SIT 103- DATABASE FUNDAMENTALS

INTERACTION WITH A DATABASE VIA A USER INTERFACE

In this project, I developed a **Customer Management System** using a combination of technologies, including XAMPP, MySQL, HTML, and PHP. The objective was to create a user-friendly interface for adding and managing customer records in a structured database.

1. **Environment Setup with XAMPP**

To facilitate the development process, I utilized **XAMPP**, a powerful software package that combines Apache, MySQL, and PHP. This setup allowed me to create a local server environment for testing and deployment without requiring an internet connection.

- **Apache**: This web server enabled me to serve web pages and handle HTTP requests.

- **MySQL**: I used MySQL to create and manage the relational database, where customer information would be stored.

- **PHP**: As the server-side scripting language, PHP was utilized to handle form submissions and interact with the database.

2. **Database Creation**

I created a **MySQL database** specifically for the customer management system. Within this database, I designed a table named `customers` with the following fields:

- **customer_id** (INT): Unique identifier for each customer.

- **name** (VARCHAR): The name of the customer.

- **age** (INT): The age of the customer.

- **email** (VARCHAR): The customer's email address.

- **phone_number** (VARCHAR): The customer's contact number.

Using SQL queries, I defined the structure of the table and populated it with existing customer records to ensure functionality during testing.

3. **Web Form Development**

To facilitate data entry, I created a **HTML form** (`customer.html`) using **Visual Studio Code**. The form includes the following fields:

- Customer ID
- Name
- Age
- Email
- Phone Number

Each field was marked as required, ensuring that users provide all necessary information before submission. The form utilized the POST method to securely send data to the server.

4. **Server-Side Processing with PHP**

The form data is processed by a PHP script (`customer.php`). This script validates the input and inserts the customer information into the MySQL database.

- **Input Validation**: I implemented checks to ensure all fields are filled out correctly before attempting to insert the data. This helps prevent errors and maintain data integrity.
- **Database Interaction**: Using PHP's MySQLi functions, the script connects to the database and executes an `INSERT` query to add new customer records.

5. **User Feedback and Error Handling**

Upon successful data entry, users receive a confirmation message indicating that the customer has been added. In the event of an error (such as missing fields or database connection issues), appropriate error messages are displayed to guide users in correcting their input.

Conclusion

This project not only strengthened my technical skills in web development but also enhanced my understanding of how to integrate front-end and back-end technologies effectively. By leveraging XAMPP, MySQL, HTML, and PHP, I successfully created a functional Customer Management System capable of handling customer records efficiently.

THE HTML CODE :

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <h2>Add New Customer</h2>
    <form action="customer.php" method="POST">
        <label for="customer_id">Customer ID:</label><br>
        <input type="number" id="customer_id" name="customer_id" required><br><br>

        <label for="name">Name:</label><br>
        <input type="text" id="name" name="name" required><br><br>

        <label for="age">Age:</label><br>
        <input type="number" id="age" name="age" required><br><br>

        <label for="email">Email:</label><br>
        <input type="email" id="email" name="email" required><br><br>

        <label for="phone_number">Phone Number:</label><br>
        <input type="text" id="phone_number" name="phone_number" required><br><br>
```

```
        <input type="submit" value="Add Customer">
    </form>

</body>
</html>
```

THE CUSTOMER.PHP CODE

```
<?php
// Database connection details
$servername = "localhost";
$username = "root"; // Change this if necessary
$password = ""; // Add your password
$dbname = "customer"; // Use the database name from your project

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

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

// Validate form data
if (isset($_POST['customer_id'], $_POST['name'], $_POST['age'], $_POST['email'],
$_POST['phone_number'])) {
    $customer_id = $_POST['customer_id'];
    $name = $_POST['name'];
    $age = $_POST['age'];
    $email = $_POST['email'];
    $phone_number = $_POST['phone_number'];
```

```
    // Check if the Customer_ID is unique
    $check_id_query = "SELECT Customer_ID FROM Customer WHERE Customer_ID =
$customer_id";
    $result = $conn->query($check_id_query);

    if ($result->num_rows > 0) {
        echo "Customer ID already exists. Please choose another ID.";
    } else {
        // Insert data into Customer table
        $sql = "INSERT INTO Customer (Customer_ID, Name, Age, Email, Phone_Number)
                VALUES ('$customer_id', '$name', '$age', '$email', '$phone_number')";

        if ($conn->query($sql) === TRUE) {
            echo "New customer added successfully";
        } else {
            echo "Error: " . $sql . "<br>" . $conn->error;
        }
    }
} else {
    echo "All fields are required!";
}

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

THE VIDEO LINK :

https://youtu.be/ZCw9PRvcMWs