



Instituto Politécnico Nacional
Escuela Superior de Cómputo

2024 - 1
13/11/2023
2CV20

Tecnologías para la Web
Ambrosi Rincón Gallardo Luis Bernardo
Práctica no.7



Índice

Índice.....	1
Introducción.....	2
Desarrollo.....	3
Conclusiones.....	7



Introducción

SQL, PHP y CRUD son tres tecnologías esenciales para el desarrollo web. SQL (lenguaje de consulta estructurado) es un lenguaje de base de datos que se utiliza para almacenar, recuperar y manipular datos. PHP (preprocesador de hipertexto) es un lenguaje de programación del lado del servidor que se utiliza para crear páginas web dinámicas. CRUD (Crear, Leer, Actualizar y Eliminar) es un conjunto de operaciones básicas realizadas con datos en una base de datos.

SQL es un lenguaje estándar utilizado para comunicarse con bases de datos relacionales. Se utiliza para crear, leer, actualizar y eliminar datos de tablas de bases de datos. SQL también se utiliza para realizar consultas complejas sobre datos, como filtrar, ordenar y unir tablas.

PHP es un popular lenguaje de programación del lado del servidor que se utiliza para crear páginas web dinámicas. PHP se puede utilizar para conectarse a bases de datos, generar resultados HTML y procesar la entrada del usuario. PHP también se utiliza para desarrollar aplicaciones web, como sitios de comercio electrónico, sistemas de gestión de contenidos y sitios de redes sociales.

CRUD es un conjunto de operaciones básicas realizadas sobre datos en una base de datos. Las operaciones CRUD son esenciales para cualquier aplicación web que necesite almacenar y administrar datos.

- Crear: Insertar un nuevo registro en una tabla de base de datos.
- Leer: recuperar un registro de una tabla de base de datos.
- Actualización: modifica un registro existente en una tabla de base de datos.
- Eliminar: eliminar un registro de una tabla de base de datos.

SQL, PHP y CRUD juntos SQL, PHP y CRUD se pueden utilizar juntos para desarrollar aplicaciones web potentes y dinámicas. Por ejemplo, PHP se puede utilizar para crear una página web que permita a los usuarios ver, agregar, editar y eliminar registros de una base de datos.



XAMPP es un paquete de software gratuito y de código abierto que proporciona un entorno de desarrollo web completo. Incluye el servidor web Apache HTTP Server, la base de datos MariaDB, el lenguaje de scripting PHP y el lenguaje de scripting Perl. Todos estos componentes están preconfigurados y funcionan juntos sin problemas, lo que facilita la configuración y el inicio del desarrollo de aplicaciones web.

XAMPP es una opción popular para el desarrollo web porque es:

- Gratuito y de código abierto
- Fácil de instalar y usar
- Multiplataforma (se ejecuta en Windows, macOS y Linux)
- Incluye todos los componentes esenciales para el desarrollo web
- Admite una amplia gama de aplicaciones web

Desarrollo

En la práctica anterior se realizó un crud que se conecta a una base de datos usando el servicio de Xampp.

Para esta práctica vamos a reutilizar el código que se hizo para una página web que contiene un CV personal. Al final de dicha página se realizó un formulario para que la persona que vea nuestra página pueda contactarnos. Así que vamos a crear un método post para poder conectar dicho formulario con una base de datos, la cuál también tenemos que crearla; se solicitó también que se haga un crud para administrar la base de datos.

De nuevo se realizará la implementación usando Xampp.

Adaptación del código de CV para que mande el formulario a una base de datos:

1. Cambiamos el nombre por index.php y borramos el `<!DOCTYPE html>`
2. Añadimos el siguiente código, al principio de la página:

```
<?php
if (isset($_POST['submit'])) {
    require_once("db.php");
    $sql = $conn->prepare("INSERT INTO contact_me
(uname,email,Msubject,tmessage) VALUES (?, ?, ?, ?)");
```



```
$uname = $_POST['uname'];
$email= $_POST['email'];
$Msubject=$_POST['Msubject'];

$message=$_POST['message'];

$sql->bind_param("ssss",$uname, $email,$Msubject,$message);
if($sql->execute()) {
    $success_message = "Added Successfully";
} else {
    $error_message = "Problem in Adding New Record";
}

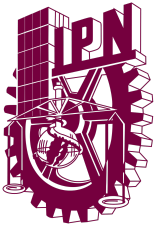
$sql->close();
$conn->close();
}
?>
```

Este código va a esperar que se realice la acción de post por medio del botón submit, establece una conexión con nuestra base de datos por medio del archivo db.php(se presenta más adelante en las instrucciones), por medio del método sql vamos a hacer un insert a nuestra tabla contac_me. Vamos a ingresar los datos de nombre de usuario (uname), email del usuario (email), el por qué del contacto (Msubject), y su mensaje (message).

3. En el form vamos a agregar method="post" el cual va a servir para mandar la información a la BD, también vamos a agregar los nombres de cada input según corresponda(uname,email,Msubject,message).

```
<section class="form-section" id="form-section">
    <div class="contact-form">
        <form name="frm" method="post" action="">
            <input type="text" name="uname" placeholder="Enter Your
Name" required>
            <input type="email" name="email" placeholder="Enter Your
Email" required>

            <input type="text" name="Msubject" placeholder="Enter
Your Subject">
            <textarea name="message" id="" cols="40" rows="18"
placeholder="Enter Your Message" required></textarea>
            <input type="submit" name="submit" value="Submit"
class="send">
        </form>
```



```
</div>
</section>
```

4. Creamos un archivo con el nombre de db.php y agregamos el siguiente código para poder establecer comunicación con nuestra base de datos.

```
<?php
    $conn =new mysqli('localhost', 'root', '' , 'portfolio');
?>
```

5. Iniciamos xampp junto con los servicios de apache y MySQL.
6. Accedemos a localhost y seleccionamos phpMyAdmin.
7. Damos click en New+.
8. El nombre de la base de datos será portfolio.
9. Hacemos click en go.
10. Seleccionamos nuestra base de datos portfolio y hacemos click en New+.
11. Seleccionamos SQL en las pestañas que se encuentran en la vista superior.
12. Añadimos el siguiente código y hacemos click en go.

```
CREATE TABLE IF NOT EXISTS `contact_me` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `uname` varchar(255) NOT NULL,
  `email` varchar(255) NOT NULL,
  `Msubject` varchar(255) NOT NULL,
  `tmessage` varchar(255) NOT NULL,
  PRIMARY KEY(`id`)
);
```

13. ¡La página debe funcionar!

14. Ingresamos Datos.

Contact Me

Let's Work Togheter

Im interested on working with you! Please email me to the following direction, or fill de contact form.

▶ lambrosi1800@alumno.ipn.mx

Prede
pedro@gmail.com

Trabajo

Hola, te dejo mi num para que trabajemos juntos
5564789912

Submit

15. Checamos en nuestra BD, ¡Funciona!.

	id	uname	email	Msubject	tmessage
<input type="checkbox"/> Edit Copy Delete	1	Jane	Fr@gmail.com	Conatco	Saludos
<input type="checkbox"/> Edit Copy Delete	4	Luis	starbucks@infinitemovil	Conatco	Saludos
<input type="checkbox"/> Edit Copy Delete	7	Prede	pedro@gmail.com	Trabajo	Hola, te dejo mi num para que trabajemos juntos 55...
<input type="checkbox"/> Edit Copy Delete	5	ADMIN	lbambrosi@gmail.com	Trabajo	Hola
<input type="checkbox"/> Edit Copy Delete	6	cam	starbucks@infinitemovil	Conatco	Hola
<input type="checkbox"/> Edit Copy Delete	3	Junaito	jnto@gmail.com	Contrato	Bienvenido

Adaptación del código del CRUD para administrar nuestra base de datos portfolio:

Para este caso ya contamos con una base de datos, en este caso en portfolio. así que tenemos que reutilizar el código de la práctica 6 y editar cada archivo cambiando el nombre de la base de datos por portfolio, cambiando también la tabla por contact_me.

1. En el archivo de db.php cambiamos el nombre de blog por portfolio.

```
<?php
    $conn =new mysqli('localhost', 'root', '' , 'portfolio');
?>
```

2. En el archivo de index.php, cambiamos la tabla tmp_emp_details por contact_me.

```
<?php
require_once("db.php");

$sql = "SELECT * FROM contact_me";
$result = $conn->query($sql);
//$conn->close();
```



?>

3. Cambiamos los datos de la tabla y agregamos campos, por Name, Email, Subject y Message. Dentro del código de php cambiamos los valores por id, uname, email, Msubject, tmessage.

```
<table class="tbl-qa">
  <thead>
    <tr>
      <th class="table-header" width="20%"> Name </th>
      <th class="table-header" width="20%"> Email </th>
      <th class="table-header" width="20%">Subject </th>
      <th class="table-header" width="20%">Message</th>
      <th class="table-header" width="20%"
colspan="2">Action</th>
    </tr>
  </thead>
  <tbody>
    <?php
      if ($result->num_rows > 0) {
        while($row = $result->fetch_assoc()) {
          ?>
          <tr class="table-row" id="row-<?php echo $row["id"]; ?>">
            <td class="table-row"><?php echo $row["uname"]; ?></td>
            <td class="table-row"><?php echo $row["email"]; ?></td>
            <td class="table-row"><?php echo $row["Msubject"];
?></td>

            <td class="table-row"><?php echo $row["tmessage"];
?></td>

            <!-- action -->
            <td class="table-row" colspan="2"><a
href="edit.php?id=<?php echo $row["id"]; ?>" class="link"></a> <a href="delete.php?id=<?php
echo $row["id"]; ?>" class="link"></a></td>
          </tr>
        <?php
        }
      }
    ?>
```




```
</tbody>
</table>
```

4. En el archivo de add.php. Cambiamos la tabla por contact_me e ingresamos los dato uname, email, Msubject y tmessage, según corresponda.

```
<?php
    if (isset($_POST['submit'])) {
        require_once("db.php");
        $sql = $conn->prepare("INSERT INTO contact_me
(uname,email,Msubject,tmessage) VALUES (?, ?, ?, ?)");
        $uname = $_POST['uname'];
        $email= $_POST['email'];
        $Msubject=$_POST['Msubject'];
        $tmessage=$_POST['tmessage'];
        $sql->bind_param("ssss",$uname, $email, $Msubject,$tmessage);
        if($sql->execute()) {
            $success_message = "Added Successfully";
        } else {
            $error_message = "Problem in Adding New Record";
        }
        $sql->close();
        $conn->close();
    }
?>
<html>
<head>
<link href="style.css" rel="stylesheet" type="text/css" />

<style>
.tbl-qa{border-spacing:0px;border-radius:4px;border:#6ab5b9 1px solid;}
</style>
<title>Add New</title>
</head>
<body>
<?php if(!empty($success_message)) { ?>
<div class="success message"><?php echo $success_message; ?></div>
<?php } if(!empty($error_message)) { ?>
<div class="error message"><?php echo $error_message; ?></div>
<?php } ?>
<form name="frmUser" method="post" action="">
```



```
<div class="button_link"><a href="index.php"> Back to  
List </a></div>  
<table border="0" cellpadding="10" cellspacing="0"  
width="500" align="center" class="tbl-qa">  
  <thead>  
    <tr>  
      <th colspan="2" class="table-header">Add New</th>  
    </tr>  
  </thead>  
  <tbody>  
    <tr class="table-row">  
      <td><label>Name</label></td>  
      <td><input type="text" name="uname" class="txtField"></td>  
    </tr>  
    <tr class="table-row">  
      <td><label>Email</label></td>  
      <td><input type="text" name="email" class="txtField"></td>  
    </tr>  
    <tr class="table-row">  
      <td><label>Subject</label></td>  
      <td><input type="text" name="Msubject"  
class="txtField"></td>  
    </tr>  
    <tr class="table-row">  
      <td><label>Message</label></td>  
      <td><input type="text" name="tmessage"  
class="txtField"></td>  
    </tr>  
    <tr class="table-row">  
      <td colspan="2"><input type="submit" name="submit"  
value="Submit" class="demo-form-submit"></td>  
    </tr>  
  </tbody>  
</table>  
</form>  
</body>  
</html>
```



5. En edit.php, creamos 5 POST, inicializamos con el nombre del dato (uname,email,Msubject,tmessage), en la tabla vamos a crear un campo para cada uno de ellos.

```
<?php
    require_once("db.php");
    if (isset($_POST['submit'])) {
        $sql = $conn->prepare("UPDATE contact_me SET uname=? , email=? ,
Msubject=? , tmessage=? WHERE id=?");
        $uname = $_POST['uname'];
        $email= $_POST['email'];
        $Msubject=$_POST['Msubject'];
        $tmessage=$_POST['tmessage'];
        $sql->bind_param("ssssi",$uname,
$email,$Msubject,$tmessage,$_GET["id"]);
        if($sql->execute()) {
            $success_message = "Edited Successfully";
        } else {
            $error_message = "Problem in Editing Record";
        }
    }

    $sql = $conn->prepare("SELECT * FROM contact_me WHERE id=?");
    $sql->bind_param("i",$_GET["id"]);
    $sql->execute();
    $result = $sql->get_result();
    if ($result->num_rows > 0) {
        $row = $result->fetch_assoc();
    }
    $conn->close();
?>

<html>
<head>
<link href="style.css" rel="stylesheet" type="text/css" />
<style>
.tbl-qa{border-spacing:0px;border-radius:4px;border:#6ab5b9 1px solid;}
</style>
<title>Edit Contact Info</title>
</head>
```



```
<body>
<?php if(!empty($success_message)) { ?>
<div class="success message"><?php echo
$success_message; ?></div>
<?php } if(!empty($error_message)) { ?>
<div class="error message"><?php echo $error_message; ?></div>
<?php } ?>
<form name="frmUser" method="post" action="">
<div class="button_link"><a href="index.php" >Back to List </a></div>
<table border="0" cellpadding="10" cellspacing="0" width="500"
align="center" class="tbl-qa">
  <thead>
    <tr>
      <th colspan="2" class="table-header">Contact Info Edit</th>
    </tr>
  </thead>
  <tbody>
    <tr class="table-row">
      <td><label>Name</label></td>
      <td><input type="text" name="uname" class="txtField"
value="<?php echo $row["uname"]?>"></td>
    </tr>
    <tr class="table-row">
      <td><label>Email</label></td>
      <td><input type="text" name="email" class="txtField"
value="<?php echo $row["email"]?>"></td>
    </tr>
    <tr class="table-row">
      <td><label>Subject</label></td>
      <td><input type="text" name="Msubject" class="txtField"
value="<?php echo $row["Msubject"]?>"></td>
    </tr>
    <tr class="table-row">
      <td><label>Message</label></td>
      <td><input type="text" name="tmessage" class="txtField"
value="<?php echo $row["tmessage"]?>"></td>
    </tr>
    <tr class="table-row">
      <td colspan="2"><input type="submit" name="submit"
value="Submit" class="demo-form-submit"></td>
```



```
</tr>
</tbody>
</table>
</form>
</body>
</html>
```

6. En delete.php solo cambiamos el nombre de la tabla por contact.me.

```
<?php
require_once("db.php");

$sql = $conn->prepare("DELETE FROM contact_me WHERE id=?");
$sql->bind_param("i", $_GET["id"]);
$sql->execute();
$sql->close();
$conn->close();
header('location:index.php');
?>
```

7. Ingresamos a la página y vemos que funciona correctamente.

Add New

Name	Email	Subject	Message	Action
Jane	Fr@gmail.com	Conatco	Saludos	 
Junaito	jnto@gmail.com	Contrato	Blenvenido	 
Luis	starbucks@infinittummovil	Conatco	Saludos	 
ADMIN	lbambrosi@gmail.com	Trabajo	Hola	 
cam	starbucks@infinittummovil	Conatco	Hola	 
Predo	pedro@gmail.com	Trabajo	Hola, te dejo mi num para que trabajemos juntos 5564789912	 

8. Realizamos cada función del crud, para agregar, editar, eliminar y vemos que funcionan adecuadamente.



Conclusiones

SQL, PHP y CRUD son tres tecnologías esenciales para el desarrollo web. SQL se utiliza para almacenar, recuperar y manipular datos en una base de datos. PHP es un lenguaje de programación del lado del servidor que se utiliza para crear páginas web dinámicas. CRUD es un conjunto de operaciones básicas realizadas sobre datos en una base de datos.

Cuando se usan juntos, SQL, PHP y CRUD se pueden utilizar para desarrollar aplicaciones web potentes y dinámicas. Por ejemplo, PHP se puede utilizar para crear una página web que permita a los usuarios ver, agregar, editar y eliminar registros de una base de datos.

SQL, PHP y CRUD son tecnologías populares que utilizan muchos desarrolladores web de todo el mundo. Si se está interesado en aprender a desarrollar aplicaciones web, entonces es importante aprender estas tres tecnologías.

XAMPP es un entorno de desarrollo web poderoso y versátil que es adecuado para desarrolladores de todos los niveles de experiencia. Es gratuito y de código abierto, fácil de usar y multiplataforma. Si se está buscando un entorno de desarrollo para crear aplicaciones web dinámicas e interactivas, XAMPP es una excelente opción.