

# Web Development Review Question + Answer

## 1. តើអ្វីជាភាពខុសគ្នារវាង back-end និង front-end? ចូរពន្យល់

- ⇒ Back-end: ជាផ្នែកសំរាប់ផ្តល់នូវទិន្នន័យ, Process data ជាមួយ database, និងគ្រប់គ្រងរាល់ business logic ទាំងអស់សម្រាប់ system ទាំងមូល។
- ⇒ Front-end: ជាផ្នែកសំរាប់បង្ហាញនូវព័ត៌មានទៅកាន់អ្នកមើល ដែលមានទាក់ទងទៅនឹងការ design layout ដូចជា រូបរាងនៃ website, រូបភាព, ពណ៌ ជាដើម។

## 2. តើអ្វីជាភាពខុសគ្នារវាង Static website និង Dynamic website?

Static website	Dynamic website
មាតិកានៃគេហទំព័រមិនអាចផ្លាស់ប្តូរបាននៅពេលកំពុងដំណើរការ	មាតិកានៃគេហទំព័រអាចផ្លាស់ប្តូរបានយ៉ាងឆាប់រហ័ស
គ្មានអន្តរកម្មជាមួយមូលដ្ឋានទិន្នន័យ (Database)	មានអន្តរកម្មជាមួយមូលដ្ឋានទិន្នន័យ (Database)
មិនមានប្រព័ន្ធគ្រប់គ្រងមាតិការ (CMS)	ជាប្រព័ន្ធគ្រប់គ្រងមាតិការ (CMS)
ភាសាសម្រាប់បង្កើតគេហទំព័រ Static មានដូចជា៖ HTML, CSS, និង JavaScript។	ភាសាសម្រាប់បង្កើតគេហទំព័រ Dynamic មានដូចជា៖ Python, PHP, Node.js, ASP.NET និងភាសាផ្សេងៗទៀត។
មាតិការដែលបានបង្កើតហើយគឺដូចគ្នារាល់ពេលដែល page ត្រូវបាន reload	មាតិការត្រូវបានបង្កើតយ៉ាងឆាប់រហ័ស និងផ្លាស់ប្តូរជាទៀងទាត់

## 3. ពន្យល់អំពីដំណើរការនៃ web-client និង web-server នៅពេលយើងបើកគេហទំព័រ

- ⇒ ពន្យល់អំពីដំណើរការនៃ web-client (browser) និង web-server នៅពេលយើងបើកគេហទំព័រ៖
- Browser បញ្ជូនសំណើ HTTP ទៅ web server។ សំណើនេះរួមបញ្ចូល URL នៃគេហទំព័រដែលអ្នកចង់ចូលប្រើ។
  - Web Server ទទួលបានសំណើ HTTP ហើយរកមើល resource ដែលបានស្នើសុំ (ឧ. ឯកសារ HTML រូបភាព ឬមាតិកាផ្សេងៗទៀត)។
  - ប្រសិនបើ resource ដែលបានស្នើសុំគឺ Static content (ឧ. ឯកសារ HTML ឬរូបភាព) web server ទាញយកពីប្រព័ន្ធឯកសាររបស់វា ហើយផ្ញើត្រឡប់ទៅ browser តាមអ៊ីនធឺណិតក្នុងការឆ្លើយតប HTTP។
  - ប្រសិនបើ resource ដែលបានស្នើសុំទាមទារ dynamic processing (ឧ. ទាញយកទិន្នន័យពីមូលដ្ឋានទិន្នន័យ ឬអនុវត្ត server-side logic) web server បញ្ជូនសំណើទៅ application server ឬប្រតិបត្តិ server-side scripts។
  - Web browser ទទួលបានការឆ្លើយតប HTTP ពី web server ដែលមានមាតិកាដែលបានស្នើសុំ (ទាំង Static ឬ Dynamic) ។

- Web browser ញែក HTML អនុវត្ត CSS Style និងប្រតិបត្តិ JavaScript ដើម្បីបង្ហាញគេហទំព័រ និងបង្ហាញរាងដល់អ្នកប្រើប្រាស់។

សង្ខេប៖ web-client (browser) បញ្ជូនសំណើ HTTP ទៅ web-server ដែលបម្រើ Static content ដោយផ្ទាល់ ឬបញ្ជូនសំណើទៅ application-server សម្រាប់ដំណើរការ Dynamic។ បន្ទាប់មកមាតិកាដែលបានបង្កើតត្រូវបានត្រលប់ទៅ browser តាមអ៊ីនធឺណិតក្នុងការឆ្លើយតប HTTP ដែលបង្ហាញ និងបង្ហាញគេហទំព័រដល់អ្នកប្រើប្រាស់។

#### 4. តើ PHP ជាអ្វី? តើការប្រើប្រាស់ PHP មានគុណសម្បត្តិ និងគុណវិបត្តិអ្វីខ្លះ?

⇒ PHP (Hypertext Preprocessor): គឺជាភាសាស្រ្តីប, open-source ដែលត្រូវបានប្រើប្រាស់យ៉ាងទូលំទូលាយដែលបានប្រតិបត្តិនៅលើ server ហើយហើយជាឧបករណ៍ដ៏មានអានុភាពសម្រាប់បង្កើត Dynamic website និងអន្តរកម្មយ៉ាងឆាប់រហ័ស។ ហើយមិនគិតថ្លៃអ្វីទេ វាអាចទាញយក និងប្រើប្រាស់ដោយឥតគិតថ្លៃ។

##### • គុណសម្បត្តិ

- PHP ដំណើរការលឿនជាងស្រ្តីបជាភាសាផ្សេងទៀតដូចជា ASP និង JSP ដែលបណ្តាលឱ្យមានដំណើរការប្រសើរជាងមុន និងលឿនដំណើរការលឿនជាងមុន។
- PHP គឺជា open-source ដែលអនុញ្ញាតឱ្យ developer អាចកែប្រែបានតាមចិត្ត និងការអាប់ដេតរបស់វាដោយមិនចាំបាច់ចំណាយបន្ថែម។
- PHP ផ្តល់នូវ libraries និង framework ជាច្រើនដែលបង្កើនល្បឿនសរសេរកូដ។
- PHP អនុញ្ញាតឱ្យ developer ធ្វើការផ្លាស់ប្តូរ ឬបញ្ចូលមុខងារថ្មីៗដោយមិនចាំបាច់សរសេរកូដឡើងវិញ ដោយអនុញ្ញាតឱ្យមានការតភ្ជាប់ជាមួយភាសា និងបច្ចេកវិទ្យាផ្សេងទៀត។
- PHP support ជាមួយមូលដ្ឋានទិន្នន័យពេញនិយមជាច្រើន រួមទាំង MySQL, PostgreSQL, SQLite និងផ្សេងទៀត។

##### • គុណវិបត្តិ

- PHP ជា open-source ដែលធ្វើឱ្យ PHP source code អាចមើលឃើញ ដែលធ្វើឱ្យគេហទំព័រងាយរងការវាយប្រហារដូចជា XSS និង SQL injection ។
- PHP ប្រហែលជាមិនស័ក្តិសមសម្រាប់គម្រោងដែលទាមទារបច្ចេកវិទ្យាកម្រិតខ្ពស់ដូចជា machine-learning ឬ big data ដោយសារការ customization មានកម្រិត។
- PHP មិនមាន debugging tools ច្រើនដែលអាចកំណត់អត្តសញ្ញាណកំហុសក្នុងពេលវេលាជាក់ស្តែង។
- PHP មិនអាចគាំទ្រមុខងារច្រើនដែលដំណើរការដោយរលូនក្នុងពេលតែមួយនោះទេ ហើយការប្រើមុខងារច្រើនទៀតអាចនាំឱ្យមានដំណើរការយឺត។
- PHP អ្នកទទួលបានឱកាសតិចក្នុងការកែប្រែគេហទំព័រ។

#### 5. តើអ្វីជាភាពខុសគ្នារវាង web dynamic client និង web dynamic server?

⇒ ភាពខុសគ្នារវាង web dynamic client និង web dynamic server មានដូចខាងក្រោម៖

Web Dynamic Client	Web Dynamic Server
Dynamic content ត្រូវបានបង្កើតនៅខាង client-side ដោយប្រើ JavaScript ដែលដំណើរការលើ browser តាមអ៊ីនធឺណិតរបស់អ្នកប្រើ។	Dynamic content ត្រូវបានបង្កើតនៅផ្នែកខាង server ដោយប្រើភាសាផ្នែកខាង server ដូចជា PHP, Python ឬ Ruby ។
កូដ JavaScript ធ្វើអន្តរកម្មជាមួយ Document Object Model (DOM) នៃគេហទំព័រ ដើម្បីផ្លាស់ប្តូរ content, layout របស់វា។	Server ធ្វើអន្តរកម្មជាមួយមូលដ្ឋានទិន្នន័យ ដើម្បីទាញយក និងរៀបចំទិន្នន័យ ដែលបន្ទាប់មកត្រូវបានបញ្ជូនទៅ client-side ជា dynamic content។
Server ធ្វើតែឯកសារ HTML, CSS និង JavaScript ដំបូងទៅ client-side ប៉ុណ្ណោះ។	Server ត្រូវគ្រប់គ្រងសមាសធាតុជាច្រើន រួមទាំង application server, database និង web server ដែលអាចនាំឱ្យស្មុគស្មាញជាង។
ដោយសារ dynamic content ត្រូវបានបង្កើតក្នុងមូលដ្ឋាននៅលើ client-side វាអាចឆ្លើយតបទៅនឹងអន្តរកម្មរបស់អ្នកប្រើប្រាស់កាន់តែលឿន។	Server ត្រូវការដំណើរការសំណើ dynamic content ដែលអាចបង្កើនការផ្ទុក server និងអាចប៉ះពាល់ដល់ដំណើរការ។
Server មិនចាំបាច់ដំណើរការសំណើ dynamic content ដែលអាចកាត់បន្ថយការផ្ទុក server និងកែលម្អដំណើរការប្រព័ន្ធទាំងមូល។	Dynamic server អាចគ្រប់គ្រងបរិមាណអ្នកប្រើប្រាស់ និងសំណើកាន់តែច្រើន ដោយសារសមត្ថភាពរបស់ពួកគេក្នុងការចែកចាយដំណើរការឆ្លងកាត់ server ច្រើន។

### 6. តើ Model View Control (MVC) គឺជាអ្វី? ចូរពន្យល់។

⇒ Model View Control (MVC) គឺជា software pattern ដែរចែកដាច់កូដទៅតាមមុខងាររៀងៗខ្លួន, ធ្វើអោយកូដមានរបៀបរាបរយនិងងាយស្រួលក្នុងការកែប្រែ និង បន្ថែមមុខងារថ្មី ។

- Model: ជាផ្នែកមួយដែរតំណាងអោយ Data ដែរវាមិនទាក់ទងនិង user ឡើយ, អ្វីដែរយើងត្រូវធ្វើក្នុង layer គឺការគ្រប់គ្រង data, logic និង rule។
- View: ជាផ្នែកដែរទាក់ទងនឹងការបង្ហាញទៅកាន់អ្នកប្រើប្រាស់ (UI: User Interface). ផ្នែកនេះគឺទទួល data ពី controller ហើយបង្ហាញវាទៅលើ view។
- Control: ជាផ្នែកដែរភ្ជាប់ទំនាក់ទំនងរវាង View និង Model, Controller ជាអ្នកសម្រេចថា គួរប្រើ Model មួយណា ហើយផ្តល់ព័ត៌មានទៅកាន់ View មួយណាវិញ ។

### 7. ពន្យល់អំពីអត្ថប្រយោជន៍នៃការប្រើប្រាស់ MVC ក្នុងការអភិវឌ្ឍន៍គេហទំព័រ។

⇒ អត្ថប្រយោជន៍នៃការប្រើប្រាស់ MVC ក្នុងការអភិវឌ្ឍន៍គេហទំព័រមានដូចជា៖

- MVC បំបែកជាសមាសធាតុបីផ្សេងគ្នា (Model, View, Control) ដែលធ្វើឱ្យរកាន់តែងាយស្រួលយល់ រក្សា និង កែប្រែមូលដ្ឋានកូដ។
- MVC ជួយសម្រួលដល់ការអភិវឌ្ឍន៍គេហទំព័រដែលងាយស្រួលប្រើ SEO បង្កើនល្បឿនគេហទំព័រ និងបទពិសោធន៍អ្នកប្រើប្រាស់ ដែលមានសារៈសំខាន់សម្រាប់រង្វាស់ SEO។
- MVC ផ្តល់នូវចំណាត់ថ្នាក់លាស់សម្រាប់រៀបចំកូដ ដែលធ្វើឱ្យរកាន់តែងាយស្រួលក្នុងការរុករក និងរក្សាមូលដ្ឋានកូដ។
- MVC អនុញ្ញាតឱ្យធ្វើការ testing និងការធ្វើតេស្តដោយស្វ័យប្រវត្តិនៃសមាសធាតុនីមួយៗ ធានានូវភាពជឿជាក់ និងភាពត្រឹមត្រូវនៃកម្មវិធី។
- ការផ្លាស់ប្តូរនៅក្នុង source code ឬ UI អាចគ្រប់គ្រងបានយ៉ាងងាយស្រួលដោយមិនប៉ះពាល់ដល់កំរិតទាំងមូលធ្វើឱ្យរកាន់តែងាយស្រួលក្នុងការបន្ថែម និងគ្រប់គ្រង។

## 8. តើអថេរ Scope មានប៉ុន្មាននៅក្នុង PHP? List all of them with examples

⇒ អថេរ Scope មាន 3 នៅក្នុង PHP ដូចជា៖

- Local Scope

*ឧទាហរណ៍៖*

```
function myFunction() {
    $x = 5; // $x is local scope
    echo $x;
}
```

- Global Scope

*ឧទាហរណ៍៖*

```
$x = 5; // $x is global scope
function myFunction () {
    global $x;
    echo $x;
}
```

- Static Scope

*ឧទាហរណ៍៖*

```
function myFunction () {
    static $x = 0;
    $x++;
    echo $x;
}
```

9. ពន្យល់ពី “global” និង “static” keyword ក្នុង PHP ព្រមទាំងលើកឧទាហរណ៍បញ្ជាក់។

⇒ Global keyword

Global keyword នៅក្នុង PHP ត្រូវបានប្រើដើម្បីចូលប្រើអថេរ global ពីក្នុង function មួយ។ វាអនុញ្ញាតឱ្យ function ចូលប្រើអថេរដែលបានប្រកាសនៅខាងក្រៅរបស់វា។ នៅពេលដែលអថេរត្រូវបានប្រកាសនៅខាងក្រៅ function វាមាន global scope ហើយអាចចូលប្រើបានពីគ្រប់កន្លែងក្នុងកូដ។ ដើម្បីចូលប្រើអថេរ global នៅក្នុង function អ្នកត្រូវប្រើ global keyword តាមពីក្រោយដោយឈ្មោះអថេរនៅខាងក្នុង function។

*ឧទាហរណ៍៖*

```
$x = 5;

function myTest() {
    global $x;
    echo "x is equal to ". $x;
    $GLOBALS['x'] ++;
}
```

⇒ Static keyword

Static keyword នៅក្នុង PHP ត្រូវបានប្រើដើម្បីប្រកាស properties និង method នៃ class ជា static។ Static properties ហើយ method អាចចូលប្រើបានដោយមិនចាំបាច់បង្កើត instance នៃ class ទេ។

*ឧទាហរណ៍៖*

```
function add10() {
    static $number = 0;
    $number ++;
    return $number;
}

echo add10;
echo add10;
echo add10;
```

10. តើអ្វីជាភាពខុសគ្នារវាង "echo " និង "print "? ផ្តល់ឧទាហរណ៍អំពីរបៀបប្រើពួកវា

⇒ ភាពខុសគ្នារវាង “echo” និង “print”

echo	print
“echo” មិនមានតម្លៃ return	“print” always returns 1

“echo” អាចទទួលយក arguments ជាច្រើនដែលបំបែកដោយសញ្ញាក្បឿង (,)។	“print” អាចទទួលយកបានតែ arguments តែមួយប៉ុណ្ណោះ
“echo” ត្រូវបានប្រើដើម្បីបញ្ចេញ string មួយ ឬច្រើន	“print” ត្រូវបានប្រើដើម្បីបញ្ចេញតែ single string
“echo” គឺជាការបង្កើតភាសា ហើយមិនតម្រូវឱ្យមានរង់ក្រចក។ វាអាចត្រូវបានប្រើដោយមានឬគ្មានរង់ក្រចក។	“print” គឺជា function ហើយទាមទាររង់ក្រចក។
“echo” លឿនជាង “print”	“echo” គឺជា function មួយ ហើយពាក់ព័ន្ធនឹងការហៅលើមុខងារ។
<b>ឧទាហរណ៍៖</b> echo "Hello"; // Outputs: Hello echo "Hello", " ", "World"; // Outputs: Hello World	<b>ឧទាហរណ៍៖</b> print "Hello"; // Outputs: Hello print "Hello"; // Outputs: Hello (prints the same string again)

## 11. តើអ្វីជាភាពខុសគ្នារវាង \$\_POST និង \$\_GET? ផ្តល់ឧទាហរណ៍

\$_POST	\$_GET
វាត្រូវបានប្រើដើម្បីទាញយកទិន្នន័យដែលបានផ្ញើទៅ server ដោយប្រើវិធីសាស្ត្រ HTTP POST ។	វាត្រូវបានប្រើដើម្បីទាញយកទិន្នន័យដែលបានផ្ញើទៅ server ដោយប្រើវិធីសាស្ត្រ HTTP GET ។
ទិន្នន័យដែលផ្ញើតាមវិធី POST មិនអាចមើលឃើញនៅក្នុង URL ទេ។	ទិន្នន័យដែលបានផ្ញើតាមរយៈវិធីសាស្ត្រ GET អាចមើលឃើញនៅក្នុង URL ។
កាន់តែមានសុវត្ថិភាព ដោយសារទិន្នន័យមិនអាចមើលឃើញនៅក្នុង URL ដែលបង្កើនការសម្ងាត់	មិនសូវមានសុវត្ថិភាព ដោយសារទិន្នន័យអាចមើលឃើញនៅក្នុង URL
ស័ក្តិសមសម្រាប់ព័ត៌មានរសើបដូចជាពាក្យសម្ងាត់ ឬព័ត៌មានលម្អិតផ្ទាល់ខ្លួន។	ស័ក្តិសមសម្រាប់ទិន្នន័យដែលមិនត្រូវការការសុវត្ថិភាព។
<b>ឧទាហរណ៍៖</b> <?php if (\$_SERVER["REQUEST_METHOD"] == "POST") { \$username = \$_POST['username']; \$password = \$_POST['password']; echo "Username: " . \$username . " "; echo "Password: " . \$password . " "; } ?> <form method="post" > <label for="username">Username:</label> <input type="text" name="username" id="username">   <label for="password">Password:</label> <input type="password" name="password" id="password">	<b>ឧទាហរណ៍៖</b> <?php if (\$_SERVER["REQUEST_METHOD"] == "GET") { \$searchQuery = \$_GET['search']; echo "Search query: " . \$searchQuery . " "; } ?> <form method="get" > <label for="search">Search:</label> <input type="text" name="search" id="search">   <button type="submit">Search</button> </form>

<pre> &lt;br&gt; &lt;button type="submit"&gt;Submit&lt;/button&gt; &lt;/form&gt; </pre>	
---	--

12. Write code PHP to allow upload multiple files (only image) to server.

```

<body>
<?php
$uploadDir = 'uploads/';

if(isset($_POST['submit'])){
    $total = count($_FILES['images']['name']);

    for($i=0; $i<$total; $i++){
        $fileName = $_FILES['images']['name'][$i];
        $fileTmp = $_FILES['images']['tmp_name'][$i];
        $fileType = $_FILES['images']['type'][$i];
        $fileSize = $_FILES['images']['size'][$i];

        $targetPath = $uploadDir . basename($fileName);

        if(move_uploaded_file($fileTmp, $targetPath)){
            echo "File uploaded successfully: " . $fileName . "<br>";
        } else {
            echo "Error uploading file: " . $fileName . "<br>";
        }
    }
}
?>

<form action="upload.php" method="post" enctype="multipart/form-data">
    <input type="file" name="images[]" accept="image/*" multiple>
    <input type="submit" value="Upload Images" name="submit">
</form>
</body>

```

13. តើអ្វីជាភាពខុសគ្នារវាង cookies និង sessions? ផ្តល់ឧទាហរណ៍

Cookies	Sessions
ទាំងនេះគឺជាឯកសារអត្ថបទតូចៗដែលផ្ទុកនៅលើឧបករណ៍របស់អ្នកប្រើ (client-side) ដោយគេហទំព័រ។	Session data ត្រូវបានរក្សាទុកនៅលើផ្នែកខាងម៉ាស៊ីនមេ (កុំព្យូទ័ររបស់គេហទំព័រ)។
អាចកំណត់កាលបរិច្ឆេទ/ពេលវេលាផុតកំណត់ ដែលអនុញ្ញាតឱ្យពួកវាបន្តសម្រាប់រយៈពេលជាក់លាក់មួយ សូម្បីតែបន្ទាប់ពី browser ត្រូវបានបិទ (persistent cookies) ឬត្រូវបានលុបនៅពេលដែល browser ត្រូវបានបិទ (session cookies)។	Session ជាធម្មតាបញ្ចប់នៅពេលអ្នកប្រើប្រាស់បិទ browser តាមអ៊ីនធឺណិត ឬចេញពីគណនី។

ពួកគេមានសមត្ថភាពផ្ទុកមានកំណត់ ជាធម្មតាប្រហែល 4KB ។ នេះគឺសមរម្យសម្រាប់ការរក្សាទុកព័ត៌មានតូចៗ។	Session អាចផ្ទុកទិន្នន័យធំជាងបើធៀបនឹង cookies។
Cookies មិនមានសុវត្ថិភាពទេ ព្រោះវាត្រូវបានរក្សាទុកក្នុងអត្ថបទធម្មតានៅ client-side ដែលធ្វើឲ្យពួកគេងាយរងគ្រោះក្នុងការចូលប្រើដោយគ្មានការអនុញ្ញាត	Session មានសុវត្ថិភាពជាងមុន ដោយសារពួកវាត្រូវបានរក្សាទុកនៅផ្នែក server-side ហើយត្រូវបាន encrypt ដែលធ្វើឱ្យវាពិបាកសម្រាប់ការចូលប្រើដោយគ្មានការអនុញ្ញាត
<b>ឧទាហរណ៍៖</b> Cookies អាចនឹងត្រូវបានប្រើប្រាស់ដើម្បីចងចាំអ្វីដែលអ្នកបានបញ្ចូលទៅក្នុងទម្រង់ទិន្នន័យរបស់អ្នក ទោះបីជាអ្នកចូលមើលទំព័រផ្សេងៗនៅលើគេហទំព័រក៏ដោយ។ cookies នេះនឹងមិនមានព័ត៌មានរសើបណាមួយដូចជាព័ត៌មានលម្អិតអំពីការបង់ប្រាក់របស់អ្នកទេ ប៉ុន្តែគ្រាន់តែជាឧបករណ៍កំណត់អត្តសញ្ញាណតែមួយគត់សម្រាប់រទេះរបស់អ្នក។	<b>ឧទាហរណ៍៖</b> បន្ទាប់ពីអ្នកប្រើប្រាស់ login អ្នកបង្កើត session មួយនៅផ្នែក server-side ដើម្បីរក្សាទុកព័ត៌មានអំពីអ្នកប្រើប្រាស់ ដូចជាឈ្មោះអ្នកប្រើប្រាស់ អ៊ីមែល និងអាសយដ្ឋានដឹកជញ្ជូនរបស់ពួកគេ។ នៅពេលដែលអ្នកប្រើប្រាស់ចេញពីប្រព័ន្ធ ឬ session របស់ពួកគេផុតកំណត់ session data ត្រូវបានសម្អាតចេញពី server។

14. តើ cookies និង session អាចរក្សាទុក binary file ដែរឬទេ? ចូរពន្យល់

ទាំង cookies និង sessions មានសមត្ថភាពរក្សាទុកទិន្នន័យគោលពីរ។ session ត្រូវបានចាត់ទុកថាមានសុវត្ថិភាពជាង ដោយសារពួកវារក្សាទុកទិន្នន័យក្នុងទម្រង់គោលពីរ ឬទម្រង់ដែលបាន encrypt ដែលអាច decrypt បានតែនៅ server ប៉ុណ្ណោះ ដែលធានាបាននូវកម្រិតសុវត្ថិភាពខ្ពស់ជាងមុន ហើយម្យ៉ាងវិញទៀត cookies ក៏អាចរក្សាទុកទិន្នន័យគោលពីរបានដែរ ប៉ុន្តែពួកវាមិនសូវមានសុវត្ថិភាពទេ ដោយសារព័ត៌មានត្រូវបានរក្សាទុកនៅ client-side ក្នុងទម្រង់ជា text ដែលអាចអានបានដោយនរណាម្នាក់។

15. Write PHP code to generate the random number with 8 digits (only number)

```
<?php

$randomNum = mt_rand(10000000, 99999999);

echo "Random 8-digits number: ". $randomNum;

?>
```

16. Write PHP code to write "Hello World!" 1000 line in a file named myfile.txt

```
<?php
$file = fopen("myfile.txt", "w");

for ($i = 1; $i <= 1000; $i++) {
    fwrite($file, "Hello World!\n");
}

fclose($file);

echo "File 'myfile.txt' created successfully.";
?>
```



17. Write 3 functions to connect, insert, update, and delete data from mysql database respectively

php

```
$conn = connectToDB('localhost', 'username', 'password', 'database');

$data = array('name' => 'John', 'age' => 30);
insertData($conn, 'users', $data);

$data = array('age' => 31);
$condition = "name = 'John'";
updateData($conn, 'users', $data, $condition);

$condition = "name = 'John'";
deleteData($conn, 'users', $condition);

$conn->close();
```

18. តើអ្វីជាភាពខុសគ្នារវាង \$var1 និង \$\$var1?

- ⇒ \$var1: \$var1 គឺជាអថេរធម្មតាដែលផ្ទុកតម្លៃ។ វាត្រូវបានបង្កើត និងផ្តល់តម្លៃដោយប្រើ syntax: \$var1 = value ហើយ ឈ្មោះអថេរ \$var1 ត្រូវបានជួសជុល ហើយមិនផ្លាស់ប្តូរកំឡុងពេលដំណើរការទេ។
- ⇒ \$\$var1: គឺជាឧទាហរណ៍នៃអថេរនៅក្នុង PHP ។ វាអនុញ្ញាតឱ្យអ្នកប្រើតម្លៃនៃអថេរមួយជាឈ្មោះនៃអថេរផ្សេងទៀត។ ឈ្មោះអថេរដែលរក្សាទុកក្នុង \$var1 ត្រូវបានចាត់ទុកជា string ។

19. តើវិធីណាខ្លះក្នុងការកំណត់ constant ក្នុង PHP?

- ⇒ វិធីដែលកំណត់ Constant ក្នុង PHP មានដូចជា៖ define() function និង const keyword ។

20. តើ isset() function ដំណើរការយ៉ាងដូចម្តេច?

- ⇒ isset() function ក្នុង PHP ត្រូវបានប្រើដើម្បីកំណត់ថាតើអថេរមួយត្រូវបានកំណត់ ហើយមិនមែនជា “null” ទេ។ វា return “true” ប្រសិនបើអថេរមាន ហើយមានតម្លៃផ្សេងពី “null” ហើយ “false”។

21. តើ \$\_SERVER ជាអ្វី? ពន្យល់អំពីអត្ថប្រយោជន៍នៃការប្រើប្រាស់អថេរនេះ។

- ⇒ \$\_SERVER ជាអថេរ superglobal associative ដែលមានព័ត៌មានអំពី server environment, path និងទីតាំង script។ វាផ្តល់ព័ត៌មានមានប្រយោជន៍អំពី script បច្ចុប្បន្នដែលកំពុងដំណើរការ ដូចជា header, path និង URLs។

ស៊ុន ស៊ីណាត

⇒ អត្ថប្រយោជន៍នៃការប្រើប្រាស់អថេរ (\$\_SERVER) នេះ៖

- ការទាញយកព័ត៌មានអំពី server ដូចជាឈ្មោះ server, IP address, port number, software និង protocol។
- ការចូលប្រើព័ត៌មានលម្អិតអំពី current script ដូចជា filename, path និង URI។
- ការទទួលបានព័ត៌មានអំពីសំណើ HTTP បច្ចុប្បន្ន ដូចជាវិធីសាស្ត្រស្នើសុំ (GET, POST ។ល។) header និង query string។
- ការទទួលបានព័ត៌មានលម្អិតអំពីអតិថិជន ដូចជាអាសយដ្ឋាន IP និង user agent។

## 22. តើអ្វីជាភាពខុសគ្នារវាង require, require\_once, include, include\_once?

⇒ ភាពខុសគ្នារវាង require, require\_once, include, include\_once:

require	require_once	include	include_once
ត្រូវបានប្រើដើម្បីបញ្ចូលឯកសារទៅក្នុង PHP script ។ ប្រសិនបើឯកសារដែលបានបញ្ជាក់មិនបានបញ្ចូល កំហុសឆ្គងឆ្ងន់ត្រូវបានបង្កើត ហើយការ script executive ត្រូវបានបញ្ឈប់។	'require_once' ស្រដៀង 'require' ដែរ ប៉ុន្តែប្រើដើម្បីពិនិត្យមើលថាតើឯកសារត្រូវបានរួមបញ្ចូលរួចហើយឬអត់។ ប្រសិនបើឯកសារត្រូវបានរួមបញ្ចូលរួចហើយ វាមិនត្រូវបានបញ្ចូលម្តងទៀតទេ ដូច្នេះការការបញ្ចូលស្មុគស្មាញ។	'include' statement ត្រូវបានប្រើដើម្បីបញ្ចូលឯកសារទៅក្នុងស្ត្រីប PHP ។ ប្រសិនបើឯកសារដែលបានបញ្ជាក់មិនអាចបញ្ចូលបាន ការប្រឆាំងត្រូវបានបង្កើត ប៉ុន្តែការ script executive នៅតែបន្ត។	'include_once' ស្រដៀង 'include' ដែរ ប៉ុន្តែវាពិនិត្យមើលថាតើឯកសារត្រូវបានបញ្ចូលនៅកន្លែងផ្សេងទៀតនៅក្នុងស្ត្រីបឬអត់។ ប្រសិនបើឯកសារត្រូវបានបញ្ចូលរួចហើយ វានឹងមិនត្រូវបានរួមបញ្ចូលម្តងទៀតទេ។

## 23. តើអ្វីជាភាពខុសគ្នារវាង mysqli និង PDO ហើយតើពេលណាដែលអ្នកនឹងប្រើវា?

mysqli	PDO
MySQLi គាំទ្រតែ MySQL ប៉ុណ្ណោះ។	PDO គាំទ្រ 12 ប្រភេទមូលដ្ឋានទិន្នន័យផ្សេងគ្នា
Support នៅ statement ដែលបានរៀបចំទុក ដែលអាចជួយការពារការរាយប្រហារ SQL injection។	ក៏ support នៅ statement ដែលបានរៀបចំផងដែរ ប៉ុន្តែការអនុវត្តរបស់វាមានភាពរឹងមាំជាងបន្តិច និងអាចចាត់តាំងតាមប្រព័ន្ធមូលដ្ឋានទិន្នន័យផ្សេងៗគ្នា។
MySQLi អនុវត្តតាម 'snake_case' convention។	PDO ប្រើឈ្មោះ 'camelCase' method

⇒ ពេលដែលត្រូវប្រើវា៖

- ប្រើ mysqli នៅពេលអ្នកកំពុងធ្វើការជាមួយជាមួយ MySQL ហើយត្រូវការលក្ខណៈពិសេស MySQL ជាក់លាក់ដែលមិនត្រូវបានផ្តល់ដោយ PDO ។
- ប្រើ PDO នៅពេលអ្នកចង់បានដំណោះស្រាយដែលអាចបត់បែនបាន និងចាត់តាំងជាមុន ដែលអាចធ្វើការជាមួយមូលដ្ឋានទិន្នន័យច្រើន។ វាអនុញ្ញាតឱ្យអ្នកប្តូរមូលដ្ឋានទិន្នន័យបានយ៉ាងងាយស្រួលដោយមិនចាំបាច់សរសេរឡើងវិញនូវផ្នែកសំខាន់នៃកូដរបស់អ្នក។

## 24. តើ object-oriented programming (OOP) ជាអ្វី? ហើយតើអ្នកបង្កើត class និង object ក្នុង PHP ដើម្បីរៀបចំ កូដ និងកែលម្អការប្រើប្រាស់កូដឡើងវិញដោយរបៀបណា?

⇒ Object-Oriented Programming (OOP): ជាវិធីសាស្ត្រនៃការបង្កើតកម្មវិធី ដោយរួមបញ្ចូលនៅ Concept របស់ Class, Object និង Concept ផ្សេងៗទៀតដែលពាក់ព័ន្ធនឹងទាំងពីរនេះ (Class & Object) ដូចជា Encapsulation, Inheritance, Polymorphism ជាដើម។

⇒ ដើម្បីបង្កើត class និង object ក្នុង PHP ដើម្បីរៀបចំកូដ និងកែលម្អការប្រើប្រាស់កូដឡើងវិញតាមរបៀបដូចខាងក្រោម៖

- Create Classes:
  - កំណត់ class ដោយប្រើ 'class' keyword តាមដោយឈ្មោះ class។
  - បន្ថែមនៅ properties (variable) ទៅ class ដើម្បីកំណត់លក្ខណៈរបស់វា។
  - ប្រើឧបករណ៍កែប្រែការចូលប្រើដូចជា 'public', 'protected' ឬ 'private' ដើម្បីគ្រប់គ្រងលទ្ធភាពមើលឃើញនៃ properties។
  - កំណត់ method (function) នៅក្នុង class ដើម្បីបញ្ចូល behavior។
- Creating Objects:
  - បន្ទាប់ពីកំណត់ class បង្កើត object (instance) នៃ class នោះដោយប្រើ 'new' keyword។
  - ចូលប្រើ properties និង method នៃ class ដោយប្រើ object instance និងសញ្ញាញាញ់ '->' ។

```
php
<?php
// Define a class named User
class User {
    public $name = 'John Doe';
    public $age = 30;

    public function greet() {
        echo "Hello, my name is " . $this->name;
    }
}

// Create objects of the User class
$user1 = new User();
$user2 = new User();

// Access object properties and methods
echo $user1->name; // Output: John Doe
$user2->greet();   // Output: Hello, my name is John Doe
?>
```

## 25. តើយើងអាចបង្កើតមូលដ្ឋានទិន្នន័យដែលមានឈ្មោះថា "mydatabase" ដោយប្រើ PHP និង MySQL យ៉ាងដូចម្តេច?

⇒ យើងអាចបង្កើតមូលដ្ឋានទិន្នន័យដែលមានថា "mydatabase" ដោយប្រើ PHP និង MySQL តាមជំហានដូចខាងក្រោម៖

- ជាដំបូង បង្កើតការតភ្ជាប់ទៅ MySQL Server ដោយប្រើ PHP ។ អ្នកត្រូវផ្តល់ Server name, Username, Password និងជាជម្រើសនូវ port number ប្រសិនបើចាំបាច់។
- ប្រើ PHP ដើម្បីប្រតិបត្តិ SQL query ដើម្បីបង្កើតមូលដ្ឋានទិន្នន័យ។ SQL query ដើម្បីបង្កើតមូលដ្ឋានទិន្នន័យគឺ 'CREATE DATABASE mydatabase'។
- ផ្ទៀងផ្ទាត់ថា ការបង្កើតមូលដ្ឋានទិន្នន័យបានជោគជ័យ និងដោះស្រាយកំហុសដែលអាចកើតឡើងក្នុងអំឡុងពេលដំណើរការ។

```
php
<?php
$servername = "localhost";
$username = "username";
$password = "password";

// Create connection
$conn = new mysqli($servername, $username, $password);

// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

// Create database
$sql = "CREATE DATABASE mydatabase";

if ($conn->query($sql) === TRUE) {
    echo "Database created successfully";
} else {
    echo "Error creating database: " . $conn->error;
}

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

26. Write a PHP code to send an email.

```
php
<?php
$to = "recipient@example.com";
$subject = "Test Email";
$message = "This is a test email sent using PHP.";
$headers = "From: sender@example.com\r\n";
$headers .= "Reply-To: sender@example.com\r\n";
$headers .= "CC: cc@example.com\r\n"; // Optional
$headers .= "BCC: bcc@example.com\r\n"; // Optional

// Send email
if (mail($to, $subject, $message, $headers)) {
    echo "Email sent successfully.";
} else {
    echo "Email sending failed.";
}
?>
```

27. តើអ្នកអាចប្រតិបត្តិ SQL query នៅក្នុង PHP ដោយរបៀបណា ហើយតើអ្វីជាការអនុវត្តល្អបំផុតសម្រាប់ការដោះស្រាយការបញ្ចូលរបស់អ្នកប្រើដើម្បីការពារការរាយប្រហារដោយ SQL injection?

- ⇒ អ្នកអាចប្រតិបត្តិ SQL query នៅក្នុង PHP តាម MySQLi ហើយនឹង PDO ហើយការអនុវត្តល្អបំផុតសម្រាប់ការដោះស្រាយការបញ្ចូលរបស់អ្នកប្រើដើម្បីការពារការរាយប្រហារដោយ SQL injection:
- ប្រើ PHP function ដូចជា mysqli\_query() ឬ PDO ដើម្បីប្រតិបត្តិ SQL query ប្រឆាំងនឹងមូលដ្ឋានទិន្នន័យ
  - ប្រើប្រាស់ statement ដែលបានរៀបចំ និងសំណួរដែលបានកំណត់ ដើម្បីការពារប្រឆាំងនឹងការរាយប្រហារដោយ SQL injection
  - Statement ដែលបានរៀបចំជួយបំបែក SQL Logic ពីការបញ្ចូលរបស់អ្នកប្រើប្រាស់ បង្កើនសុវត្ថិភាព និងដំណើរការ
  - ធ្វើបច្ចុប្បន្នភាពកំណែ PHP និង MySQLi ជាទៀងទាត់ ដើម្បីជួសជុលភាពងាយរងគ្រោះដែលគេស្គាល់ និងបង្កើនសុវត្ថិភាព
  - ប្រើ framework ដូចជា Laravel ដែលគ្រប់គ្រងអន្តរកម្មមូលដ្ឋានទិន្នន័យកម្រិតទាប និងវិធានការសុវត្ថិភាពដោយស្វ័យប្រវត្តិ

28. Write a code let user upload only image to server and the image must less than 3MB.

```
<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $targetDir = "uploads/";
    $targetFile = $targetDir . basename($_FILES["fileToUpload"]["name"]);
    $uploadOk = 1;
    $imageFileType = strtolower(pathinfo($targetFile, PATHINFO_EXTENSION));

    $check = getimagesize($_FILES["fileToUpload"]["tmp_name"]);
    if ($check !== false) {
        if ($_FILES["fileToUpload"]["size"] > 3 * 1024 * 1024) { // 3MB limit
            echo "Sorry, your file is too large. Please upload an image less than 3MB.";
            $uploadOk = 0;
        }
        if ($imageFileType != "jpg" && $imageFileType != "png" && $imageFileType != "jpeg" && $imageFileType != "gif") {
            echo "Sorry, only JPG, JPEG, PNG & GIF files are allowed.";
            $uploadOk = 0;
        }
        if ($uploadOk == 0) {
            echo "Your file was not uploaded.";
        } else {
            if (move_uploaded_file($_FILES["fileToUpload"]["tmp_name"], $targetFile)) {
                echo "The file ". basename( $_FILES["fileToUpload"]["name"]). " has been uploaded.";
            } else {
                echo "Sorry, there was an error uploading your file.";
            }
        }
    } else {
        echo "File is not an image.";
        $uploadOk = 0;
    }
}
```

```
?>
<!DOCTYPE html>
<html>
<body>
    <form action="" method="post" enctype="multipart/form-data">
        Select image to upload:
        <input type="file" name="fileToUpload" id="fileToUpload">
        <input type="submit" value="Upload Image" name="submit">
    </form>
</body>
</html>
```

29. Write a php to connect to database and insert a data to database below:

- 1) Database name: mydatabase [id (auto number), Name (varchar (100)), sex (boolean), email (varchar(200)), phone (varchar(100)), password(varchar(100))
- 2) Password: 123
- 3) Host: localhost

Name :

Password :

Gender : ☐ Male ☐ Female

Email :

Phone no :

(Note that password is autogenerated (8 digit (only number))

```
<?php
$host = 'localhost';
$username = 'root';
$password = '';
$database = 'mydatabase';

$conn = mysqli_connect($host, $username, $password, $database);

if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}

function generatePassword()
{
    return mt_rand(10000000, 99999999);
}

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $name = $_POST['name'];
    $gender = $_POST['gender'];
    $email = $_POST['email'];
    $phone = $_POST['phone'];

    $password = generatePassword();
    $sql = "INSERT INTO users (Name, sex, email, phone, password) VALUES ('$name', '$gender', '$email', '$phone', '$password')";
    if (mysqli_query($conn, $sql)) {
        echo "New record created successfully";
    } else {
        echo "Error: " . $sql . "<br>" . mysqli_error($conn);
    }
}

mysqli_close($conn);
?>
```

```

<!DOCTYPE html>
<html>

<head>
<title>Insert Data</title>
</head>

<body>
<form method="post" action="<?php echo $_SERVER['PHP_SELF']; ?>">
    Name: <input type="text" name="name" required><br><br>
    Password: <input type="password" name="password"><br><br>
    Gender:
    <input type="radio" name="gender" value="Male" checked>Male
    <input type="radio" name="gender" value="Female">Female
    <br><br>
    Email: <input type="email" name="email" required><br><br>
    Phone no: <input type="combobox" name="phone_prefix" value="(997)" disabled>
    <input type="text" name="phone" required><br><br>
    <input type="submit" value="Submit">
</form>
</body>

</html>

```

30. Write to search for a student based on their name and date of birth in a single php file. If the student is found, "Found!" message will be display while if not, "Not Found!" message is shown in the page

```

<?php
// Define the student data
$students = array(
    array(
        "name" => "Sinat Sin",
        "dob" => "1990-05-15"
    ),
    array(
        "name" => "Jane Smith",
        "dob" => "1995-09-20"
    ),
    array(
        "name" => "Michael Johnson",
        "dob" => "1985-12-01"
    )
);

// Check if the form is submitted
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $name = $_POST["name"];
    $dob = $_POST["dob"];

    // Search for the student
    $found = false;
    foreach ($students as $student) {
        if ($student["name"] == $name && $student["dob"] == $dob) {
            echo "Found!";
            $found = true;
            break;
        }
    }

    if (!$found) {
        echo "Not Found!";
    }
}
?>

```



```

<!DOCTYPE html>
<html>

<head>
<title>Student Search</title>
</head>

<body>
<h1>Student Search</h1>
<form method="post" action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]); ?>">
    Name: <input type="text" name="name"><br>
    Date of Birth (YYYY-MM-DD): <input type="text" name="dob"><br>
    <input type="submit" name="submit" value="Search">
</form>
</body>

</html>

```

