

កម្រងសំណួរចម្លើយមុខវិជ្ជា Web Development Application

1. **Web Application** ជាការសំដៅទៅលើ Software Application ទាំងឡាយណាដែលប្រើប្រាស់នៅលើ Web Browser ធ្វើជា Client។ Web Application ត្រូវបានរចនាដោយដំណើការនៅលើ Website ហើយអាចចូលទៅប្រើប្រាស់តាមរយៈ Internet។

❖ **សារៈប្រយោជន៍នៃ Web Application៖**

- **Accessible From Anywhere:** Web application ផ្តល់នូវលិទ្ធភាពអោយ Client or End-User អាចចូលប្រើប្រាស់បានគ្រប់ទីកន្លែង ដោយគ្រាន់តែភ្ជាប់នឹង Internet។
- **Cross-platform Compatibility:** Client or End-User អាចប្រើប្រាស់ Web Application បានគ្រប់ប្រភេទនៃ Devices រួមមាន៖ Desktop Computer, Laptop, Tablet, or Smart និងគ្រប់ Operating System។
- **Scalable:** ការ Hosting Web Application នៅលើ Server ឬ Cloud បង្កភាពងាយស្រួលដល់ការបន្ថែម ឬបន្ថយនៅ features ទៅតាមតម្រូវការជាក់ស្តែង ដោយមិនបញ្ហាកើតឡើងដល់ Performance នៃ Web Application បើធៀបនឹង Desktop Application។
- **Easy Update:** រាល់ពេលធ្វើការ Update features ដែលមាននៅលើ Web Application មិនត្រូវបានទាមទារអោយមានការតម្លើង Software ណាមួយឡើយ ផ្ទុយទៅវិញការ Update មានលក្ខណៈងាយស្រួល និងលឿនរហ័ស។
- **Cost-Effective:** ការចំណាយទៅលើការអភិវឌ្ឍន៍ Web Application មានតម្លៃសមរម្យ និងងាយស្រួលក្នុងការថែទាំ បើធៀបទៅនឹងការអភិវឌ្ឍន៍ Desktop Application ហើយមិនការតម្រូវអោយធ្វើការតម្លើង Hardware and Software នោះទេ។

❖ **ភាពខុសគ្នារវាង Web Dynamic Client និង Web Dynamic Server៖**

Web Dynamic Client	Web Dynamic Server
Code ត្រូវបាន run នៅលើ User Computer	Code ត្រូវបាន run នៅលើ Web Server
ប្រើប្រាស់ HTML, CSS, and JavaScript	ប្រើប្រាស់ PHP, Python, Ruby, or ASP.Net
Source Code អាចមើលបាន	Source Code មិនអាចមើលបាន
Less Security	More Security
More interactive and personalized	More scalable and performant

❖ **និយាយពីដំណើរការ Process របស់ Web Dynamic Server:**

- A Client or End-User requests a web page from a web browser
- The web browser sends a request to the web server
- The web server receives the request and parses it
- The web server then executes the code that generates the web page
- The web server then sends the generated web page back to web browser
- The web browser then displays the web page to the client or end-user

2. **Web Server** គឺជា application software ដែលបម្រើដល់ការបើក និងបិទ web page ទៅតាមសំណើ របស់ client or end-user ដែលបានធ្វើការ request ទៅកាន់ web server តាមរយៈ web browser។ Web Server ត្រូវបានគេប្រើជាចម្បងក្នុងការ hosting website។ ម្យ៉ាងវិញទៀត web server ក៏ត្រូវ បានប្រើដើម្បីចែកចាយមាតិកាផ្សេងទៀតដូចជា៖ email, file sharing and streaming media ។

❖ **សារៈប្រយោជន៍នៃ Web Server:**

- **Scalability:** Web server អាចដោះស្រាយនៅ request ជាច្រើនដែល client or end-user បាន sent request មកក្នុងពេលតែមួយ។
- **Security:** Web server បានផ្តល់នូវវិធានការការពារផ្សេងៗរួមមាន៖ SSL/TLS encryption, firewalls, and access control ដើម្បីការពារទិន្នន័យនៃ Website or Web Content ពីការវាយ ប្រហារផ្សេងៗពីសំណាក់ Hacker និងការចូលទៅប្រើប្រាស់ទិន្នន័យដោយគ្មានការអនុញ្ញាត។
- **Performance:** Web server អាចធ្វើអោយ performance កាន់តែប្រសើរឡើង ដើម្បីធានាអោយការបើក Website កាន់តែឆាប់រហ័សជាងមុន។ ការបើក Website កាន់តែឆាប់រហ័សជាង មុនទាមទារអោយមានការប្រើប្រាស់នៅបច្ចេកទេសជាច្រើនដូចជា caching, compressing file, and load balancing។
- **Reliability:** Web server បានផ្តល់នូវភាពទុកចិត្តខ្ពស់ដល់ដំណើរការ website។ នេះមាន ន័យថា website របស់យើងនៅតែអាចដំណើរការបាន បើទោះបីជាមានបញ្ហាជាមួយនឹង Internet connection or Web server hardware ក៏ដោយ។

3. Database Server គឺជា application software ដែលគ្រប់គ្រងលើការប្រើប្រាស់ database និងផ្តល់នូវ database service ទៅអោយ client application ទាំងឡាយណាដែលត្រូវការប្រើប្រាស់ជាមួយនឹង database។ Database server អាចដំណើរការ physical server ដាច់ដោយឡែក ឬអាចត្រូវបានតម្លើងនៅលើ server តែមួយដូចគ្នាទៅនឹង client application។

❖ **សារៈប្រយោជន៍នៃ Database Server:**

- **Centralized data management:** Database server ផ្តល់នូវ centralized location ដើម្បីរក្សាទុក និងគ្រប់គ្រងទិន្នន័យ ដែលអាចចូលទៅប្រើប្រាស់បានដោយ client application ជាច្រើនក្នុងពេលដំណាលគ្នា។
- **Scalability:** Database server អាចគ្រប់គ្រងបរិមាណដ៏ធំនៃទិន្នន័យ ហើយទិន្នន័យនោះអាចនឹងមានការប្រែប្រួលទៅតាមតម្រូវការ។
- **Security:** Database server ផ្តល់នូវមុនងារសុវត្ថិភាពដូចជា User Authentication and access control ដើម្បីការពារទិន្នន័យពីការចូលទៅប្រើប្រាស់ដោយគ្មានការអនុញ្ញាត។
- **Data integrity:** Database server អាចពង្រឹងការរឹតត្បិតទិន្នន័យ និងធានាថាទិន្នន័យទាំងអស់ត្រូវបានរក្សាទុកយ៉ាងត្រឹមត្រូវនិងជាប់លាប់។
- **Performance:** Database server បានធ្វើអោយការទាញយកទិន្នន័យ និងការផ្ទុកទិន្នន័យកាន់តែល្អប្រសើរជាងមុន និងកាត់បន្ថយពេលវេលាដែលត្រូវចូលទៅប្រើប្រាស់ទិន្នន័យ។
- **Backup and Recovery:** Database server បានផ្តល់នូវការ Backup and Recovery Service ដើម្បីការពារទិន្នន័យពី Hardware Failures ឬគ្រោះមហន្តរាយផ្សេងៗ។

4. PHP ជា Server-side script language ដែលត្រូវបានគេប្រើប្រាស់យ៉ាងទូលំទូលាយក្នុងការអភិវឌ្ឍន៍ Dynamic Web Page and Dynamic Web Application។ PHP code ត្រូវ run នៅលើ server។

❖ **សារៈប្រយោជន៍នៃ PHP:**

- **Easy to learn:** PHP ជាភាសាដែលមាន Simple Syntax និងងាយស្រួល ដែលអាចអោយ Beginner អាចរៀនបាន។
- **Cross-platform compatibility:** PHP អាច run នៅលើ Operating System ផ្សេងៗដូចជា Windows, Linux, and MacOS ហើយ PHP ក៏អាចប្រើជាមួយនឹង Web Servers ជាច្រើនរួមមាន Apache and Nginx។

- **Open-source and Free:** PHP ជា open-source language ដែលអាចប្រើប្រាស់ដោយឥតគិតថ្លៃ ។
- **Large developer community:** PHP មាន large Community និង Active ដែលរួមចំណែកដល់ការ develop, share knowledge, and provide support forum documentation, and tutorials ។
- **Support various databases:** PHP អាចដំណើរការជាមួយនិង databases ដ៏ពេញនិយមរួមមាន MySQL, Oracle, and PostgreSQL ។
- **Flexibility:** PHP ត្រូវបានគេប្រើសម្រាប់ដើម្បីបង្កើតនូវ Web Applications ផ្សេងៗរួមមាន Dynamic Website, E-Commerce Platforms, Content Management Systems, and Web-Base Application ។
- **High-performance:** PHP Performance ត្រូវបានធ្វើអោយកាន់តែប្រសើរឡើង ដោយបច្ចេកទេសដូចជា Caching, minimizing database queries, and optimizing code ។

5. Software ចាំបាច់សម្រាប់ Run PHP៖

- ❖ Apache
- ❖ Nginx
- ❖ IIS (Internet Information Services)
- ❖ XAMPP
- ❖ WAMPP
- ❖ LAMP
- ❖ MAMP

6. ទម្រង់ទូទៅនៅក្នុងការ Develop PHP Web Application៖

- ❖ **Planning and Analysis:** កំណត់ Requirements, កំណត់ Scope និងការវិភាគលទ្ធភាពនៃ Project ។
- ❖ **Design:** Design UI (User Interface), functionality, design database schema, and design the outline the architecture ។

- ❖ **Development:** Writing code for the web application including the Server-Side PHP code, Client-Side HTML, CSS, Bootstrap and JavaScript។
- ❖ **Testing:** Testing for bugs, errors, usability issues, and making any necessary revisions។
- ❖ **Deployment:** Deploying to web server and Configuring it for use by client or end-users។
- ❖ **Maintenance and Update:** Maintaining and Updating over time to ensure its continued functionality and security។

7. លក្ខណៈពិសេសរបស់ Variable នៅក្នុង PHP ៖

- ❖ **Case-Sensitivity:** រាល់ការប្រកាស Variable នៅក្នុង PHP ត្រូវតែចាប់ផ្តើមដោយអក្សរ ឬ Underscore និងអាចផ្ទុកបានតែ Alpha-Numeric Characters។
- ❖ **No Data Type Declaration:** រាល់ការប្រកាស Variable នៅក្នុង PHP មិនតម្រូវអោយមានការប្រកាសភ្ជាប់ជាមួយនឹង Data types ដូចជា៖ int, string, float, double, Boolean... នោះទេ។
- ❖ **Start with Dollar Sign:** រាល់ការប្រកាស Variable នៅក្នុង PHP ត្រូវចាប់ផ្តើមដោយសញ្ញា \$។
- ❖ **Can store different Data Types:** រាល់ Variable នីមួយៗនៅក្នុង PHP អាចផ្ទុក Data types ផ្សេងៗគ្នាដូចជា strings, integers, floats, Booleans, arrays, and objects។
- ❖ **Scope:** រាល់ Variable នីមួយៗនៅក្នុង PHP អាចធ្វើប្រើប្រាស់បានគ្រប់ទីកន្លែង ដោយគ្រាន់តែប្រកាសជា Local Scope or Global Scope។

8. Array នៅក្នុង PHP មាន 3 មានដូចជា៖ Indexed Arrays, Associative Arrays, and Multidimensional Arrays។

❖ ឧទាហរណ៍នៃ Array នីមួយៗមានដូចខាងក្រោម៖

- **Indexed Array use numbers as keys to access their elements។**

```
// Indexed array
$array = array(1, 2, 3, 4, 5);
```

- **Associative Array use string as keys to access their elements។**

```
// Associative array
$array = array("name" => "John Doe", "age" => 30, "gender" => "male");
```

- **Multidimensional Array is an that contain other arrays as elements។**

```
// Multidimensional array
$array = array(
    array(1, 2, 3),
    array(4, 5, 6),
    array(7, 8, 9)
);
```

9. របៀបបង្កើត Function នៅក្នុង PHP៖

❖ Using “function” Keyword

➤ Syntax:

```
function function_name([parameter1, parameter2, ...]) {  
    // Body of the function  
}
```

- “function” keyword is used to declared a function in PHP.
- “function_name” is the name of the function.
- “[parameter1, parameter2, ...]” are the parameters that the function accepts.
- “Body of the function” is the code that will be executed when the function is called.

➤ Example:

```
function add_numbers($number1, $number2) {  
    $sum = $number1 + $number2;  
    return $sum;  
}
```

➤ Call the “add_numbers()” function:

```
$result = add_numbers(10, 20);  
  
echo $result; // 30
```

❖ Using “fn” Keyword:

```
$result = fn($number1, $number2) => $number1 + $number2;  
  
echo $result(10, 20); // 30
```

- “fn” Keyword is a new feature in PHP 8. It can be used to declared a function inline, without having to declare it beforehand. The syntax above is declaring a function with the “fn” keyword.

10. របៀបបង្កើត Class នៅក្នុង PHP ៖

❖ Using “class” Keyword៖

➤ Syntax៖

```
class ClassName {  
    // Body of the class  
}
```

- “class” Keyword is used to declare a class in PHP.
- “className” is the name of the class.
- “Body of the class” is the code that will be executed when the class is instantiated.

➤ Example៖

```
class Person {  
    public $name;  
    public $age;  
  
    public function __construct($name, $age) {  
        $this->name = $name;  
        $this->age = $age;  
    }  
  
    public function sayHello() {  
        echo "Hello, my name is $this->name.";  
    }  
}
```

11. Include VS Include_Once

❖ “include” is used to include the file every time it’s called.

➤ Example:

```
<?php  
include "header.php";  
?>  
<h1>Welcome to my website</h1>  
<p>This is some content.</p>  
<?php  
include "footer.php";  
?>
```

- ❖ “include_once” is used to include the file only once, even if it is called multiple times.

➤ Example:

```
<?php
include_once "header.php";
?>
<h1>Welcome to my website</h1>
<p>This is some content.</p>
<?php
include_once "footer.php";
?>
```

12. របៀបរក្សាទិន្នន័យនៅលើ Web ជា Cookies, Sessions, File I/O, and Databases៖

- ❖ Cookies ជា small piece of data មួយដែលត្រូវបាន Stored ទុកនៅលើ computer របស់ user ដោយ web server។ ជាធម្មតា Cookies ត្រូវបានគេប្រើប្រាស់សម្រាប់ store នូវ information អំពី user រួមមាន login credentials or user preferences។

➤ Example:

```
// Set a cookie
setcookie('name', 'value');

// Get a cookie
$value = $_COOKIE['name'];
```

- ❖ Sessions មានភាពស្រដៀងគ្នាដូចទៅនឹង Cookies ដែរ ប៉ុន្តែទិន្នន័យត្រូវបាន Stored នៅលើ Server-side ជាជាងលើ Client-side។ ទិន្នន័យដែល stored នៅលើ Session នឹងបាត់បង់នៅពេលដែល browser បញ្ចប់សកម្មភាព ឬឈប់ប្រើ។

➤ Example:

```
// Start a session
session_start();

// Set a session variable
$_SESSION['name'] = 'value';

// Get a session variable
$value = $_SESSION['name'];

// Destroy a session
session_destroy();
```


- ❖ **File I/O** សំដៅលើការ Read and Write ទិន្នន័យចូលទៅក្នុង File ដែលនៅលើ Server ឬ Local Machine។ File I/O អាចរក្សាទុកទិន្នន័យបានច្រើនជាង Cookies and Sessions។

➤ **Example**

```
// Open a file for reading
$file = fopen('file.txt', 'r');

// Read the contents of the file
$contents = fread($file, filesize('file.txt'));

// Close the file
fclose($file);

// Write to a file
$file = fopen('file.txt', 'w');

// Write some data to the file
fwrite($file, 'This is some data.');
```

- ❖ **Database** ជាមធ្យោបាយមួយសម្រាប់រក្សាទុកទិន្នន័យជាអចិន្ត្រៃយ៍ក្នុង Structured format ដែលមានភាពងាយស្រួលក្នុងការ Query and Manipulate។ Database ត្រូវបានគេប្រើជាញឹកញាប់ក្នុងការរក្សាទុកទិន្នន័យធំៗ និងច្រើន។ ការរក្សាទុកទិន្នន័យអាចផ្ទុកនៅលើ Database systems មួយចំនួនរួមមាន៖ MySQL, PostgreSQL, SQL Server, Oracle, and SQLite។

➤ **Example:**

```
// Connect to a database
$db = new PDO('mysql:host=localhost;dbname=mydb', 'username', 'password');

// Insert a row into a table
$sql = 'INSERT INTO users (name, email) VALUES (?, ?)';

// Prepare the statement
$stmt = $db->prepare($sql);

// Bind the values
$stmt->bindValue(1, 'John Doe');
$stmt->bindValue(2, 'johndoe@example.com');

// Execute the statement
$stmt->execute();
```

13. File Upload ជាដំណើរការនៃការ Uploading file ពី User Computer ទៅកាន់ Web Server។ File Upload អនុញ្ញាតអោយអ្នកប្រើប្រាស់អាចធ្វើការ Upload files ជាច្រើនរូបមាន៖ Images, Documents, and other files ទៅកាន់ Web Application។

❖ **Example of File Upload:**

- **In index.php file:**

```
<form action="upload.php" method="post" enctype="multipart/form-data">
    <input type="file" name="file" />
    <input type="submit" name="submit" value="Upload" />
</form>
```

- **In upload.php file:**

```
<?php
    if(isset($_POST['submit'])) {
        $file = $_FILES['file'];

        // Check if file was uploaded successfully
        if($file['error'] === UPLOAD_ERR_OK) {
            $filename = basename($file['name']);
            $destination = "uploads/" . $filename;

            // Move uploaded file to destination
            if(move_uploaded_file($file['tmp_name'], $destination)) {
                echo "File uploaded successfully!";
            } else {
                echo "Error uploading file.";
            }
        } else {
            echo "Error uploading file.";
        }
    }
?>
```

14. សរសេរ PHP Code ធ្វើការ Read File ដើម្បីបង្ហាញរូបភាពទាំងអស់៖

```
<?php
    // Get the directory path where the images are stored
    $dir = 'uploads/';

    // Get all the files in the directory
    $files = scandir($dir);

    // Loop through the files
    foreach ($files as $file) {

        // Check if the file is an image
        $ext = pathinfo($file, PATHINFO_EXTENSION);
        if (in_array($ext, ['jpg', 'jpeg', 'png', 'gif'])) {

            // Display the image
            echo '';

        }

    }
?>
```

15. សរសេរ Function ក្នុង PHP ដើម្បី upload file PowerPoint ដែលមានទំហំតូចជាង ឬស្មើ 10MB៖

❖ In index.php file

```
<form action="upload.php" method="post" enctype="multipart/form-data">
    <input type="file" name="pptx_file"/>
    <input type="submit" value="Upload" />
</form>
```

❖ In upload_pptx.php file

```
<?php
function uploadPowerPointFile() {
    $file_pptx = $_FILES['pptx_file'];

    // Check if the file was uploaded
    if (!isset($file_pptx)) {
        return false;
    }

    // Check if the file is a PowerPoint file
    $ext = pathinfo($file_pptx['name'], PATHINFO_EXTENSION);
    if ($ext !== 'ppt' && $ext !== 'pptx') {
        return false;
    }

    // Check if the file size is less than or equal to 10MB
    if ($file_pptx['size'] > 10000000) {
        return false;
    }

    // Move the file to the uploads directory
    $destination = 'uploads/' . $file_pptx['name'];
    if (!move_uploaded_file($file_pptx['tmp_name'], $destination)) {
        return false;
    }

    // Success!
    return true;
}
?>
```