

WEB DEVELOPMENT USING PHP
PG DIPLOMA IN WEB DESIGNING

Lecture 1

PHP Introduction

PHP का परिचय (Introduction to PHP)

PHP का पूरा नाम है: **Hypertext Preprocessor**। यह एक **server-side scripting language** है, जो खासतौर पर **web development** के लिए प्रयोग होती है। PHP को HTML के साथ आसानी से जोड़ा जा सकता है और यह वेबसाइट को डायनामिक बनाने में मदद करती है।

PHP कैसे काम करता है (How PHP Works)

1. यूज़र ब्राउज़र में कोई PHP फ़ाइल मांगता है।
2. रिक्वेस्ट सर्वर पर जाती है जहाँ PHP इंस्टॉल होता है।
3. PHP इंटरप्रेटर स्क्रिप्ट को एक्सिक्यूट करता है।
4. PHP का आउटपुट HTML के रूप में यूज़र को ब्राउज़र में दिखाई देता है।

php.ini फ़ाइल क्या है?

php.ini PHP की **configuration file** होती है। इसमें PHP का behavior सेट किया जाता है जैसे:

- Error reporting
- File uploads
- Max execution time
- Timezone सेटिंग आदि।

PHP का बेसिक सिंटैक्स (Basic PHP Syntax)

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

- PHP को <?php से शुरू और ?> से बंद किया जाता है।
- echo का प्रयोग आउटपुट देने के लिए किया जाता है।

PHP Variables (वैरिएबल्स)

PHP में वैरिएबल \$ से शुरू होते हैं।

```
<?php
$name = "Amit";
$age = 25;
?>
```

PHP Statements और Operators

Statements: PHP में हर स्टेटमेंट ; से समाप्त होती है।

Operators:

- Arithmetic: +, -, *, /, %
- Assignment: =, +=, -=
- Comparison: ==, !=, <, >, <=, >=
- Logical: &&, ||, !

Decision Making (if, if-else, switch)

```
<?php
$age = 20;

if ($age >= 18) {
    echo "Adult";
} else {
    echo "Minor";
}
?>
```

Switch Statement:

```
<?php
$day = "Monday";
switch($day) {
    case "Monday":
        echo "Start of Week";
        break;
    default:
        echo "Some other day";
}
?>
```

Loops (While, For, Foreach)

While Loop:

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

For Loop:

```
<?php
for ($i = 1; $i <= 5; $i++) {
    echo $i;
}
?>
```

Foreach (Array के लिए):

```
<?php
$fruits = ["Apple", "Banana"];
foreach ($fruits as $fruit) {
    echo $fruit;
}
?>
```

PHP Arrays

Indexed Array:

```
$colors = ["Red", "Green", "Blue"];
```

Associative Array:

```
$age = ["Amit" => 25, "Rahul" => 30];
```

Multidimensional Array:

```
$marks = [
    ["Maths", 90],
    ["Science", 85]
];
```

PHP Strings

```
$str = "Hello";
echo strlen($str);      // String length
echo strrev($str);      // Reverse
echo strpos($str, "e");  // Position
```

PHP OOPs Concepts

- **Class & Object**
- **Constructor**
- **Inheritance**
- **Encapsulation**
- **Polymorphism**

Example:

```

class Car {
    public $name;
    function __construct($name) {
        $this->name = $name;
    }
    function show() {
        echo "Car name: " . $this->name;
    }
}

$car1 = new Car("BMW");
$car1->show();

```

PHP Forms Handling और Validation

HTML Form:

```

<form method="post" action="form.php">
    Name: <input type="text" name="name">
    <input type="submit">
</form>

```

PHP Code (form.php):

```

<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $name = $_POST['name'];
    if (empty($name)) {
        echo "Name is required";
    } else {
        echo "Hello, $name";
    }
}
?>

```

GET और POST Methods

- **GET:** URL में डेटा भेजता है (सुरक्षित नहीं)
- **POST:** डेटा हिडन होता है (सुरक्षित)

```

$_GET['name']
$_POST['name']

```

PHP Functions

```

function greet($name) {
    return "Hello, $name";
}

echo greet("Amit");

```

यहाँ आपको **Cookies और Sessions** पर आधारित पूरी जानकारी **हिंदी में** दी जा रही है। इस टॉपिक में हम सीखेंगे कि Cookies और Sessions क्या होते हैं, कैसे काम करते हैं, और इन्हें कैसे प्रयोग किया जाता है।

Cookies का परिचय (Introduction to Cookies)

Cookies एक छोटी फ़ाइल होती है जिसे वेब सर्वर ब्राउज़र पर स्टोर करता है। इसका उपयोग यूज़र की जानकारी को क्लाइंट साइड (यूज़र ब्राउज़र) में स्टोर करने के लिए किया जाता है।

उदाहरण: जब आप किसी वेबसाइट में लॉगिन करते हैं और वह आपको लॉगिन रखा रखती है, तो यह Cookies की मदद से होता है।

Cookies कहाँ स्टोर होती हैं? (Storage of Cookies at Client Side)

- Cookies यूज़र के ब्राउज़र में **text file** के रूप में स्टोर होती हैं।
- एक cookie में key-value pair होती है।
- आप Cookies को समय अवधि (expiry time) के साथ सेट कर सकते हैं।

PHP में Cookies का उपयोग कैसे करें?

1. Cookie सेट करना (Set Cookie)

```
<?php
setcookie("user", "Amit", time() + (86400 * 7)); // 7 दिन तक वैध
?>
```

- "user" – Cookie का नाम
- "Amit" – Cookie का वैल्यू
- time() + (86400 * 7) – 7 दिन का टाइम

2. Cookie पढ़ना (Read Cookie)

```
<?php
if(isset($_COOKIE["user"])) {
    echo "User is: " . $_COOKIE["user"];
} else {
    echo "Cookie is not set!";
}
?>
```

3. Cookie डिलीट करना (Delete Cookie)

```
<?php
setcookie("user", "", time() - 3600); // Expire कर दी
?>
```

Cookies की जानकारी का उपयोग (Using Information of Cookies)

Cookies का उपयोग:

- यूज़र को पहचानने के लिए

- वेबसाइट विज़िट ट्रैक करने के लिए
- लॉगिन की स्थिति (login state) याद रखने के लिए किया जाता है।

Sessions का परिचय (Introduction to Sessions)

Session एक तरीका है server पर यूज़र की जानकारी स्टोर करने का। Cookies क्लाइंट पर स्टोर होती हैं जबकि **Sessions** सर्वर पर स्टोर होती हैं।

Sessions ज़्यादा सुरक्षित होते हैं क्योंकि डाटा ब्राउज़र में नहीं बल्कि सर्वर में स्टोर होता है।

PHP में Session का उपयोग कैसे करें?

1. Session स्टार्ट करना (Start Session)

```
<?php
session_start();
$_SESSION["username"] = "Amit";
?>
```

2. Session एक्सेस करना (Access Session Data)

```
<?php
session_start();
echo $_SESSION["username"];
```

3. Multiple Sessions बनाना

Sessions associative array की तरह होते हैं, आप कई values store कर सकते हैं:

```
<?php
session_start();
$_SESSION["user_id"] = 101;
$_SESSION["role"] = "admin";
?>
```

4. Session डिलीट करना (Unset/Destroy Session)

```
<?php
session_start();
session_unset(); // सभी session variables को हटाता है
session_destroy(); // पूरा session समाप्त करता है
?>
```

Session Timeout सेट करना

Session timeout से मतलब है - एक निश्चित समय बाद session को एक्सपायर कर देना।

तरीका 1: PHP.INI में सेटिंग

```
session.gc_maxlifetime = 1800 // 30 मिनट
```

तरीका 2: कोड में कस्टम Timeout

```
<?php
session_start();
$expireAfter = 30; // 30 minutes

if(isset($_SESSION['last_action'])) {
    $secondsInactive = time() - $_SESSION['last_action'];
    $expireAfterSeconds = $expireAfter * 60;

    if($secondsInactive >= $expireAfterSeconds) {
        session_unset();
        session_destroy();
        echo "Session expired due to inactivity.";
    }
}
$_SESSION['last_action'] = time();
?>
```

Cookies और Sessions में अंतर

विशेषता	Cookies	Sessions
स्टोरेज स्थान	क्लाइंट ब्राउज़र	सर्वर साइड
सुरक्षा	कम सुरक्षित	ज्यादा सुरक्षित
डेटा आकार	छोटा (4KB तक)	कोई सख्त सीमा नहीं
एक्सपायरी	निर्धारित की जा सकती है	ऑटोमैटिक / कस्टम

निष्कर्ष (Conclusion)

- **Cookies और Sessions** वेबसाइट में यूज़र की जानकारी को ट्रैक और प्रबंधित करने के लिए आवश्यक हैं।
- Cookies क्लाइंट साइड में स्टोर होती हैं, Sessions सर्वर साइड में।
- Session timeout बहुत जरूरी होता है लॉगिन सिक्योरिटी के लिए।

Lecture 2

PHP and MySQL

यहाँ आपको MySQL और PHP के साथ Database Operations पर आधारित एक पूर्ण हिंदी कंटेंट सोलूशन दिया जा रहा है। इसमें आपको मिलेगा:

1. MySQL का परिचय
2. PHP से MySQL से कनेक्शन
3. डेटाबेस बनाना
4. डेटा इनसर्ट, डिलीट और रिट्रीव करना

MySQL का परिचय (Introduction to MySQL)

MySQL एक ओपन-सोर्स रिलेशनल डेटाबेस मैनेजमेंट सिस्टम (RDBMS) है। इसमें डेटा को टेबल्स में स्टोर किया जाता है।

यह Structured Query Language (SQL) का उपयोग करता है:

- डेटा स्टोर करने के लिए
- डेटा को जोड़ने, हटाने, अपडेट करने के लिए

MySQL का उपयोग PHP, Java, Python जैसी भाषाओं के साथ आसानी से किया जा सकता है।

PHP से MySQL से कनेक्ट करना (Connecting to MySQL using PHP)

PHP में MySQL से कनेक्ट करने के लिए हम `mysqli_connect()` या PDO का उपयोग करते हैं।

उदाहरण: `mysqli_connect()` द्वारा

```
<?php
$host = "localhost";
$username = "root";
$password = "";
$dbname = "mydatabase";

// कनेक्शन बनाना
$conn = mysqli_connect($host, $username, $password, $dbname);

// कनेक्शन चेक करना
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}
```

```
}  
echo "Connected successfully";  
?>
```

Database बनाना (Creating Database)

```
<?php  
$conn = mysqli_connect("localhost", "root", "");  
  
// डेटाबेस बनाना  
$sql = "CREATE DATABASE mydatabase";  
if (mysqli_query($conn, $sql)) {  
    echo "Database created successfully";  
} else {  
    echo "Error creating database: " . mysqli_error($conn);  
}  
?>
```

Table बनाना (Create Table in MySQL)

```
<?php  
$conn = mysqli_connect("localhost", "root", "", "mydatabase");  
  
$sql = "CREATE TABLE users (  
    id INT(6) AUTO_INCREMENT PRIMARY KEY,  
    name VARCHAR(50),  
    email VARCHAR(50)  
)";  
if (mysqli_query($conn, $sql)) {  
    echo "Table created successfully";  
} else {  
    echo "Error creating table: " . mysqli_error($conn);  
}  
?>
```

डेटा इनसर्ट करना (Inserting Data into MySQL using PHP)

```
<?php  
$conn = mysqli_connect("localhost", "root", "", "mydatabase");  
  
$sql = "INSERT INTO users (name, email) VALUES ('Ravi', 'ravi@gmail.com')";  
if (mysqli_query($conn, $sql)) {  
    echo "Data inserted successfully";  
} else {  
    echo "Error: " . mysqli_error($conn);  
}  
?>
```

डेटा डिलीट करना (Deleting Data in MySQL using PHP)

```
<?php  
$conn = mysqli_connect("localhost", "root", "", "mydatabase");  
  
$sql = "DELETE FROM users WHERE id = 1";  
if (mysqli_query($conn, $sql)) {  
    echo "Data deleted successfully";  
} else {  
    echo "Error: " . mysqli_error($conn);  
}
```

```
}  
?>
```

डेटा प्राप्त करना (Retrieving Data from MySQL using PHP)

```
<?php  
$conn = mysqli_connect("localhost", "root", "", "mydatabase");  
  
$sql = "SELECT * FROM users";  
$result = mysqli_query($conn, $sql);  
  
if (mysqli_num_rows($result) > 0) {  
    while($row = mysqli_fetch_assoc($result)) {  
        echo "ID: ".$row["id"]." - Name: ".$row["name"]." - Email: ".  
$row["email"]."<br>";  
    }  
} else {  
    echo "No data found";  
}  
?>
```

डेटा अपडेट करना (Updating Data in MySQL)

```
<?php  
$conn = mysqli_connect("localhost", "root", "", "mydatabase");  
  
$sql = "UPDATE users SET name='Amit' WHERE id=2";  
if (mysqli_query($conn, $sql)) {  
    echo "Record updated successfully";  
} else {  
    echo "Error updating record: " . mysqli_error($conn);  
}  
?>
```

निष्कर्ष (Conclusion)

क्र.सं	कार्य	PHP फंक्शन
1	कनेक्शन बनाना	mysqli_connect()
2	डेटाबेस बनाना	CREATE DATABASE
3	टेबल बनाना	CREATE TABLE
4	डेटा जोड़ना	INSERT INTO
5	डेटा पढ़ना	SELECT
6	डेटा हटाना	DELETE
7	डेटा अपडेट करना	UPDATE

Lecture 3

AJEX

AJAX का परिचय (Introduction to AJAX)

AJAX का मतलब है:

Asynchronous JavaScript And XML

AJAX एक तकनीक है जो वेब पेज को बिना रीलोड किए **बैकएंड सर्वर से डेटा** भेजने और प्राप्त करने की सुविधा देती है।

AJAX का उपयोग वेबसाइट को अधिक इंटरैक्टिव और तेज़ बनाने के लिए किया जाता है।

AJAX कैसे काम करता है? (How AJAX Works?)

1. यूज़र किसी इवेंट पर एक AJAX request भेजता है (जैसे बटन क्लिक)
2. JavaScript में XMLHttpRequest ऑब्जेक्ट बनाया जाता है
3. यह request सर्वर पर जाती है (जैसे PHP या API)
4. सर्वर response देता है (JSON, Text, HTML, etc.)
5. पेज को रीलोड किए बिना JavaScript द्वारा परिणाम दिखाया जाता है

XMLHttpRequest Object क्या है?

XMLHttpRequest AJAX की main object है जिससे हम:

- सर्वर से Request भेजते हैं
- Response प्राप्त करते हैं

XMLHttpRequest के Methods:

- open(method, url, async) – request खोलता है
- send() – request भेजता है
- onreadystatechange – response ट्रैक करता है

AJAX का सिंटैक्स (Syntax Example)

```
let xhttp = new XMLHttpRequest();
xhttp.onreadystatechange = function() {
  if (this.readyState == 4 && this.status == 200) {
    document.getElementById("result").innerHTML = this.responseText;
  }
};
xhttp.open("GET", "data.php", true);
xhttp.send();
```

यहाँ data.php सर्वर स्क्रिप्ट है जो response भेजती है।

Server Response समझना (Understanding Server Response)

AJAX response को हम JavaScript में इस तरह पढ़ते हैं:

- responseText: सामान्य टेक्स्ट या HTML
- responseXML: XML डेटा
- JSON.parse(responseText): JSON को JavaScript object में बदलना

AJAX Events (AJAX में इवेंट्स)

Event	उपयोग
onreadystatechange	जब readyState बदले
onload	जब request पूरी हो
onerror	एरर आने पर
onprogress	डेटा ट्रांसफर के समय

AJAX से Form Validation (AJAX with Validation)

उदाहरण: यूज़रनेम पहले से मौजूद है या नहीं चेक करना

HTML:

```
<input type="text" id="username" onkeyup="checkUsername(this.value)">
<span id="msg"></span>
```

JavaScript (AJAX):

```
function checkUsername(name) {
    let xhttp = new XMLHttpRequest();
    xhttp.onreadystatechange = function () {
        if (this.readyState == 4 && this.status == 200) {
            document.getElementById("msg").innerHTML = this.responseText;
        }
    };
    xhttp.open("GET", "check_username.php?name=" + name, true);
    xhttp.send();
}
```

PHP (check_username.php):

```
<?php
$name = $_GET['name'];
if($name == "admin") {
    echo "Username already taken";
} else {
    echo "Available";
}
```

```
}  
?>
```

AJAX के साथ API Interaction (Working with APIs using AJAX)

JSONPlaceholder API से डेटा लाना:

```
<button onclick="loadData()">Get Data</button>  
<div id="output"></div>  
  
function loadData() {  
  let xhttp = new XMLHttpRequest();  
  xhttp.onreadystatechange = function () {  
    if (this.readyState == 4 && this.status == 200) {  
      let users = JSON.parse(this.responseText);  
      let output = "";  
      users.forEach(function(user) {  
        output += "<p>" + user.name + " - " + user.email + "</p>";  
      });  
      document.getElementById("output").innerHTML = output;  
    }  
  };  
  xhttp.open("GET", "https://jsonplaceholder.typicode.com/users", true);  
  xhttp.send();  
}
```

यहाँ हमने एक **public API** से JSON डेटा लाकर उसे DOM में दिखाया।

निष्कर्ष (Conclusion)

टॉपिक	विवरण
AJAX क्या है	रीलोड किए बिना सर्वर से डेटा एक्सचेंज
XMLHttpRequest क्या है	AJAX की मूल object
Response कैसे प्राप्त करें	responseText, responseXML, JSON
API से इंटरैक्शन	Public या Custom API से डेटा फेच
AJAX Validation	Live username/email check

अगर आप चाहें तो मैं इस पर आधारित **मिनी प्रोजेक्ट** जैसे:

- लाइव सर्च सिस्टम
- कस्टम API से डाटा फेच
- फॉर्म Submit बिना रीलोड के

भी तैयार कर सकता हूँ। बताइए क्या चाहिए?

Lecture 4

WordPress (CMS)

WordPress Basic - हिंदी में पूरा पाठ्यक्रम

1. CMS का परिचय (Introduction to CMS)

CMS (Content Management System) एक ऐसा सॉफ्टवेयर है जो आपको बिना कोडिंग के वेबसाइट बनाने, कंटेंट पब्लिश करने और उसे मैनेज करने की सुविधा देता है।

कुछ पॉपुलर CMS:

- WordPress
- Joomla
- Drupal
- Magento (e-commerce के लिए)

CMS की मुख्य विशेषताएँ:

विशेषता	विवरण
कोडिंग की जरूरत नहीं	टेक्निकल नॉलेज के बिना वेबसाइट बनाना
कंटेंट एडिटिंग आसान	Rich text editor की सहायता से
थीम और प्लगिन्स सपोर्ट	वेबसाइट की डिजाइन और फंक्शनलिटी बढ़ाना
यूज़र मैनेजमेंट	Admin, Editor, Subscriber जैसे roles
SEO फ्रेंडली	Google में रैंक करने में सहायक

2. WordPress का परिचय (Introduction to WordPress)

WordPress एक ओपन-सोर्स CMS है जो PHP और MySQL पर आधारित है।

दुनिया की लगभग 40% वेबसाइटें WordPress पर बनी हुई हैं।
ब्लॉग, बिज़नेस वेबसाइट, पोर्टफोलियो, ई-कॉमर्स - सब WordPress से बनाया जा सकता है।

WordPress के दो प्रकार:

प्रकार	विवरण
WordPress.com	Fully hosted (free और paid plans)
WordPress.org	Self-hosted (खुद का domain और hosting चाहिए)

3. WordPress कैसे काम करता है? (How WordPress Works)

WordPress मुख्यतः 3 चीज़ों पर आधारित होता है:

1. **PHP:** सर्वर-साइड scripting language जो core functionality देती है।
2. **MySQL:** Database जहाँ सारे कंटेंट (पोस्ट, पेज, यूज़र) स्टोर होते हैं।
3. **HTML/CSS/JS:** Frontend design के लिए

Workflow:

- यूज़र ब्राउज़र में वेबसाइट खोलता है
- PHP backend से डेटा लेकर HTML generate करता है
- ब्राउज़र HTML/CSS render करता है

4. WordPress इंस्टॉलेशन (Installation of WordPress)

Step-by-Step WordPress Installation (Locally using XAMPP):

1. **XAMPP डाउनलोड करें**
→ <https://www.apachefriends.org/>
2. **XAMPP इंस्टॉल करें और Apache व MySQL स्टार्ट करें**
3. **WordPress डाउनलोड करें**
→ <https://wordpress.org/download>
4. **WordPress फोल्डर को XAMPP के htdocs फोल्डर में रखें**
जैसे: C:\xampp\htdocs\myblog
5. **MySQL में Database बनाएं**
 - localhost/phpmyadmin खोलें
 - नया database बनाएं जैसे: myblog_db
6. **WordPress इंस्टॉलेशन शुरू करें**
ब्राउज़र में खोलें: localhost/myblog
फिर:
 - भाषा चुनें
 - Database details डालें:
 - DB Name: myblog_db
 - Username: root
 - Password: *खाली छोड़ें*
 - Site Title, Admin Username, Password डालें
 - Install पर क्लिक करें

7. WordPress Site तैयार है!

5. WordPress Dashboard की जानकारी

सेक्शन	कार्य
Dashboard	Overview और अपडेट्स
Posts	ब्लॉग पोस्ट बनाना/एडिट करना
Pages	Static pages जैसे About Us
Appearance	Themes, Widgets, Menus
Plugins	नई सुविधाएँ जोड़ना
Users	यूज़र का मैनेजमेंट
Settings	साइट की बेसिक सेटिंग्स

WordPress का इस्तेमाल क्यों करें?

SEO Friendly
मोबाइल फ्रेंडली थीम्स
हज़ारों फ्री प्लगिन्स
आसान Customization
Multilingual सपोर्ट
Fast Development

निष्कर्ष:

टॉपिक	सारांश
CMS क्या है	Content मैनेजमेंट का आसान तरीका
WordPress	PHP आधारित सबसे लोकप्रिय CMS
कैसे काम करता है	PHP, MySQL और HTML पर आधारित
इंस्टॉलेशन	XAMPP के साथ लोकल इंस्टॉलेशन सरल है
डैशबोर्ड	पूरी वेबसाइट को बिना कोडिंग कंट्रोल करना

अगर आप चाहें तो मैं आपको:

- WordPress में Custom Theme बनाना
- Plugin Development
- WooCommerce के साथ ई-कॉमर्स वेबसाइट

पर भी गाइड कर सकता हूँ।

बताइए, आगे किस टॉपिक पर डिटेल चाहिए?

Post & Pages in WordPress – पूरा कंटेंट सोलूशन (हिंदी में)

1. ब्लॉगिंग का परिचय (Introduction to Blogging)

ब्लॉगिंग का मतलब है अपने विचार, जानकारी, न्यूज़ या टॉपिक-आधारित कंटेंट को इंटरनेट पर पब्लिश करना।

ब्लॉग का मुख्य उद्देश्य:

- जानकारी शेयर करना
- दूसरों से जुड़ना
- ऑनलाइन कमाई करना (Ads, Affiliate, Sponsored Posts)

ब्लॉग का उदाहरण:

www.travelwithme.com/blog

2. Blog Post कैसे बनाएं (Creating Blogs/Post)

1. WordPress Dashboard में जाएं
2. **Posts** → **Add New** पर क्लिक करें
3. Title (शीर्षक) डालें
4. Content एरिया में अपना लेख लिखें
5. Featured Image और Category चुनें
6. Publish पर क्लिक करें

3. Images का उपयोग और Text Wrap करना

इमेज डालना:

- पोस्ट एडिटर में “**Add Media**” पर क्लिक करें
- इमेज अपलोड करें या मीडिया लाइब्रेरी से चुनें
- Insert into post पर क्लिक करें

Text Wrap करना:

- इमेज को क्लिक करें
- Align Left या Align Right चुनें
- अब टेक्स्ट इमेज के चारों ओर आ जाएगा

4. Comments का उपयोग (Using Comments)

WordPress में पोस्ट के नीचे यूज़र कमेंट कर सकते हैं
आप कमेंट को Approve, Reply, या Delete कर सकते हैं

Settings → **Discussion** में जाकर:

- कमेंट Allow/Disallow कर सकते हैं
- Approval सेट कर सकते हैं

5. Post Formats का उपयोग

WordPress कुछ अलग-अलग **Post Formats** देता है जैसे:

Format	उपयोग
Standard	सामान्य लेख
Image	केवल इमेज पोस्ट
Video	वीडियो कंटेंट
Quote	कोई कोटेशन
Gallery	इमेज गैलरी

ये Format Theme पर निर्भर करते हैं।

6. Posts में Linking करना (Internal Linking)

पोस्ट के अंदर किसी अन्य पोस्ट का लिंक जोड़ने के लिए:

- टेक्स्ट को सेलेक्ट करें
- लिंक बटन पर क्लिक करें
- पोस्ट URL या पोस्ट का नाम लिखें → Select करें → Apply

इससे SEO भी बेहतर होता है!

7. Categories और Tags

Categories:

- बड़े टॉपिक जैसे "Technology", "Health"
- नेविगेशन में उपयोगी

Tags:

- Specific टॉपिक जैसे "WordPress", "PHP Tips"
- Post के अंदर डिटेल समझाने में मददगार

Post Add/Edit करते समय Categories और Tags सेट करें

8. Using Smilies (इमोजी/स्माइलीज़ का उपयोग)

WordPress में आप जैसे इमोजी सीधे कंटेंट में डाल सकते हैं।

या : -) जैसे सिंबल टाइप करें, WordPress उसे ऑटोमैटिक इमोजी बना देता है।

9. Links Manager का उपयोग (For External Links)

पहले WordPress में **Links** सेक्शन हुआ करता था, जो अब by default छिपा होता है।

उसे एक्टिव करने के लिए Plugin “Link Manager” इंस्टॉल कर सकते हैं।

इसका उपयोग आप External Websites, Affiliate Links, Partner Links आदि मैनेज करने के लिए कर सकते हैं।

10. WordPress Feeds (RSS Feeds)

- RSS Feed का उपयोग लोग आपके ब्लॉग को Subscribe करने के लिए करते हैं
- हर पोस्ट का automatic feed बनता है:

URL Example:

`www.yourdomain.com/feed`

आप feed को customize करने के लिए plugins का उपयोग कर सकते हैं जैसे:

- WP RSS Aggregator
- Feedzy RSS Feeds

11. Password Protected Posts का उपयोग

पोस्ट को Password Protect करने के लिए:

1. Add/Edit Post में जाएं
2. “Visibility” → “Password Protected” चुनें
3. एक पासवर्ड सेट करें
4. Save/Publish करें

अब यह पोस्ट सिर्फ पासवर्ड डालने के बाद ही पढ़ी जा सकती है।

सारांश टेबल

टॉपिक	कार्य
Blogging	कंटेंट लिखना और पब्लिश करना
Images	इमेज जोड़ना और टेक्स्ट के साथ रैप करना
Comments	यूज़र कमेंट्स का हैंडल करना
Formats	पोस्ट के अलग-अलग प्रकार
Linking	पोस्ट्स के बीच लिंक बनाना

टॉपिक	कार्य
Categories & Tags	कंटेंट को व्यवस्थित करना
Smilies	इमोजी का उपयोग
Links Manager	बाहरी लिंक मैनेज करना
Feeds	सब्सक्रिप्शन और ऑटो अपडेट
Password Protection	पोस्ट को सुरक्षित करना

अगर आप चाहें तो मैं आपको:

- Advance Blogging Techniques in WordPress
- SEO Optimization for WordPress Posts
- Custom Post Types & Taxonomies

पर भी गाइड कर सकता हूँ।

बताइए, अगले किस टॉपिक पर डिटेल चाहिए?

WordPress Themes & Site Appearance Customization (हिंदी में)

Color Scheme Develop करना

Color Scheme आपकी वेबसाइट की पहचान बनती है।

चुनते समय ध्यान दें:

- ब्रांड के अनुसार Primary & Secondary Colors चुनें
- Tools:
 - coolers.co
 - Adobe Color
- WordPress में:
 - Appearance → Customize → Colors

Example:

```
body {
  background-color: #ffffff;
  color: #333333;
}
a {
  color: #0073aa;
}
```

Header Design करना

- Appearance → Customize → Header
- Site Title, Logo, Navigation सेट करें
- Custom CSS से और डिज़ाइन कर सकते हैं

Example:

```
.site-header {  
  background: #f8f8f8;  
  padding: 20px;  
}
```

CSS Horizontal Menus बनाना

style.css में Navigation Menu के लिए Horizontal CSS:

```
nav ul {  
  list-style: none;  
  display: flex;  
}  
nav ul li {  
  margin: 0 15px;  
}
```

Menu सेट करने के लिए:

- Appearance → Menus

Dynamic Menu Highlighting

Menu में Active Page को Highlight करना:

```
.current-menu-item a {  
  color: #ff6600;  
  font-weight: bold;  
}
```

यह WordPress खुद क्लास जोड़ता है, बस CSS से स्टाइल करें।

Navigation Links (Next & Previous)

Theme में Add करें:

```
<?php previous_post_link(); ?>  
<?php next_post_link(); ?>
```

Custom CSS Styling:

```
.nav-links {  
  display: flex;  
  justify-content: space-between;  
}
```

Styling for Print (प्रिंट के लिए स्टाइलिंग)

style.css में प्रिंट स्टाइल:

```
@media print {  
    .site-header, .site-footer, .sidebar {  
        display: none;  
    }  
    body {  
        font-size: 12pt;  
    }  
}
```

Post Meta Data Design करना

पोस्ट के नीचे दिखने वाला Author, Date, Category सेक्शन:

```
<p class="post-meta">  
    By <?php the_author(); ?> on <?php the_time('F j, Y'); ?>  
</p>
```

CSS Styling:

```
.post-meta {  
    font-size: 14px;  
    color: #999;  
}
```

Categories को अलग दिखाना

```
$categories = get_the_category();  
foreach( $categories as $cat ) {  
    echo '<span class="cat-label">' . $cat->name . '</span>';  
}
```

CSS:

```
.cat-label {  
    background: #eee;  
    padding: 4px 8px;  
    margin-right: 5px;  
}
```

"Read More" को कस्टमाइज़ करना

WordPress में:

```
the_content('Continue Reading →');
```

CSS से स्टाइल करें:

```
.more-link {  
    color: #0066cc;  
    font-weight: bold;  
}
```

Date & Time Format करना

Dashboard → Settings → General

Format बदल सकते हैं

या फिर Theme में:

```
the_time('d M Y'); // 05 Apr 2025
```

CSS Styles ढूँढना (Finding CSS Styles)

Tools:

- Browser DevTools (Inspect → Styles)
- WordPress Theme files में `style.css`

Individual Pages बनाना

Dashboard → Pages → Add New

Pages के लिए Template Assign कर सकते हैं:

```
<?php
/*
Template Name: Custom Page Template
*/
get_header(); ?>
<!-- Content -->
<?php get_footer(); ?>
```

Files Upload करना (WordPress Themes के साथ)

- Dashboard → Media → Add New
या
- Theme के `functions.php` में file uploader code जोड़ सकते हैं

Templates, Template Tags और Hierarchy

Template Files:

- `single.php` (Single Post)
- `page.php` (Pages)
- `archive.php` (Category, Tags)

Template Tags (PHP Functions):

```
the_title(), the_content(), the_permalink(), the_time()
```

Template Hierarchy: WordPress decides file based on priority:

```
single-{post-type}.php → single.php → index.php
```


Website Validation (वेबसाइट को Validate करना)

Tools:

- [W3C HTML Validator](#)
- CSS Validator

HTML & CSS Errors को ठीक करें
मोबाइल फ्रेंडली टेस्ट करें (Google Mobile Test)

Know Your Sources (सही सोर्स जानें)

Trusted Resources:

- [wordpress.org](#)
- [developer.wordpress.org](#)
- ThemeForest / GitHub

Unverified Plugins या Themes से बचें

WordPress Site Maintenance

Regular Tasks:

- Backup (UpdraftPlus, Jetpack)
- Update (Themes, Plugins, Core)
- Security (Wordfence, iThemes)
- Speed Optimize (Caching Plugins, Image Compression)

Maintenance Mode:

- Plugins जैसे: “Maintenance”, “WP Maintenance Mode”

Quick Summary (सारांश)

फीचर	काम
Color Scheme	साइट का Look & Feel
Header Design	लोगो और Navigation सेट करना
CSS Menu	Horizontal मेनू बनाना
Dynamic Highlight	Active Page को दिखाना
Navigation Links	Previous/Next पोस्ट लिंक
Print Styling	प्रिंट-फ्रेंडली CSS
Post Meta	लेखक, तारीख, केटेगरी दिखाना
Read More	लिंक कस्टमाइज़ करना
Date Format	तारीख का फॉर्मेट बदलना

फीचर

CSS Styles
Page Templates
Template Tags
Validation
Maintenance

काम

Dev Tools से पता करना
पेज टेम्प्लेट बनाना
PHP Functions का उपयोग
Code की जांच करना
Backup, Security, Update