



มหาวิทยาลัยราชภัฏนครปฐม
Nakhon Pathom Rajabhat University





มหาวิทยาลัยราชภัฏนครปฐม
Nakhon Pathom Rajabhat University

บทที่ 6 การเชื่อมต่อฐานข้อมูลและการสร้างแบบฟอร์ม รับข้อมูล

รายวิชา 3603610 การพัฒนาเว็บแอปพลิเคชันสำหรับสตาร์ทอัพ

ผศ.ดร.เดช รสสมศิริ





มหาวิทยาลัยราชภัฏนครปฐม
Nakhon Pathom Rajabhat University

คำอธิบายโค้ด PHP: conn.php และ add.php

วิเคราะห์และอธิบายโค้ดภาษา PHP อย่างละเอียด
พร้อมแนวทางการพัฒนาต่อยอด



conn.php: เชื่อมต่อฐานข้อมูล MySQL

- กำหนดตัวแปรเซิร์ฟเวอร์ ชื่อผู้ใช้ รหัสผ่าน และชื่อฐานข้อมูล
- ใช้ mysqli สร้างการเชื่อมต่อฐานข้อมูล
- ตรวจสอบการเชื่อมต่อ หากล้มเหลวจะแจ้ง error และหยุดโปรแกรม
- ตัวแปร \$conn จะเก็บการเชื่อมต่อฐานข้อมูลไว้ใช้ต่อ



conn.php: โค้ดและคำอธิบายทีละบรรทัด

- <?php
- \$servername = "localhost"; // ชื่อเซิร์ฟเวอร์ฐานข้อมูล
- \$username = "root"; // ชื่อผู้ใช้งานฐานข้อมูล
- \$password = ""; // รหัสผ่านผู้ใช้งานฐานข้อมูล (ว่างเปล่า)
- \$dbname = "mydb"; // ชื่อฐานข้อมูล
- // สร้างการเชื่อมต่อฐานข้อมูล
- \$conn = new mysqli(\$servername, \$username, \$password, \$dbname);
- // ตรวจสอบเชื่อมต่อ ถ้าล้มเหลวจะแสดง error และหยุด
- if (\$conn->connect_error) {
- die("Connection failed: " . \$conn->connect_error);
- }
- ?>



add.php: รับข้อมูลและเพิ่มข้อมูลลงฐานข้อมูล

- include('conn.php') เพื่อนำตัวแปร \$conn มาใช้เชื่อมต่อฐานข้อมูล
- ตรวจสอบว่ามีการส่งข้อมูลจากฟอร์มหรือไม่ ด้วย isset(\$_POST['submit'])
- รับค่าจากฟอร์ม ชื่อและอีเมล
- เขียนคำสั่ง SQL INSERT เพื่อเพิ่มข้อมูลลงในตาราง users
- รับคำสั่ง SQL และตรวจสอบผลลัพธ์
- แสดงข้อความยืนยันหรือตัวแจ้งข้อผิดพลาด
- ปิดการเชื่อมต่อฐานข้อมูล

add.php: โค้ดและคำอธิบายทีละบรรทัด

- <?php
- include('conn.php'); // นำเข้าไฟล์เชื่อมต่อฐานข้อมูล
- if (isset(\$_POST['submit'])) { // ตรวจสอบว่ามีการส่งข้อมูลจากฟอร์มหรือไม่
- \$name = \$_POST['name']; // รับค่าชื่อจากฟอร์ม
- \$email = \$_POST['email']; // รับค่าอีเมลจากฟอร์ม
- \$sql = "INSERT INTO users (name, email) VALUES ('\$name', '\$email')"; // คำสั่ง SQL เพิ่มข้อมูล
- if (\$conn->query(\$sql) === TRUE) {
- echo "New record created successfully"; // ข้อความเมื่อเพิ่มข้อมูลสำเร็จ
- } else {
- echo "Error: " . \$sql . "
" . \$conn->error; // ข้อความเมื่อเกิดข้อผิดพลาด
- }
- \$conn->close(); // ปิดการเชื่อมต่อ
- }
- ?>



ข้อควรระวังและคำแนะนำเพิ่มเติม

- โค้ดมีความเสี่ยง SQL Injection เนื่องจากรับค่าจากผู้ใช้โดยตรง
- ควรใช้ Prepared Statements เพื่อลดความเสี่ยง
- ตรวจสอบและกรองข้อมูลที่ได้รับมาก่อนบันทึก เช่น ตรวจสอบรูปแบบอีเมล
- แนะนำเพิ่มระบบแจ้งเตือนผู้ใช้และการนำทางหลังเพิ่มข้อมูล
- เพิ่มฟีเจอร์ CRUD เพื่อจัดการข้อมูลอย่างครบวงจร
- ปรับโครงสร้างเป็น OOP หรือ MVC เพื่อความยืดหยุ่นและดูแลรักษาง่าย

แนวทางพัฒนาต่อยอด

- ใช้ Prepared Statements ด้วย mysqli_stmt หรือ PDO
- สร้างฟอร์ม HTML พร้อมตรวจสอบข้อมูลฝั่งลูกค้า (client-side validation)
- จัดการข้อความแจ้งเตือนและสถานะโดยใช้ session
- เพิ่มระบบแสดงข้อมูลและแก้ไขข้อมูล (CRUD)
- แยกโค้ดและใช้โครงสร้าง MVC หรือ Framework เช่น Laravel
- เพิ่มระบบยืนยันตัวตน (Authentication) และสิทธิ์การใช้งาน (Authorization)



ทดลองปฏิบัติกัน



มหาวิทยาลัยราชภัฏนครปฐม
Nakhon Pathom Rajabhat University

ขอบคุณครับ

