PHP and MySQL Lab Manual: Building a Small Blog Application

Duration: 1:30 Hours

Introduction

- **Objective**: Introduction to building a simple blog application using PHP and MySQL.
- **Overview**: Discuss PHP and MySQL's role in web development and the importance of learning these skills.

Part 1: Setting Up the Database

- **Objective**: Create a MySQL database and a table for the blog application.
- **Tasks**:
    1. Open XAMPP and start the Apache and MySQL modules.
    2. Access PHPMyAdmin through your browser.
    3. Create a new database named 'blog'.
    4. Inside the 'blog' database, create a 'posts' table.
    5. Define fields: 'id' (int, primary, auto-increment), 'title' (varchar), 'content' (text), 'author' (varchar), 'created_at' (datetime).

Part 2: Creating a New Post (Using PDO)

- **Objective**: Insert blog posts into the database.
- **Tasks**:
    1. Write a PHP script to display an HTML form for blog post submission.
    2. Include fields in the form for 'title', 'content', and 'author'.
    3. After form submission, collect the data using PHP.
    4. Connect to the 'blog' database using PDO.
    5. Write and execute a PDO query to insert the data into the 'posts' table.

Part 3: Displaying Posts (Using MySQLi OOP)

- **Objective:** Fetch and display blog posts.
- **Tasks:**
    1. Write a PHP script using MySQLi in OOP style to retrieve posts.
    2. Connect to the 'blog' database and select all posts.
    3. Display each post in an HTML table with 'view', 'edit', and 'delete' buttons.

Part 4: Updating Posts (Using PDO)

- **Objective:** Update existing blog posts.
- **Tasks:**
    1. Add an 'edit' link for each post in the display table.
    2. The link redirects to an 'edit post' PHP script with the post ID as a parameter.
    3. The script retrieves the post's data from the database using PDO.
    4. Display an editable form pre-filled with the post's data.
    5. After form submission, use PDO to update the post in the database.

Part 5: Deleting Posts (Using MySQLi OOP)

- **Objective:** Delete blog posts.
- **Tasks:**
    1. Include a 'delete' button for each post in the display table.
    2. The button links to a 'delete post' PHP script with the post ID.
    3. Script uses MySQLi OOP to delete the specified post from the database.

Part 6: Searching for Posts

- **Objective**: Implement a search feature.
- **Tasks**:
    1. Create a search form with a dropdown and a text box.
    2. The dropdown contains options: 'id', 'title', 'content', and 'author'.
    3. The user selects a field and enters a query in the text box.
    4. The script searches the 'posts' table for matches on form submission.
    5. Display the search results in an HTML format.