**SCT221-0896/2022**

**KARIUKI EVANGELINE WAITHERA**

**JOMO KENYATTA UNIVERSITY OF AGRICULTURE AND TECHNOLOGY**

**DEPARTMENT OF IT**

**WEB DEVELOPMENT II ASSIGNMNET 1**

**1. Explain the difference between include() and require() in PHP. When would you use each? (3**

**Marks)**

* include(): Includes and evaluates the specified file. If the file is not found, it shows a warning and continues executing the script.
* require(): Also includes and evaluates the file. However, if the file is missing, it throws a fatal error and stops execution.

Use require() when the file is critical (e.g., database config).  
Use include() for optional files (e.g., headers, footers).

**2. Describe the purpose of sessions in PHP. How are they initiated and destroyed? (3 Marks)**

* Sessions store user-specific data (like login status) across multiple pages.
* Initiated with: session\_start();
* Destroyed with:

session\_unset(); // remove all session variables

session\_destroy(); // destroy the session

**3. Write a PHP script snippet to connect to a MySQL database named school\_db using MySQLi**

**extension. (3 Marks)**

<?php

$host = "localhost";

$user = "root";

$pass = "";

$db = "school\_db";

$conn = new mysqli($host, $user, $pass, $db);

if ($conn->connect\_error) {

die("Connection failed: " . $conn->connect\_error);

}

?>

**4. List and briefly explain three common SQL commands used for data manipulation. (3 Marks)**

* SELECT: Retrieves data from a table.  
  *Example*: SELECT \* FROM students;
* INSERT: Adds new records.  
  *Example*: INSERT INTO students (name, email) VALUES ('John', 'john@example.com');
* UPDATE: Modifies existing records.  
  *Example*: UPDATE students SET email = 'new@example.com' WHERE id = 1;

**5. What is SQL injection? How can it be prevented in PHP applications? (3 Marks)**

SQL Injection is a vulnerability where attackers manipulate SQL queries via user input to access or destroy data.

Prevention:

* Use prepared statements with mysqli or PDO.
* Validate and sanitize inputs using functions like filter\_input() or htmlspecialchars().

**6. Write a PHP script that retrieves all records from a table called students and displays their**

**names and email addresses in an HTML table. (5 Marks)**

<?php

$conn = new mysqli("localhost", "root", "", "school\_db");

$result = $conn->query("SELECT name, email FROM students");

echo "<table border='1'><tr><th>Name</th><th>Email</th></tr>";

while ($row = $result->fetch\_assoc()) {

echo "<tr><td>{$row['name']}</td><td>{$row['email']}</td></tr>";

}

echo "</table>";

$conn->close();

?>

**7. Create a simple HTML form with fields for username and password. Write the PHP code to process the form data, check if the submitted username exists in the users table, and display an appropriate message. (5 Marks)**

HTML:

<form method="post">

Username: <input type="text" name="username"><br>

Password: <input type="password" name="password"><br>

<input type="submit" value="Login">

</form>

PHP:

<?php

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

$conn = new mysqli("localhost", "root", "", "school\_db");

$username = $\_POST['username'];

$stmt = $conn->prepare("SELECT \* FROM users WHERE username = ?");

$stmt->bind\_param("s", $username);

$stmt->execute();

$result = $stmt->get\_result();

if ($result->num\_rows > 0) {

echo "Username exists.";

} else {

echo "Username not found.";

}

$stmt->close();

$conn->close();

}

?>

**8. Write a SQL statement to create a table named orders with the following fields: order\_id**

**(auto-increment primary key), product\_name (varchar), quantity (int), order\_date (date). Also,**

**demonstrate how to insert a new order record into this table. (5 Marks)**

-- Create the table

CREATE TABLE orders (

order\_id INT AUTO\_INCREMENT PRIMARY KEY,

product\_name VARCHAR(100),

quantity INT,

order\_date DATE

);

-- Insert a new order

INSERT INTO orders (product\_name, quantity, order\_date)

VALUES ('Laptop', 2, '2025-08-02');