កម្រងសំណួរសម្រាប់ប្រឡងបញ្ចប់

1. តើអ្វីទៅជា PHP? ឬ តើ Dynamic Server ជាអ្វី?

2. តើ PHP មានអត្ថប្រយោជន៍សម្រាប់ធ្វើអ្វី?

3. ចូរនិយាយការបង្កើត Structure របស់ PHP​។

4. តើការបង្កើតអថេរ (Variable) នៅក្នុង​ PHP មានប៉ុន្មានប្រភេទអ្វីខ្លះ? ចូរពន្យល់។

5.​​​ តើការបង្កើត Function​ នៅក្នុង PHP មានអ្វីខ្លះ? ចូរពន្យល់។

6. ចូរសរសេរពីការប្រើប្រាស់ for-each (control structure) នៅក្នុង PHP​​​ ។

7. Input និង​ Output នៅក្នុង PHP

* ចូរបង្កើត Form​ មួយដែល Input data ទៅកាន់ Server (Input)
* ប្រៀបធៀប GET and POST

8. តើអ្វីទៅជា OOP? មានប៉ុន្មានប្រភេទ? អ្វីខ្លះ? ចូររៀបរាប់។

9. និយាយពីអត្ថប្រយោជន៍របស់ OOP ។

10. ចូរនិយាយ​ពីការ Call File មួយទៅ file មួយទៀត នៅក្នុង PHP និងលើកឧទាហរណ៍។

11. ចូរពន្យល់អំពីអត្ថប្រយោជន៍នៃការប្រើប្រាស់ Pattern MVC នៅក្នុងការអភិវឌ្ឍន៍គេហទំព័រ ។

12. តើអ្វីទៅជា Cookie, Session, File IO and Database? តើគេប្រើប្រាស់ដើម្បីធ្វើអ្វី? ចូរលើកឧទាហរណ៍។

13. ចូរនិយាយពីការភ្ជាប់ Database​ នៅក្នុង PHP និង លើកឧទាហរណ៍។

14. ហេតុអ្វីបានជា Web Application ភាគច្រើនប្រើប្រាស់ Pattern MVC របស់ Laravel?

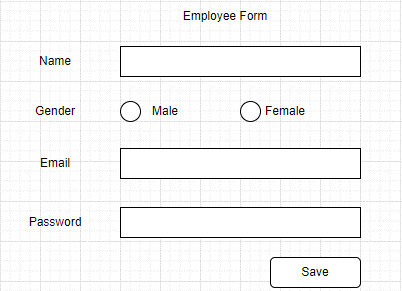
15. ប្រៀបធៀប Static Website and Dynamic Website?

16. ប្រៀបធៀប Web Application ដែលប្រើប្រាស់​​ Laravel and PHP​។

17. ចូរនិយាយពី ដំណើរការ (Process Flow) នៃ web-client (browser) និង web-server នៅពេល Client បើក Web page នៅក្នុង Laravel​។

(Route->Controller->Model->Controller->View)

Review WEB Application Development



Notice:

Form can have more like radio or dropdown.

Make sure to find out that

លំហាត់៖

1. គេមាន៖

* Database Name: mydb
* Username: admin
* Password: 222333
* Table: tbuser(id, name, email, password)

|  |  |  |
| --- | --- | --- |
| Name | Email | Password |
| sok | sok@gmail.com | 123456 |
| dara | dara@gmail.com | 234567 |
| tong | tong@gmail.com | 345678 |
| kong | kong@gmail.com | 456789 |

ក. សរសេរ Code PHP ដើម្បីភា្ជប់ (connect) ទៅកាន់ Web page

ខ. បញ្ចូល (Insert) ទិន្នន័យ រួច Save និង display on table list

- Before save to database Check data មានឬអត់?​ បើអត់មាន Save and display on list

គ.​ Select data ណាមួយ ធ្វើការកែប្រែ និង បញ្ជូនទៅ Database (Update)

ចម្លើយ៖

1. តើអ្វីទៅជា PHP? ឬ តើ Dynamic Server ជាអ្វី?

* PHP មកពីពាក្យ Hypertext Preprocessor គឺជាប្រភេទ scripting language សម្រាប់សរសេរកូដផ្នែកខាងក្រោយនៃ Website ឬហៅថា server-side ដែលត្រូវបានប្រើដើម្បីបង្កើត​ Website ដែលមានលក្ខណះជា Dynamic។
* Dynamic Server: គឺជាម៉ាស៊ីនមួយដែលបង្កើតមាតិកានៃគេហទំព័រអាចផ្លាស់ប្តូរបានយ៉ាងឆាប់រហ័ស ហើយមានអន្តរកម្មជាមួយមូលដ្ឋានទិន្នន័យ (database) និងផ្លាស់ប្តូរមាតិកាជាទៀងទាត់។​ ភាសាសម្រាប់បង្កើត Dynamic Website មានដូចជា៖ PHP, Node.js, Python, Java, ASP.NET និងភាសាផ្សេងៗទៀត។

2. តើ PHP មានអត្ថប្រយោជន៍សម្រាប់ធ្វើអ្វី?

PHP មានអត្ថប្រយោជន៍សម្រាប់៖

* បង្កើតគេហទំព័រដែលមានលក្ខណះ Dynamic
* ភ្ជាប់និង Database មានដូចជា MySQL, PostgreSQL ជាដើម
* បង្កើត Web Application ដូចជា blog, Social Media, eCommerce
* អាចដំណើរការលើម៉ាស៊ីនជាច្រើន Linux, Window and macOS
* ងាយស្រួលក្នុងការប្រើប្រាស់ ដោយសារវាជា Open Source និងមានឯកសារច្រើន

3. ចូរនិយាយការបង្កើត Structure របស់ PHP​ និង ឧទាហរណ៍។

Structure របស់​ PHP ជារបៀបសរសរេកូដ នៅក្នុង extension (.php)

|  |
| --- |
| // syntax of php  <?php  // code php here  ?> |

ឧទាហរណ៍៖

|  |
| --- |
| // example php  <?php  echo “Example PHP”;  echo “Hello World!”;  ?> |

4. តើការបង្កើតអថេរ (Variable) នៅក្នុង​ PHP មានប៉ុន្មានប្រភេទអ្វីខ្លះ? ចូរពន្យល់។

ការបង្កើតអថេរ (Variable) នៅក្នុង​ PHP មានដូចជា៖ Array, Constant, String, Integer, Float and Boolean ។

វិធីដែលកំណត់ constant និង​ Array ក្នុង PHP មានដូចជា៖

* define​() function​: អថេរមិនផ្ទេរតម្លែបាន​ ​(តម្លៃថេរ)
* const keyword​​: ប្រើនៅខាងក្រៅ function or class
* Array: ផ្ទុកទិន្នន័យច្រើនក្នុងអថេរមួយ
* String
* Integer
* Float
* Boolean

ឧទាហរណ៍៖

|  |  |
| --- | --- |
| <?php  //1. String – Text data  $name = “Sok Dara”;  echo “My name is $name “;  //2. Integer – Whole numbers  $age = 20;  echo ”I am $age years old”;  //3. Float – Decimal numbers  $price = 9.99;  echo “The price is $$price “; | //4. Boolean – true or false  $isStudent = true;  If ($isStudent) {  echo “You are a student”;  } else {  Echo “You aren’t a student”;  }  //5. Array  $colors = array(“red”, “green”, “blue”);  echo “First color is: “ .$colors[0] .;  //6. Constant – a fixed value  Define (“PI”, 3.14159);  echo “Value of PI is: “ .PI. ;  // You can also use const  const APP\_NAME = “MyApp”;  echo “App Name: “ . APP\_NAME . ; |

$file = fopen("log.txt", "a");

fwrite($file, "User visited at " . date("Y-m-d H:i:s") . "\n");

fclose($file);

$conn = new mysqli("localhost", "root", "", "mydb");

$result = $conn->query("SELECT \* FROM users");

session\_start();

$\_SESSION["username"] = "Dara";

setcookie(“user”, “Dara”, time() + 3600);

12. តើអ្វីទៅជា Cookie, Session, File IO and Database? តើគេប្រើប្រាស់ដើម្បីធ្វើអ្វី?

ចូរលើកឧទាហរណ៍។

* Cookie: ជាប្រភេទ data storage មួយដែលមានតួនាទីផ្ទុកទិន្នន័យផ្នែកខាង Client (Web Browser) គិតជារយះពេលជាក់លាក់ណាមួយ អាចគិតជា វិនាទី នាទី ម៉ោង ឬ ថ្ងៃ។

ឧទាហរណ៍៖ ដើម្បីចងចាំទិន្នន័យអ្នកប្រើប្រាស់ (username, language, login state)

Cookie store 1 day:

* Session: ជាប្រភេទ data storage មួយដែលមានតួនាទីផ្ទុកទិន្នន័យខាង Server អំឡុងពេល User កំពុងប្រើប្រាស់ Application ឬ Web page​ ។

ឧទាហរណ៍៖ នៅពេលចូលប្រើ​ (login) ចងចាំទិន្នន័យរបស់អ្នកប្រើ

* File IO: ជាប្រភេទ data storage មួយដែលអនុញ្ញាតអោយ Read និង Write ទិន្នន័យនៅក្នុង file នៅលើ Server ។

ឧទាហរណ៍៖ កត់ Log, រក្សាទិន្នន័យ បង្កើត report

* Database: ជាប្រភេទ data storage មួយដែលមានតួនាទីរៀបចំទិន្នន័យអោយមានរបៀបរៀបរយ មានទម្រង់ជាតារាង និងគ្រប់គ្រងទិន្នន័យសម្រាប់ ប្រព័ន្ធធំៗ ជាអចិន្រ្ទៃយ៍។

ឧទាហរណ៍៖ រក្សាទុកទិន្នន័យរបស់អ្នកប្រើប្រាស់

14. ហេតុអ្វីបានជា Web Application ភាគច្រើនប្រើប្រាស់ Pattern MVC របស់ Laravel?

* មូលហេតុដែល Web Application ភាគច្រើនប្រើប្រាស់ Pattern MVC របស់ Laravel ដោយសារតែ MVC Pattern គឺជាវិធីសាស្រ្តមួយដែលបានបែងចែកជា ៣ ផ្នែកដែលមានដូចជា៖
* Model: គឺជាផ្នែកមួយដែលដំណាងអោយ Data និង business logic ដែលមានតួនាទីទទួលខុសត្រូវចំពោះការទាញយកទិន្នន័យពី database និងរក្សាទុកវា។
* View: ជាផ្នែកមួយដែលទាក់ទងនិងការបង្ហាញ UI (User Interface) ទៅកាន់អ្នកប្រើប្រាស់ និងបង្ហាញទិន្នន័យដែលផ្តល់ពី Model តាមរយះ Controller។
* Controller:​ គឺជាផ្នែកមួយដែលភ្ជាប់ទំនាក់ទំនងរវាង View និង Model​ មានតួនាទីទទួល ទិន្នន័យពី Request របស់ Client បញ្ជូនទៅ Model ហើយបង្ហាញតាមរយះ View​ ។

11. ចូរពន្យល់អំពីអត្ថប្រយោជន៍នៃការប្រើប្រាស់ Pattern MVC នៅក្នុងការអភិវឌ្ឍន៍គេហទំព័រ ។

* អត្ថប្រយោជន៍នៃការប្រើប្រាស់ Pattern MVC នៅក្នុងការអភិវឌ្ឍន៍គេហទំព័រមានដូចជា៖
* MVC បំបែកជាសមាសធាតុបីផ្សេងគ្នា (Model, View, Controller) ងាយស្រួលក្នុងការសរសេរកូដ
* ការកែប្រែ UI គឺមិនប៉ះពាល់ទៅលើ Logic និង database
* Reusability: Model និង Controller អាច reuse បានក្នុង project ឬ page ផ្សេងៗ
* បង្កើនផលិតភាពការងារ ងាយស្រួលយល់នៃ Structure
* ងាយស្រួល Test or Debug ព្រោះ Code ត្រូវបានបែងចែកច្បាស់លាស់ ងាយស្រួល Test logic ជាផ្នែកៗ ធានាបាននូវភាពត្រឹមត្រូវនៃកម្មវិធី។

15. ប្រៀបធៀប Static Website and Dynamic Website?

* ការប្រៀបចំណុចមួយចំនួនរវាង Static Website និង Dynamic Website មានដូចជា៖

|  |  |
| --- | --- |
| Static Website | Dynamic Website |
| មាតិការមានលក្ខណះថេរ មិនអាចផ្លាស់ប្តូរបាន | មាតិការប្រែប្រួលដោយផ្អែកលើ User ឬ ទិន្នន័យ |
| មិនមានការភ្ជាប់ជាមួយមូលដ្ឋានទិន្នន័យ(database) | មានការភ្ជាប់ជាមួយ Database (MySQL, PostgreSQL) |
| ប្រើប្រាស់ភាសា HTML, CSS, JavaScript ឬ Framework ផ្សេងៗទៀត | ប្រើប្រាស់ភាសា Python, PHP, Node.js, ASP.NET, Java និងភាសាផ្សេងៗទៀត |
| ត្រូវកែប្រែដោយដៃ លើរាល់ទំព័រនៃមាតិកា | អាចកែប្រែតែមួយកន្លែង ហើយមានការផ្លាស់ប្តូរលើទំព័រទាំងអស់ |
| មានល្បឿនលឿនជាង ព្រោះមិនមានទិន្នន័យពី database | យឺតជាង ត្រូវទាញយកទិន្នន័យពី database |

17. ចូរនិយាយពី ដំណើរការ (Process Flow) នៃ web-client (browser) និង web-server នៅពេល Client បើក Web page នៅក្នុង Laravel​។

* ដំណើរការនៃ web-client និង web-server នៅពេល Client បើក website មានដូចជា៖
* ​Client Request: Client វាយ URL(request) នៅលើ Browser បន្ទាប់មក Browser បញ្ចូន HTTP Request ទៅកាន់ web server ។
* Web Sever ទទួលសំណើ HTTP និងទៅត្រួតពិនិត្យមើល Route (Resource) ដើម្បីរក URL ដែលដូចនិងសំណើ បន្ទាប់ពីរកឃើញ វានិងបញ្ចូនទៅកាន់ Controller
* Controller ទទួលសំណើ និងធ្វើការដំណើរការ​ Model ឬ Business Logic
* Model ជាផ្នែកដែលទាក់ទងនិង database វាធ្វើការទាញយកទិន្នន័យនៅក្នុងតារាង និងបញ្ចូនទៅកាន់ View
* View បញ្ចូននូវទិន្នន័យទាំងនោះ និង HTML response ទៅកាន់ Browser ហើយបង្ហាញទៅកាន់អ្នកប្រើប្រាស់ (Client) ។

17. ប្រៀបធៀប $\_GET and $\_POST និងផ្តល់ឧទាហរណ៍។

|  |  |
| --- | --- |
| $\_GET | $\_POST |
| បញ្ចូនទិន្នន័យតាម URL (query string) | បញ្ចូនទិន្នន័យនៅក្នុង body របស់ HTTP Request |
| និន្នន័យដែលផ្ញើតាមរយះ GET អាចមើលឃើញនៅក្នុង URL | មិនអាចមើលឃើញទិន្នន័យនៅក្នុង URL (លាក់ក្នុង body) |
| មិនសូវមាន​ Security ព្រោះអាចView, Copy បាន | មាន Security ដែលអាចលាក់ទិន្នន័យ |
| ការផ្ញើទិន្នន័យមានកំណត់ | អាចផ្ញើទិន្នន័យបានច្រើនជាង |
| ឧទាហរណ៍៖  <!-- form\_GET.html -->  <form method="GET" action="get\_example.php">  <input type="text" name="name" placeholder="Enter your name">  <input type="submit" value="Submit">  </form>  <?php  ​​ echo "Hello, " . $\_GET['name'];  ?>  **URL:** get\_example.php?name=Chan | ឧទាហរណ៍៖  <!-- form\_POST.html -->  <form method="POST" action="post\_example.php">  <input type="text" name="username" placeholder="Username">  <input type="password" name="password" placeholder="Password">  <input type="submit" value="Login">  </form>  <?php  ​​​ echo "Username, " . $\_POST['username'];  echo "Password, " . $\_POST[‘password’];  ?> |

10. ចូរនិយាយ​ពីការ Call File មួយទៅ file មួយទៀត នៅក្នុង PHP និងលើកឧទាហរណ៍។

* នៅក្នុង PHP ផ្តល់នូវ functions 4 ប្រភេទសំខាន់ៗសម្រាប់ call file មួយទៅ file​ មួយទៀតដូចជា៖
* include: ប្រើសម្រាប់បញ្ចូនឯកសារទៅក្នុង Script PHP បើសិន file មិនមាន និងបង្ហាញ warning ប៉ុន្តែ script executive នៅតែបន្ត។
* require:​ ប្រើដើម្បីបញ្ចូនឯកសារ​ដូចនិង include ប៉ុន្តែបើមិនមាន file និងត្រូវបញ្ឃប់​។
* include\_once: ស្រដៀងនិង include ដែរ តែវាពិនិត្យមើល file បើមានរួចហើយ និងមិនបញ្ចូនម្តងទៀតទេ​។
* require\_once:​ ស្រដៀងនិង require ដែរតែវាពិនិត្យមើល file បើមានរួចហើយ និងមិនបញ្ចូនម្តងទៀតទេ​​​ (import once only) ។

ឧទាហរណ៍៖

|  |
| --- |
| 1. header.php  <?php  echo "<h1> Welcome to My PHP Website</h1>";  ?>  2. content.php  <?php  echo "<p>This is the main content of the page </p>";  ?>  3. index.php (Main File)  <?php  // Load header  include\_once 'header.php';  // Load main content  include 'content.php';  ?> |

13. ចូរនិយាយពីការភ្ជាប់ Database​ នៅក្នុង PHP និង លើកឧទាហរណ៍។

នៅក្នុង PHP យើងអាចភ្ជាប់ទៅកាន់ Database ដើម្បីទាញយក ឬផ្ទុកទិន្នន័យ។ មានវិធីភ្ជាប់ទៅកាន់ PHP 2 ប្រភេទធំៗដែលប្រើប្រាស់ជាធម្មតា៖

1. MySQLi (MySQL Improved): ជា សម្រាប់ភ្ជាប់ទៅកាន់ MySQL Database ដែល support បែប object-oriented and Procedural​​ ។

|  |
| --- |
| <?php  $host = "localhost";  $user = "root";  $password = "";  $dbname = "test\_db";    $conn = mysqli\_connect($host, $user, $password, $dbname);  ​​ if (!$conn) {  die("Connection failed: " . mysqli\_connect\_error());  }  echo "Connected successfully!";  ?> |

2. PDO (PHP Data Objects): ជា extension ប្រភេទ Object-Oriented support ជាមួយ database ច្រើនដូចជា MySQL, SQLite, PostgreSQL, MS SQL។

|  |
| --- |
| <?php  $host = "localhost";  $dbname = "test\_db";  $user = "root";  $password = "";  try {  $conn = new PDO("mysql:host=$host;dbname=$dbname", $user, $password);  $conn->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);  echo "Connected successfully!";  } catch (PDOException $e) {  echo "Connection failed: " . $e->getMessage();  }  ?> |

8. តើអ្វីទៅជា OOP? និយាយពីអត្ថប្រយោជន៍របស់ OOP ។ និងឧទាហរណ៍នៃការបង្កើត Class and Object​ នៅក្នុង PHP។

OOP (Object-Oriented-Programming) គឺជាវិធីសាស្រ្តសរសេរកម្មវិធីដោយផ្អែកលើ Object and Classes វាបំបែកកូដជាប្លុកៗ ដែលងាយស្រួលយល់និង អាចប្រើឡើងវិញបាន។

OOP មានសមាសធាតុសំខាន់ៗ​៤ យ៉ាងគឺ៖

* Encapsulation: បិទខ្ទប់ទិន្នន័យនៅក្នុង Object
* Inheritance: Object​ មួយទទួលលក្ខណះពី Object​ មួយទៀត
* Polymorphism: មុខងារនិងវិធីសាស្រ្តដូចគ្នាអាចមាន Behavior ផ្សេងគ្នា
* Abstraction: លាក់ព័ត៌មានលម្អិត និងបង្ហាញតែអ្វីដែលត្រូវការ

និយាយពីអត្ថប្រយោជន៍របស់ OOP៖

* ការរៀបចំកូដជាប្លុកៗ ឬជាក្រុម ងាយស្រួលក្នុងការគ្រប់គ្រង
* Object/Class អាចប្រើឡើងវិញបាននៅកន្លែងផ្សេងបាន
* ការកែសម្រួលមួយកន្លែង មិនប៉ះពាល់និងកន្លែងៗផ្សេងៗ
* Encapsulation ការពារទិន្នន័យមិនអោយ Access ដោយផ្ទាល់
* បង្កើនប្រសិទ្ធភាពការងារជាក្រុម ។

ឧទាហរណ៍៖​ ការបង្កើត Class និង Object នៅក្នុង PHP:

|  |
| --- |
| <?php  // បង្កើត Class ឈ្មោះ Car  class Car {  // Properties  public $brand;  public $color;  // Constructor function  public function \_\_construct($brand, $color) {  $this->brand = $brand;  $this->color = $color;  }  // Method  public function drive() {  return "Driving a " . $this->color . " " . $this->brand;  }  }  // បង្កើត Object ពី Class  $myCar = new Car("Toyota", "Red");  // ប្រើ Method  echo $myCar->drive(); // លទ្ធផល: Driving a Red Toyota  ?> |

5.​​​ តើការបង្កើត Function​ នៅក្នុង PHP មានអ្វីខ្លះ? ចូរពន្យល់។

Function គឺជាវិធីសាស្រ្តដែលបែងចែកកូដជាប្លុកៗ ងាយស្រួលក្នុងការហៅប្រើរាល់ពេលត្រូវការ។

ប្រភេទនៃ Function ​នៅក្នុង PHP មាន 2​ប្រភេទ គឺ៖

1. Built-in Function: ជា function ដែល PHP មានស្រាប់ (echo(), array\_push())

2. User-defined Function: ​ ជា function ដែលអ្នកសរសេរកូដបង្កើតដោយខ្លួនឯង​ (custom function)

ឧទាហរណ៍៖

|  |
| --- |
| <?php  // Create Function  function greet($name) {  return "Hello, " . $name . "!";  }  // Call Function  echo greet("Sokha"); // លទ្ធផល: Hello, Sokha!  ?> |

6. ចូរសរសេរពីការប្រើប្រាស់ for-each (control structure) នៅក្នុង PHP​​​ ។

Foreach គឺជា​ក្នុង ដែលប្រើប្រាស់សម្រាប់ Loop នៅលើ Array or Collection ដើម្បីទាញយកទិន្នន័យម្តងមួយៗ។

ឧទាហរណ៍៖

|  |
| --- |
| <?php  // Create List of Student  $students = ["Sokha", "Vannak", "Dara"];  // Use foreach  foreach ($students as $student) {  echo "Student: " . $student . "<br>";  }  ?> |

ចម្លើយលំហាត់

ក. សរសេរ Code PHP ដើម្បីភា្ជប់ (connect) ទៅកាន់ database

|  |
| --- |
| <?php  $host = "localhost";  $db = "mydb";  $user = "admin";  $pass = "222333";    // Create connection  $conn = new mysqli($host, $user, $pass, $db);    // Check connection  if ($conn->connect\_error) {  die("Connection failed: " . $conn->connect\_error);  }  ?> |

ខ. បញ្ចូល (Insert) ទិន្នន័យ រួច Save និង display on table list

- Before save to database Check data មានឬអត់?​ បើអត់មាន Save and display on list

Form សម្រាប់បង្កើត New Employee:

|  |
| --- |
| <center>  <h1>List of Employee</h1>  <form action="listEmployee.php" method="post">  Name: <input type="text" name="name" required><br><br>  Email: <input type="email" name="email" required><br><br>  Password: <input type="password" name="password" required><br><br>  <button type="submit">Save</button>  </form>  </center> |

បញ្ចូល (Insert) ទិន្នន័យ រួច Save and Check data already exists

|  |
| --- |
| // 2. Insert and check employee exists  if ($\_SERVER["REQUEST\_METHOD"] == "POST") {  if (isset($\_POST['name'])) {  $name = $\_POST['name'];  $email = $\_POST['email'];  $passd = $\_POST['password'];  // Check if employee already exists  $check = $conn->prepare("SELECT \* FROM tbuser WHERE name = ?");  $check->bind\_param("s", $name);  $check->execute();  $check->store\_result();  if ($check->num\_rows == 0) {  // Insert new employee if not exists  $insert = $conn->prepare("INSERT INTO tbuser (name, email,password) VALUES (?, ?, ?)");  $insert->bind\_param("sss", $name, $email, $passd);  $insert->execute();  $insert->close();  } else {  echo "<script>alert('Employee with this name already exists.')</script>";  }  $check->close();  }  } |

Fetch and Display data to the table list

|  |
| --- |
| $sql = "SELECT \* FROM tbuser";  $rs = $conn->query($sql);  echo "<table border='1' width='50%'>  <thead>  <tr>  <th>ID</th>  <th>Name</th>  <th>Email</th>  <th>Password</th>  </tr>  </thead>  <tbody>  ";  while ($row = $rs->fetch\_row()) {  echo "<tr>";  echo "<td>" . htmlspecialchars($row[0]) . "</td>";  echo "<td>" . htmlspecialchars($row[1]) . "</td>";  echo "<td>" . htmlspecialchars($row[2]) . "</td>";  echo "<td>" . htmlspecialchars($row[3]) . "</td>";  echo "</tr>";  }  echo "</tbody></table>"; |

គ.​ Select data ណាមួយ ធ្វើការកែប្រែ និង បញ្ជូនទៅ Database (Update)

|  |
| --- |
| // ចាប់ទិន្នន័យតាម id ដើម្បីបង្ហាញ form  if ($\_SERVER["REQUEST\_METHOD"] == "GET" && isset($\_GET['id'])) {  $id = $\_GET['id'];  $result = $conn->query("SELECT \* FROM tbuser WHERE id = $id");  $row = $result->fetch\_assoc();  }  // ប្រើ POST ដើម្បី update ទិន្នន័យ  if ($\_SERVER["REQUEST\_METHOD"] == "POST") {  $id = $\_POST['id'];  $name = $\_POST['name'];  $email = $\_POST['email'];  $password = $\_POST['password'];  $stmt = $conn->prepare("UPDATE tbuser SET name=?, email=?, password=? WHERE id=?");  $stmt->bind\_param("sssi", $name, $email, $password, $id);  if ($stmt->execute()) {  echo "Update Success! <a href='listEmployee.php'>Back to List</a>";  exit();  } else {  echo " Update Failed!";  }  }  ?>  បង្ហាញ Form Edit  <h2>Edit User</h2>  <form action="edit.php" method="POST">  <input type="hidden" name="id" value="<?= $row['id'] ?? '' ?>">  Name: <input type="text" name="name" value="<?= $row['name'] ?? '' ?>" required><br><br>  Email: <input type="email" name="email" value="<?= $row['email'] ?? '' ?>" required><br><br>  Password: <input type="text" name="password" value="<?= $row['password'] ?? '' ?>" required><br><br>  <button type="submit">Update</button>  </form> |

**For Bachelor's Degree Final Exam Coming Soon:**  
Wishing everyone the best of luck!  
May you achieve all your goals and succeed with confidence.

Enjoy your day and stay positive!

Credit by: noy rada