

Converted Document

Unit 2

Static vs Dynamic Website

What is Website?

Website is the collection of web pages, different multimedia content such as text, images, and videos which can be accessed by the URL which you can see in the address bar of the browser. For example:

How to access Websites?

When we type a certain URL in a browser search bar, the browser requests the page from the Web server and the Web server returns the required web page and its content to the browser. Now, it differs how the server returns the information required in the case of static and dynamic websites.

Types of Website:

Static Website

Dynamic Website

Static Website: In Static Websites, Web pages are returned by the server which are prebuilt source code files built using simple languages such as HTML, CSS, or JavaScript. There is no processing of content on the server (according to the user) in Static Websites. Web pages are returned by the server with no change therefore, static Websites are fast. There is no interaction with databases. Also, they are less costly as the host does not need to support server-side processing with different languages.

Architecture of Static Website

Note: Static does not mean that it will not respond to user actions, These Websites are called static because these cannot be manipulated on the server or interact with databases (which is the case in Dynamic Websites).

Dynamic Website: In Dynamic Websites, Web pages are returned by the server which are processed during runtime means they are not prebuilt web pages but they are built during runtime according to the user's demand with the help of server-side scripting languages such as PHP, Node.js, ASP.NET and many more supported by the server. So, they are slower than static websites but updates and interaction with databases are possible.

Dynamic Websites are used over Static Websites as updates can be done very easily as compared to static websites (Where altering in every page is required) but in Dynamic Websites, it is possible to do a common change once and it will reflect in all the web pages.

Architecture of Dynamic Website

Converted Document

Difference Between Static and Dynamic Websites:

Dynamic Contents from Databases:

To dynamically display content from a database in PHP, you'll need to follow these basic steps:

Connect to the Database: Use PHP's mysqli or PDO to connect to your MySQL database.

Write an SQL Query: Create a query to retrieve the data you need from the database.

Execute the Query: Use PHP to execute the query and fetch the results.

Display the Data: Loop through the results and display them dynamically on your webpage.

Example Code (Using MySQLi)

Let's walk through a simple example where you retrieve and display a list of users from a database.

Step 1: Database Connection

```
<?php

$servername = "localhost";

$username = "root"; // your database username

$password = ""; // your database password

$dbname = "testdb"; // your database name

// Create connection

$conn = new mysqli($servername, $username, $password, $dbname);

// Check connection

if ($conn->connect_error) {

    die("Connection failed: " . $conn->connect_error);

}

?>
```

Step 2: SQL Query to Retrieve Data

Here, we want to get all rows from a table called users.

```
$sql = "SELECT id, name, email FROM users";

$result = $conn->query($sql);
```

Step 3: Execute and Fetch Data

Now, we will check if the query returned any rows and then loop through the results to display them.

Converted Document

```
if ($result->num_rows > 0) {  
  
    // Output data of each row  
  
    while($row = $result->fetch_assoc()) {  
  
        echo "ID: " . $row["id"]. " - Name: " . $row["name"]. " - Email: " . $row["email"]. "<br>";  
  
    }  
  
    } else {  
  
        echo "0 results";  
  
    }  
}
```

Step 4: Close the Database Connection

After displaying the results, always close the database connection.

```
$conn->close();
```

Full Example

Here's the full example combined:

```
<?php  
  
$servername = "localhost";  
  
$username = "root"; // your database username  
  
$password = ""; // your database password  
  
$dbname = "testdb"; // your database name  
  
// Create connection  
  
$conn = new mysqli($servername, $username, $password, $dbname);  
  
// Check connection  
  
if ($conn->connect_error) {  
  
    die("Connection failed: " . $conn->connect_error);  
  
}  
  
$sql = "SELECT id, name, email FROM users";  
  
$result = $conn->query($sql);  
  
if ($result->num_rows > 0) {  
  
    // Output data of each row
```

Converted Document

```
while($row = $result->fetch_assoc()) {  
  
    echo "ID: " . $row["id"]. " - Name: " . $row["name"]. " - Email: " . $row["email"]. "<br>";  
  
}  
  
} else {  
  
    echo "0 results";  
  
}  
  
// Close the connection  
  
$conn->close();  
  
?>
```

Notes:

This example assumes a table named users with columns id, name, and email.

Make sure you replace localhost, root, and other database credentials with your actual values.

Error handling is basic in this example. In production, you would want to handle database errors more gracefully.

PHP - Web Concepts

PHP is a server-side scripting language that is used to create dynamic webpages. It is one of the most popular programming languages for web development. This chapter aims to let you get familiarized with certain important concepts of web application development using PHP.

A web-based application is a collection of webpages. A webpage is mainly created with HTML tags. HTML consists of different HTML tags which are required to define the appearance of page elements like text, image, table, etc. Hence, HTML essentially creates a static webpage.

A Web application is hosted on a HTTP server with PHP module installed. The browser acts as a http client, to establish communication with the server, following HTTP protocol.

How to Add Dynamic Content on a Webpage?

To add dynamic content to a webpage, there are two possibilities.

JavaScript is a client-side scripting language, that can access the HTML document object model and render dynamic content on the client browser. JavaScript code can be embedded in HTML page.

The browser may collect data from the user in the form of HTML form elements and send it to a HTTP server for processing. PHP is a widely used Server-side processing language. PHP script can also be embedded

Converted Document

inside HTML page.

Example

In the following script, JavaScript code embedded in HTML renders the current date as per the client browser, and the PHP code displays the current date as per the server, where this script is hosted.

```
<!DOCTYPE html>

<html>

<body>

<script type="text/JavaScript">

document.write("Client's date :"+Date()+"\n");

</script>

<?php

date_default_timezone_set("Asia/Calcutta");

echo "server's date is " . date("Y-m-d") . "\n";

echo "The time is " . date("h:i:sa");

?>

</body>

</html>
```

PHP can intercept and process the data from HTML forms. This allows you to collect information from your users. The next chapter discusses PHP's form handling.

PHP can be used to interact with databases such as MySQL and PostgreSQL. This allows you to store and retrieve data from your database, and dynamically populate the web pages or to power the web applications. PHP includes mysql, mysqli and PDO extensions for database handling.

PHP can handle the data received from the client with HTTP GET as well as POST methods. We shall discuss in detail, how PHP handles GET/POST methods in one of the latter chapters.

HTTP is a stateless protocol. However, it allows Sessions and cookies to be maintained on server and client respectively. PHP can be used to create and manage sessions and cookies. Sessions allow you to track individual users as they navigate your website, while cookies allow you to store information on the user's computer for later use. In of the subsequent chapters, we shall learn how PHP handles sessions and cookies.

PHP can be used to upload files to your web server. This allows you to create web applications that allow

Converted Document

users to upload files, such as images, videos, or documents.

You can use PHP to create a login page for your website. When the user enters their username and password, PHP can check the database to see if the user is valid. If the user is valid, PHP can log the user in and redirect them to the main page of your website.

Identifying Browser & Platform

PHP creates some useful environment variables that can be seen in the `phpinfo.php` page that was used to setup the PHP environment.

One of the environment variables set by PHP is `HTTP_USER_AGENT` which identifies the user's browser and operating system.

PHP provides a function `getenv()` to access the value of all the environment variables. The information contained in the `HTTP_USER_AGENT` environment variable can be used to create dynamic content appropriate to the browser.

Example

Following example demonstrates how you can identify a client browser and operating system.

NOTE ? The function `preg_match()` is discussed in session.

```
<?php
function getBrowser() {
    $u_agent = $_SERVER['HTTP_USER_AGENT'];
    $bname = 'Unknown';
    $platform = 'Unknown';
    $version = '';
    //First get the platform
    if (preg_match('/linux/i', $u_agent)) {
        $platform = 'linux';
    } elseif (preg_match('/macintosh|mac os x/i', $u_agent)) {
        $platform = 'mac'; }
    elseif (preg_match('/windows|win32/i', $u_agent)) {
        $platform = 'windows';
    }
}
```

Converted Document

```
// Next get the name of the useragent yes seperately and for good reason
if(preg_match('/MSIE/i',$u_agent) && !preg_match('/Opera/i',$u_agent)) {
    $bname = 'Internet Explorer';
    $sub = "MSIE";
} elseif(preg_match('/Firefox/i',$u_agent)) {
    $bname = 'Mozilla Firefox';
    $sub = "Firefox";
} elseif(preg_match('/Chrome/i',$u_agent)) {
    $bname = 'Google Chrome';
    $sub = "Chrome";
} elseif(preg_match('/Safari/i',$u_agent)) {
    $bname = 'Apple Safari';
    $sub = "Safari";
} elseif(preg_match('/Opera/i',$u_agent)) {
    $bname = 'Opera';
    $sub = "Opera";
} elseif(preg_match('/Netscape/i',$u_agent)) {
    $bname = 'Netscape';
    $sub = "Netscape";
}

// finally get the correct version number
$known = array('Version', $sub, 'other');
$pattern = '#(<browser>' . join('|', $known) . '
[ ]+(<version>[0-9.|a-zA-Z.]*)#';
if (!preg_match_all($pattern, $u_agent, $matches)) {
    // we have no matching number just continue }

// see how many we have
$i = count($matches['browser']);
```

Converted Document

```
if ($i != 1) {  
    //we will have two since we are not using 'other' argument  
    yet  
    //see if version is before or after the name  
    if (stripos($u_agent,"Version") < stripos($u_agent,$sub)){  
        $version= $matches['version'][0];  
    } else {  
        $version= $matches['version'][1];  
    }  
    } else {  
        $version= $matches['version'][0];  
    } // check if we have a number  
    if ($version == null || $version == "") {$version = "?";}  
    return array(  
        'userAgent' => $u_agent,  
        'name' => $bname,  
        'version' => $version,  
        'platform' => $platform,  
        'pattern' => $pattern  
    );  
}  
// now try it  
$ua = getBrowser();  
$yourbrowser = "Your browser: " . $ua['name'] . " " .  
$ua['version'] .  
" on " . $ua['platform'] . " reports: <br >" . $ua['userAgent'];  
print_r($yourbrowser); ?>
```

This is producing following result on my machine. This result may be different for your computer depending

Converted Document

on what you are using.

It will produce the following result ?

Your browser: Google Chrome 54.0.2840.99 on windows reports:

Mozilla/5.0 (Windows NT 6.3; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)

Chrome/54.0.2840.99 Safari/537.36

Display Images Randomly

The PHP rand() function is used to generate a random number. This function can generate numbers within a given range. The random number generator should be seeded to prevent a regular pattern of numbers being generated. This is achieved using the srand() function that specifies the seed number as its argument.

Example

Following example demonstrates how you can display different image each time out of four images ?

```
<?php
srand( microtime() * 1000000 );

$num = rand( 1, 4 );

switch( $num ) {

case 1: $image_file = "/php/images/php_image_sample_1.jpg";
break;

case 2: $image_file = "/php/images/php_image_sample_2.jpg";
break;

case 3: $image_file = "/php/images/php_image_sample_3.jpg";
break;

case 4: $image_file = "/php/images/php_image_sample_4.jpg";
break;

}

echo "Random Image : <img src=$image_file />";

?>
```

PHP Warning: A non-numeric value encountered in /home/cg/root/44553/main.php on line 2

It will produce the following result ?

Converted Document

Using HTML Forms

The most important thing to notice when dealing with HTML forms and PHP is that any form element in an HTML page will automatically be available to your PHP scripts.

Example

Try out following example by putting the source code in test.php script.

```
<?php
if( $_POST["name"] || $_POST["age"] ) {
if (preg_match("/^[A-Za-z'-]"/,$_POST['name']) ) {
die ("invalid name and name should be alpha");
}
echo "Welcome ". $_POST['name']. "<br />";
echo "You are ". $_POST['age']. " years old.";
exit();
}
?>

<form action = "<?php <b>$_PHP_SELF</b> ?>" method = "POST">
Name: <input type = "text" name = "name" />
Age: <input type = "text" name = "age" />
<input type = "submit" />
</form>
```

It will produce the following result ?

The PHP default variable `$_PHP_SELF` is used for the PHP script name and when you click "submit" button then same PHP script will be called and will produce following result ?

The `method = "POST"` is used to post user data to the server script. There are two methods of posting data to the server script which are discussed in chapter.

Browser Redirection

The `PHP header()` function supplies raw HTTP headers to the browser and can be used to redirect it to another location. The redirection script should be at the very top of the page to prevent any other part of the page from loading.

Converted Document

The target is specified by the Location: header as the argument to the header() function. After calling this function the exit() function can be used to halt parsing of rest of the code.

Example

Following example demonstrates how you can redirect a browser request to another web page. Try out this example by putting the source code in test.php script.

```
<?php
if( $_POST["location"] ) {
$location = $_POST["location"];
header( "Location:$location" );
exit();
}
?>

<p>Choose a site to visit :</p>

<form action = "<?php <b>$_SERVER['PHP_SELF']</b> ?>" method = "POST"> <select name = "location">.
<option value = "http://www.tutorialspoint.com"> Tutorialspoint.com </option>
<option value = "http://www.google.com"> Google Search Page
</option>
</select>
<input type = "submit" />
</form>
```

It will produce the following result ?

Developing dynamic internet applications

It involves creating websites or web-based applications that can respond to user input, interact with databases, and provide real-time or customized content. These applications typically involve multiple layers, including frontend (client-side), backend (server-side), and the database.

Here is a guide to developing dynamic internet applications, covering the key steps and technologies involved:

1. Define the Project Requirements

Before diving into code, it's crucial to clearly define the application's requirements:

Converted Document

Purpose: What is the application supposed to do? E.g., an online store, a social media platform, a blog, etc.

Features: What specific features are needed? E.g., user authentication, product listings, comments, etc.

User Flow: How will users interact with the application? What are the key actions and pages?

2. Choose Technologies

For dynamic applications, you'll generally use a combination of frontend, backend, and database technologies.

Frontend Technologies (Client-Side)

The frontend is responsible for the look and feel of your application. The user interacts with the frontend.

HTML: Structure of the website.

CSS: Styling the website (layout, design, colors, fonts).

JavaScript: Adding interactivity (e.g., button clicks, form validation).

Frontend Frameworks/Libraries (optional):

React: A popular JavaScript library for building user interfaces.

Vue.js: Another popular JavaScript framework for building dynamic user interfaces.

Angular: A TypeScript-based framework for building complex single-page applications (SPAs).

Backend Technologies (Server-Side)

The backend handles business logic, database interactions, and serves data to the frontend.

Programming Languages:

PHP: Widely used for web development, often with MySQL databases.

Node.js: A JavaScript runtime for building scalable server-side applications.

Python: With frameworks like Django or Flask.

Ruby: With Ruby on Rails.

Backend Frameworks:

Express.js: A minimal and flexible Node.js framework.

Laravel: A PHP framework for building robust applications.

Django: A Python web framework for rapid development.

Database

Databases store data that the application will use, and interact with it based on queries.

Converted Document

Relational Databases (for structured data):

MySQL: A popular open-source relational database.

PostgreSQL: Another powerful open-source relational database.

NoSQL Databases (for unstructured or flexible data):

MongoDB: A popular NoSQL database that stores data in JSON-like format.

Firebase: A real-time NoSQL database by Google.

3. Set Up the Development Environment

Frontend: Choose a text editor (VS Code, Sublime Text, etc.) and install any dependencies or packages needed (e.g., React, Vue, etc.).

Backend: Install server-side software (e.g., Apache for PHP, Node.js for Express).

Database: Install your chosen database (e.g., MySQL, PostgreSQL, MongoDB).

For a simple PHP application, you may install a local server stack like XAMPP (which includes Apache, MySQL, and PHP).

4. Develop the Frontend

Start building the user interface (UI) and user experience (UX). Some common tasks in frontend development are:

Layout: Use HTML and CSS to structure the pages.

Interactivity: Add interactivity using JavaScript or a frontend framework like React or Vue.js. You can create dynamic content that changes without reloading the page (e.g., using AJAX for asynchronous requests).

Responsiveness: Ensure the application is responsive, so it works well on all devices (mobile, tablet, desktop). Use CSS frameworks like Bootstrap or write your own media queries.

Example: Create a simple index.html file with dynamic content loaded through JavaScript.

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
<meta charset="UTF-8">
```

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
<title>Dynamic App</title>
```

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

Converted Document

```
</head>

<body>

<h1>Welcome to Dynamic App</h1>

<div id="content"></div>

<button onclick="loadData()">Load Data</button>

<script>

function loadData() {

fetch('data.php')

.then(response => response.json())

.then(data => {

document.getElementById('content').innerHTML = data.message;

});

}

</script>

</body>

</html>
```

5. Develop the Backend

The backend will serve data to the frontend or perform tasks like authentication.

Example (PHP Backend)

A PHP file (data.php) that returns dynamic data:

```
<?php

// data.php

header('Content-Type: application/json');

// Simulate fetching data from a database

$data = [

'message' => 'Hello, this is dynamic data from PHP!'

];

echo json_encode($data);
```

Converted Document

?>

Example (Node.js Backend)

For Node.js, create a simple server using Express:

```
const express = require('express');

const app = express();

const port = 3000;

app.get('/data', (req, res) => {

  res.json({ message: 'Hello, this is dynamic data from Node.js!' });

});

app.listen(port, () => {

  console.log(`Server is running on http://localhost:${port}`);

});
```

6. Database Integration

Create Database: Design your database with tables for the application (e.g., Users, Products, Posts).

Connect to Database: Use PHP (MySQLi or PDO), Node.js (using an ORM like Sequelize or Mongoose for MongoDB), or other backend technologies to connect to the database.

Example: Fetch data from a MySQL database using PHP:

```
<?php

// database.php

$servername = "localhost";

$username = "root";

$password = "";

$dbname = "testdb";

// Create connection

$conn = new mysqli($servername, $username, $password, $dbname);

if ($conn->connect_error) {

  die("Connection failed: " . $conn->connect_error);

}
```

Converted Document

```
$sql = "SELECT * FROM users";  
  
$result = $conn->query($sql);  
  
if ($result->num_rows > 0) {  
  
    // Fetch the results  
  
    while($row = $result->fetch_assoc()) {  
  
        echo "ID: " . $row["id"]. " - Name: " . $row["name"]. "<br>";  
  
    }  
  
} else {  
  
    echo "No results";  
  
}  
  
$conn->close();  
  
?>
```

7. Testing

Testing is crucial to ensure the app works as expected. There are two main types of testing:

Frontend Testing: Ensure the UI is responsive, functional, and bug-free.

Backend Testing: Test API endpoints and database interactions to ensure data is fetched or submitted correctly.

Use tools like Postman or Insomnia for API testing.

8. Deploy the Application

Once your application is ready, deploy it to a live server:

Frontend: Use hosting platforms like Netlify or Vercel for static sites or dynamic sites with serverless functions.

Backend: Use platforms like Heroku, DigitalOcean, AWS, or Linode for dynamic applications.

Database: Use services like Amazon RDS, Heroku Postgres, or MongoDB Atlas for managed databases.

9. Security Considerations

Authentication: Implement user login and registration (consider using OAuth, JWT tokens, or sessions).

Authorization: Control access based on user roles (admin, user, guest, etc.).

Input Validation: Validate user inputs to avoid security risks like SQL Injection, XSS, etc.

Converted Document

HTTPS: Ensure your application is using HTTPS to secure the data in transit.

Conclusion

Building dynamic internet applications requires careful planning, choosing the right technologies, and following best practices. By combining frontend and backend development, integrating databases, and ensuring security, you can create powerful, scalable, and interactive web applications.

What Exactly is Dynamic Web Application Development?

In this article, you will discover what dynamic web application development is and why it's important to learn about it.

A dynamic web application (DWA) is a web application that requires the use of Javascript to deliver content to the . This can be done using technologies such as PHP, Java, or Python. It goes much further than just a front end which uses a back-end . A DWA can be built either as a single-page application or as a more traditional website with multiple pages and multiple levels of site navigation. Not everyone knows what dynamic web applications are, so we'll quickly break down the term before diving into the benefits.

Dynamic web applications are one of the hottest technologies in IT. There are millions of them across thousands of companies and every month more and more businesses are looking at how they can adopt these innovative technologies. If you are a , freelancer, or web development business, you should be aware that DWAs will be one of the most popular technologies in 2022.

Defining Dynamic Web Application

What is a dynamic web application?

A dynamic web application is a type of application that can change its content, appearance, and functionality in response to user input, system events, and information. Dynamic web applications may make use of features such as load-time data-driven forms, AJAX, databases, or other services that respond to external events.

Dynamic Web App

Dynamic Web Applications can change their functionality without having to rewrite parts of their codebase. They offer developers an opportunity to create applications that can adapt based on user behavior and other external factors. They are made up of three components: the backend component (the database), client-side components, and frontend components. These pieces work together to enable users to interact with a DWA, which can change over time just like it does in any other dynamic application.

Examples of Dynamic Web Applications

Converted Document

A dynamic web application is a web application that can be offered under various configurations, depending on the needs of the user. These applications can present themselves differently to specific users based on authentication and other factors. Some examples of dynamic web applications are social media sites like Facebook, Twitter, and forums. These sites not only show different content to people who log in using different means, but they also provide interactivity between users and allow them to express their personalities by posting pictures or writing comments.

1.

Youtube Web Application

When you search for something on YouTube. YouTube sends a request to the server, which returns results for your search. Following that, new content will be displayed on the website. This is a pretty good example of a dynamic application. YouTube's website content was dynamically updated in response to your input.

2.

Twitter Web App

When you like a tweet on Twitter, it increases the number of likes by one and displays the total count. Twitter calculated the number of likes on the server and delivered the value to the client. The updated result is then displayed by the client.

3.

Medium Web App

It is really simple to publish blogs on Medium. You simply enter content into the editor and press the publish button. Your blog will be published in a matter of seconds, and you will be able to view it immediately after publishing. Consider medium to be a static site where you must submit a draft and wait for them to add that page. See, how fast dynamic web applications are.

And a lot more. There are millions of dynamic web applications on the internet like these.

Difference Between Dynamic and Static Web App

Static and dynamic web apps. They're all taken for granted in the market today. Yet, they differ in what they do and how they work. Understanding their differences is absolutely vital if you want to fully comprehend the endless world of possibilities thematic applications can offer your business.

Static . In static web applications, all the data is stored in the browser. They do not require an Internet connection to work and can be hosted anywhere users have access to a computer or Internet connection.

Static websites may have some advantages over dynamic sites, but have the disadvantage of being difficult

Converted Document

for users to update themselves and analyze performance issues in most cases. There are certain limitations with static websites such as adding features, fixing bugs and importing users' information from another database or application that connects with it.

Despite the fact that static website needs minimal development costs (no database setup), they are more secure as they cannot be modified by bad people easily and if security policies also hide their complexity from them (i.e.), they are easy to use by people who know nothing about yet need their sites working just fine without knowledge of programming languages like JavaScript, HTML, and CSS required for dynamic ones.

Dynamic web apps are well-known as an alternative to static websites. Dynamic web applications are based on models, business rules, and data stored in the backend. Dynamic web apps allow you to respond to user input in real time, so transactions can take place instantly. Static web applications are more traditional, where static pages are viewed unchanged by users each time they visit the website.

Both dynamic and static web apps are used in the development of websites. Dynamic is a type of methodology that makes it easy for you to change the content on your web page. It also allows you to generate new HTML code automatically without saving or saving it into an external file. Dynamic web apps work by using server-side languages to retrieve data from a database or other application, manipulate it and respond in an adequate way. Static web apps do not require any sort of programming, since they only use information from files on disk (browser).

Static website

The page's content will not change unless the site's owner updates it.

The user is unable to interact with the web page, such as commenting on the post, liking the post, searching for content, and so on.

No need for database or server

Less expensive

Dynamic web app

Without the requirement of the site owner, content changes dynamically based on user interaction

The user can interact in a variety of ways.

Need a server and database

Can be expensive to build

Both dynamic and static web apps are used in the development of websites. Understanding their differences is absolutely vital if you want to fully comprehend the possibilities thematic applications can offer your

Converted Document

business. Dynamic web apps allow you to respond to user input in real time, so transactions can take place instantly. Static web applications are more traditional, where static pages are viewed unchanged by users each time they visit the website.

Benefits of dynamic web application

Everything is moving at an incredible speed and we are expecting the future to be even more dynamic and faster than the present. A dynamic web application is one of the most recent innovations in website development. Dynamic web applications can change their design, appearance, and functionality to match their users' expectations or the current context or environment.

The most popular use of dynamic web applications is on mobile smartphones and other small devices that have limited resources for running heavy desktop applications. How can they help in providing a solution to the problems that businesses face these days? So, what are their benefits and how can you take advantage of this innovative technology?

A dynamic web application is a web application that dynamically generates pages based on some pre-established rules. By default, most sites on the internet use static pages. Imagine having to make all those switches manually every time you want to create a page! That would be very time-consuming and resource-consuming for a proper website.

DWA has made it easier for developers and designers to create applications that run across different types of devices. Although there are plenty of technical specifications involved if you take a closer look at dynamic web applications, one thing stands out at first glance that is its functionality and user experience.

Web application development has come a long way in the last few years. Gone are the days of static websites that required you to code everything by hand. Dynamic web applications (DWA) make it possible to develop rapidly while delivering immediate results as soon as they are published.

How? There's no need to build everything from scratch or actually change what is already there once you publish; dynamic web apps push content to live with minimal fuss. As we all know this can be a real boon for businesses that need a website for their business but don't want to spend unnecessary time, money, and resources on developing the site.

Conclusion

A dynamic web application (DWA) is a web application that requires the use of Javascript to deliver content to the browser. It can be built either as a single-page application or as a more traditional website with multiple pages and multiple levels of site navigation.

Converted Document

Dynamic web applications may make use of features such as load-time data-driven forms, AJAX, databases, or other services that respond to external events. Static web apps are websites that are not stored in any database or get updated when the user visits. They do not require an Internet connection to work and can be hosted anywhere users have access to a computer or Internet connection.

Dynamic web apps use server-side languages to retrieve data from a database or other application, manipulate it and respond in an adequate way. A dynamic web application is a web application that dynamically generates pages based on some pre-established rules.

By default, most sites on the internet use static pages; DWA has made it easier for developers and designers to create applications that run across different types of devices. Dynamic web applications can change their appearance and functionality to match their users' expectations or the current context or environment.

Difference between Server Side Scripting and Client Side Scripting

1. Client-side scripting :

execute client-side scripting. It is used when browsers have all code. Source code is used to transfer from to user?s computer over the and run directly on browsers. It is also used for validations and functionality for user events.

It allows for more interactivity. It usually performs several actions without going to the user. It cannot be basically used to connect to databases on a web server. These scripts cannot access the file system that resides in the web browser. Pages are altered on basis of the user?s choice. It can also be used to create ?cookies? that store data on the user?s computer.

2. Server-side scripting :

Web servers are used to execute server-side scripting. They are basically used to create dynamic pages. It can also access the file system residing at the webserver. A server-side environment that runs on a scripting language is a web server.

Scripts can be written in any of a number of server-side scripting languages available. It is used to retrieve and generate content for dynamic pages. It is used to require to download plugins. In this load times are generally faster than client-side scripting. When you need to store and retrieve information a database will be used to contain data. It can use huge resources of the server. It reduces client-side computation overhead. The server sends pages to the request of the user/client.

Difference between client-side scripting and server-side scripting :

Advantages and Disadvantages of PHP

Converted Document

:

The name PHP stands for Hypertext Preprocessor and denotes a server-side scripting language, which suggests that applications are written thereon run on web servers and don't depend upon the online browser. Syntax of PHP language is analogous to C language. It is created by Rasmus Lerdorf and appeared in 1995. PHP is being widely utilized in developing web applications and became one of the main languages for developers to make new applications.

Leading social networking sites like Facebook and reputed organizations like Harvard University are both using PHP which makes PHP popular and increases its credibility.

However, over years, its area of use has shifted, and nowadays PHP coding language is ranked among the simplest and hottest programming tools for web development thanks to its many virtues which can be the main target of this text. It's considered a really effective technology that gives a convenient development process with many additional tools to assist it. In fact, consistent with recognition of programming language Index (PYPL), PHP is the fifth hottest coding language in the world.

PHP is getting used widely in developing web-based and other applications across all domains. Few technologies whose development is supported by PHP are listed below :

Content Management System.

Web-based applications and development of sites.

E-commerce websites and applications.

Data Analytics and Representation.

Processing of Images.

Graphical interface design-based applications.

Developing the features of Flash.

Advantages of PHP :

The most important advantage of PHP is that it's open-source and free from cost. It can be downloaded anywhere and is readily available to use for events or web applications.

It is platform-independent. PHP-based applications can run on any OS like UNIX, Linux, Windows, etc.

Applications can easily be loaded which are based on PHP and connected to the database. It's mainly used due to its faster rate of loading over slow internet speed than other programming language.

It has less learning curve because it is simple and straightforward to use. Someone familiar with C programming can easily work on PHP.

Converted Document

It is more stable for a few years with the assistance of providing continuous support to various versions.

It helps in reusing an equivalent code and not got have to write lengthy code and sophisticated structure for events of web applications.

It helps in managing code easily.

It has powerful library support to use various function modules for data representation.

PHP?s built-in database connection modules help in connecting databases easily reducing trouble and time for the development of web applications and content-based sites.

The popularity of PHP gave rise to various communities of developers, a fraction of which may be potential candidates for hire.

Flexibility makes PHP ready to effectively combine with many other programming languages in order that the software package could use foremost effective technology for every particular feature.

Disadvantages of PHP :

It is not that secure due to its open-source, because the ASCII text file is often easily available.

It is not suitable for giant content-based web applications.

It has a weak type, which can cause incorrect data and knowledge to users.

PHP frameworks got to learn to use PHP built-in functionalities to avoid writing additional code.

Using more features of PHP framework and tools cause poor performance of online applications.

PHP doesn?t allow change or modification in the core behavior of online applications.

The PHP frameworks aren?t equivalent in behavior so does their performance and features.

While PHP may be a powerful tool supported by an outsized community and plentiful reference documentation, there are easier programming languages for web apps.

It is widely believed by the developers that PHP features a poor quality of handling errors. PHP lacks debugging tools, which are needed to look for errors and warnings. PHP has less number of debugging tools in comparison to other programming languages.

It?s highly tough to manage because it?s not competent modular. It already imitates the features of the Java language.

PHP, or Hypertext Preprocessor, is a programming language that can be used to create websites and web applications. It has many capabilities, including:

Dynamic page content: PHP can generate dynamic content for web pages.

Converted Document

File management: PHP can create, read, write, delete, and close files on the server.

Form data: PHP can collect data from forms.

Cookies: PHP can send and receive cookies.

Database management: PHP can add, delete, and modify data in a database.

User access: PHP can be used to control user access.

Data encryption: PHP can encrypt user data.

System administration: PHP can automate system administration tasks.

Email handling: PHP can manage database interactions for handling emails.

PHP is a server-side scripting language that is encoded with HTML. It is compatible with many operating systems, including Linux, Unix, and Windows.

PHP has many advantages, including:

Open-source: PHP is free to use, including the language, frameworks, debugging tools, and databases.

Community support: There is extensive community support for PHP.

Database connectivity: PHP has good database connectivity.

Flexibility: PHP is flexible.

Speed: PHP is fast.

Cloud services: PHP is compatible with cloud services.

Library support: PHP has extensive library support.

When comparing PHP (Hypertext Preprocessor) and ASP (Active Server Pages), there are several factors to consider, including ease of use, performance, compatibility, community support, and use cases. Both are powerful server-side scripting languages that can be used to create dynamic websites and web applications, but they have different ecosystems and strengths. Here's a breakdown of the key differences between PHP and ASP:

1. Overview of PHP and ASP

PHP:

A popular open-source server-side scripting language, primarily used for web development.

PHP is platform-independent and works on almost any operating system, including Linux, Windows, and macOS.

PHP is often used with MySQL or MariaDB for database-driven websites and applications.

Converted Document

ASP (more specifically ASP.NET today):

ASP was originally developed by Microsoft in the late 1990s to create dynamic web pages on Windows servers.

ASP.NET is the modern version of ASP, built on the .NET framework, and is used for building web applications in a more structured and object-oriented way.

ASP.NET is typically used on Windows servers and is a part of the larger Microsoft ecosystem.

2. Language Syntax and Usage

PHP:

PHP is a relatively simple language with easy syntax, making it a good choice for beginners.

PHP code is embedded directly within HTML code (using `<?php ... ?>` tags).

It has a rich set of built-in functions and is known for being highly flexible.

ASP.NET:

ASP.NET is object-oriented and heavily reliant on the .NET framework, making it more structured.

It uses languages like C# or VB.NET for scripting, making it more versatile but also more complex than PHP.

In ASP.NET, you write code in separate files (code-behind files) and HTML files. It's based on the Model-View-Controller (MVC) architecture, which encourages better design patterns.

3. Platform and Hosting

PHP:

Cross-Platform: PHP works on various platforms, including Linux, Windows, and macOS. This makes it very versatile for hosting on different servers.

Web Hosting: PHP is supported by almost all web hosting providers. Shared hosting platforms and inexpensive VPS solutions typically offer PHP support.

ASP.NET:

Primarily Windows-based: Historically, ASP.NET has been tied to Microsoft technologies and works best on Windows servers. However, with the advent of .NET Core, it has become cross-platform, allowing ASP.NET applications to run on Linux and macOS as well.

Web Hosting: Hosting ASP.NET applications generally requires Windows-based hosting or cloud services like Microsoft Azure. Though ASP.NET Core can be hosted on Linux, it's often easier with Windows.

4. Performance

Converted Document

PHP:

PHP performs well for most small-to-medium-sized applications. Its performance has improved over the years, particularly with recent versions (PHP 7.x and beyond).

However, PHP is generally slower than compiled languages like C# in ASP.NET, especially for very high-traffic applications, although the difference may not always be noticeable in typical use cases.

ASP.NET:

ASP.NET (especially ASP.NET Core) tends to outperform PHP, especially for large, complex web applications.

Since ASP.NET applications are compiled into machine code before being executed, they often offer better performance in terms of speed and memory management.

5. Learning Curve

PHP:

PHP has a lower learning curve, especially for beginners, as it is easier to get started with basic web development.

There are many online resources, tutorials, and documentation available to help new developers.

ASP.NET:

ASP.NET has a steeper learning curve because of its reliance on the .NET ecosystem, its structure, and the use of object-oriented programming (OOP) principles.

However, the MVC pattern (Model-View-Controller) makes it easier to scale large applications, and developers with experience in object-oriented programming or C# will find ASP.NET more intuitive.

6. Community Support and Ecosystem

PHP:

PHP has a large, active, and mature community, with a vast number of resources, tutorials, and open-source frameworks available.

Popular frameworks like Laravel, Symfony, and CodeIgniter are built on PHP and help developers create scalable, maintainable applications.

There are countless open-source content management systems (CMS) like WordPress, Drupal, and Joomla built with PHP.

ASP.NET:

Converted Document

ASP.NET has a strong Microsoft-backed ecosystem, with comprehensive documentation, tutorials, and a well-established community of developers.

Many large-scale enterprise applications are built with ASP.NET.

It integrates well with other Microsoft products like Azure, SQL Server, and Visual Studio, which may make it appealing to organizations already using Microsoft technologies.

ASP.NET Core (the open-source, cross-platform version) has gained significant traction in the developer community.

7. Security

PHP:

PHP provides built-in features for handling security, such as filtering inputs, password hashing, and handling sessions.

However, because of PHP's flexibility and wide adoption, it is often targeted by hackers. Developers need to ensure that they follow best practices to avoid common security risks like SQL injection and cross-site scripting (XSS).

ASP.NET:

ASP.NET provides strong security features, including built-in authentication and authorization mechanisms, cross-site scripting (XSS) and cross-site request forgery (CSRF) prevention, and input validation.

Being part of the Microsoft ecosystem, ASP.NET benefits from regular security updates and patches, making it a reliable choice for enterprise-level applications.

8. Cost and Licensing

PHP:

PHP is open-source and free to use, which makes it an attractive choice for developers and small businesses.

It is also supported by most low-cost or shared hosting providers, making it very cost-effective for smaller projects.

ASP.NET:

ASP.NET is also free to use, especially with ASP.NET Core, which is open-source.

However, if you're using the full .NET ecosystem, hosting might be more expensive since it may require Windows servers or specific Microsoft services like Azure.

Converted Document

For enterprise solutions, the costs can add up due to licensing requirements for tools like SQL Server and other Microsoft services.

9. Popular Use Cases

PHP:

Content Management Systems (CMS): WordPress, Joomla, and Drupal are all built with PHP.

Small to Medium Web Applications: Websites and applications that require rapid development and don't need highly complex back-end systems.

E-commerce Platforms: Platforms like WooCommerce (WordPress) and Magento are built on PHP.

ASP.NET:

Enterprise Applications: Large-scale, complex business applications often use ASP.NET due to its robust framework, scalability, and integration with other Microsoft products.

High-Performance Web Applications: Applications that require more performance, like gaming platforms, banking systems, or high-traffic websites.

Web APIs: ASP.NET Core is often used to build RESTful APIs for backend services and mobile apps.

Conclusion

Both PHP and ASP.NET are excellent choices for developing dynamic web applications, but they cater to different needs and environments.

PHP is a great choice for:

Open-source projects.

Small-to-medium-sized web applications.

Developers looking for ease of use and quick development.

Cost-effective hosting on shared servers.

ASP.NET is ideal for:

Large, enterprise-level applications.

Developers already in the Microsoft ecosystem.

Performance-critical applications.

Applications requiring robust security and integration with other Microsoft products.

Ultimately, your choice depends on the project's specific requirements, your familiarity with the technologies, and the environment you're working in.

Converted Document

Difference between PHP and ASP.NET

1. : Hypertext Preprocessor (PHP) is a server side coding/programming language and its first release was on November, 1997. It is created by Rasmus Lerdorf. It supports small to medium size web solutions. This PHP can be embedded into . It is suitable for

Independent Software Vendors (ISVs)

Small or medium business websites

Billing/Invoicing websites

Startups, CRMs etc.

Advantages of PHP :

It is suitable for huge projects.

Easy communication with different databases.

It is highly customizable.

It is secure and protects websites from security attacks.

It can be learned easily.

Disadvantages of PHP :

It is not suitable for the development of desktop applications.

There is a lack of opportunity to modify core behavior.

Too much customization gives rise to bugs.

2. : ASP.NET is a web application framework and its first release was on January, 2002. It is created by Microsoft. It supports large to medium size enterprise applications. It is suitable for

Enterprise level websites

Enterprise level CRMs

ERP Apps

Console applications

Advantages of ASP.NET :

It is highly scalable also before compilation notifies error.

It is suitable for enterprise or windows application development.

Provides best UI for developers.

Converted Document

It includes built-in caching features.

It is supported by all programming languages.

Disadvantages of ASP.NET :

Little difficult to learn.

It is comparatively costly.

It is based on closed source technology.

It has low community so less community support.

Difference between PHP and ASP.NET :