<!DOCTYPE html>
<html>
<body>
<div id="result"></div>
<button onclick="loadData()">Click Me</button>
<script>
function loadData() {
const xhttp = new XMLHttpRequest();
```

```
xhttp.onload = function() {  
  document.getElementById("result").innerHTML = this.responseText;  
}  
xhttp.open("GET", "data.txt");  
xhttp.send();  
}  
</script>  
</body>  
</html>
```

Kaushal Yadav