1. **Explain about database connectivity and CRUD operations.**

Database connectivity refers to the mechanisms that allow applications to interact with databases. It involves establishing a connection between an application and a database management system (DBMS) to store, retrieve, update, and manage data.

Key Components of Database Connectivity:

1. **Database Driver/Connector**: Software that enables communication between the application and the database (e.g., JDBC for Java, ODBC, PDO for PHP)
2. **Connection String**: A string that contains information needed to establish a connection to the database (server name, database name, credentials, etc.)
3. **API (Application Programming Interface)**: Provides methods to interact with the database (e.g., JDBC API, ADO.NET)

Common Database Connectivity Technologies:

* **JDBC (Java Database Connectivity)**: For Java applications
* **ODBC (Open Database Connectivity)**: Platform-independent standard
* **ADO.NET**: For .NET applications
* **PDO (PHP Data Objects)**: For PHP applications
* **ORM (Object-Relational Mapping)**: Tools like Hibernate, Entity Framework

CRUD Operations

CRUD stands for the four basic operations performed on database data:

* **C**reate
* **R**ead
* **U**pdate
* **D**elete

These operations correspond to the SQL statements:

* INSERT (Create)
* SELECT (Read)
* UPDATE (Update)
* DELETE (Delete)

1. Create (INSERT)

Adds new records to a database table.

**SQL Example:**

INSERT INTO users (username, email) VALUES ('john\_doe', 'john@example.com');

**Java (JDBC) Example:**

String sql = "INSERT INTO users (username, email) VALUES (?, ?)";

PreparedStatement stmt = connection.prepareStatement(sql);

stmt.setString(1, "john\_doe");

stmt.setString(2, "john@example.com");

stmt.executeUpdate();

2. Read (SELECT)

Retrieves data from the database.

**SQL Example:**

SELECT \* FROM users WHERE id = 1;

**Java (JDBC) Example:**

String sql = "SELECT \* FROM users WHERE id = ?";

PreparedStatement stmt = connection.prepareStatement(sql);

stmt.setInt(1, 1);

ResultSet rs = stmt.executeQuery();

while(rs.next()) {

System.out.println(rs.getString("username"));

}

3. Update (UPDATE)

Modifies existing records in the database.

**SQL Example:**

UPDATE users SET email = 'new\_email@example.com' WHERE id = 1;

**Java (JDBC) Example:**

String sql = "UPDATE users SET email = ? WHERE id = ?";

PreparedStatement stmt = connection.prepareStatement(sql);

stmt.setString(1, "new\_email@example.com");

stmt.setInt(2, 1);

stmt.executeUpdate();

4. Delete (DELETE)

Removes records from the database.

**SQL Example:**

DELETE FROM users WHERE id = 1;

**Java (JDBC) Example:**

String sql = "DELETE FROM users WHERE id = ?";

PreparedStatement stmt = connection.prepareStatement(sql);

stmt.setInt(1, 1);

stmt.executeUpdate();

**Q.2. Perform the following operations using server-side scripting language(php).**

- Create database

- Create table in the database created above Insert data in the database (You can use following syntax for multiple insertion)

INSERT INTO table\_name (column\_list) VALUES (value\_list\_1), (value\_list\_2), ... (value\_list\_n);

- Select data and display them in HTML tables

- Delete data from database

<?php

$conn = mysqli\_connect('localhost', 'root', '');

if (!$conn) {

die("Connection failed: " . mysqli\_connect\_error());

}

$sql = "CREATE DATABASE IF NOT EXISTS student\_management";

if (mysqli\_query($conn, $sql)) {

echo "Database created successfully<br>";

} else {

echo "Error creating database: " . mysqli\_error($conn) . "<br>";

}

mysqli\_select\_db($conn, "student\_management");

$sql = "CREATE TABLE IF NOT EXISTS student (

ID INT PRIMARY KEY,

Name VARCHAR(50) NOT NULL,

Percentage DECIMAL(5,2) NOT NULL,

Location VARCHAR(50) NOT NULL,

DateOfBirth DATE NOT NULL

)";

if (mysqli\_query($conn, $sql)) {

echo "Table created successfully<br>";

} else {

echo "Error creating table: " . mysqli\_error($conn) . "<br>";

}

$sql = "INSERT INTO student VALUES

(1, 'Manthan Koli', 79, 'Delhi', '2003-08-26'),

(2, 'Dev Dixit', 75, 'Pune', '1999-06-17'),

(3, 'Aakash Deshmukh', 87, 'Mumbai', '1997-09-12')";

(4, 'Aaryan Jaiswal', 90, 'Chennai', '2005-10-02'),

(5, 'Rahul Khanna', 92, 'Ambala', '1996-03-04'),

(6, 'Pankaj Deshmukh', 67, 'Kanpur', '2000-02-02'),

(7, 'Gaurav Kumar', 84, 'Chandigarh', '1998-07-06'),

(8, 'Sanket Jain', 61, 'Shimla', '1990-09-08'),

(9, 'Sahil Wagh', 90, 'Kolkata', '1968-04-03'),

(10, 'Saurabh Singh', 54, 'Kashmir', '1989-01-06')";

if (mysqli\_query($conn, $sql)) {

echo "Data inserted successfully<br>";

}

echo "<h2>Student Records</h2>";

$result = mysqli\_query($conn, "SELECT \* FROM student");

echo "<table border='1'>

<tr>

<th>ID</th>

<th>Name</th>

<th>Percentage</th>

<th>Location</th>

<th>Date of Birth</th>

<th>Action</th>

</tr>";

while ($row = mysqli\_fetch\_assoc($result)) {

echo "<tr>

<td>{$row['ID']}</td>

<td>{$row['Name']}</td>

<td>{$row['Percentage']}</td>

<td>{$row['Location']}</td>

<td>{$row['DateOfBirth']}</td>

<td><a href='?delete={$row['ID']}'>Delete</a></td>

</tr>";

}

echo "</table>";

if (isset($\_GET['delete'])) {

$id = (int)$\_GET['delete'];

mysqli\_query($conn, "DELETE FROM student WHERE ID=$id");

echo "<script>alert('Record deleted'); location.href='';</script>";

}

mysqli\_close($conn);

?>

**Q.3. Create a login page that authenticate a user and redirects to a particular page.**

HTML PART

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title><?php echo $is\_logged\_in ? 'Dashboard' : 'Login'; ?></title>

</head>

<body>

<div class="container">

<?php if ($is\_logged\_in): ?>

<!-- Dashboard Content -->

<div class="dashboard">

<h2>Welcome, <?php echo htmlspecialchars($\_SESSION['user\_id']); ?>!</h2>

<p>You have successfully logged in.</p>

<p>This is your protected dashboard page.</p>

<a href="?action=logout"><button>Logout</button></a>

</div>

<?php else: ?>

<!-- Login Form -->

<div class="login-form">

<h2>Login</h2>

<?php if (isset($login\_error)): ?>

<div class="error"><?php echo $login\_error; ?></div>

<?php endif; ?>

<form method="POST" action="">

<div class="form-group">

<input type="text" name="username" placeholder="Username" required>

</div>

<div class="form-group">

<input type="password" name="password" placeholder="Password" required>

</div>

<button type="submit" name="login">Login</button>

</form>

</div>

<?php endif; ?>

</div>

</body>

</html>

CSS PART

body {

font-family: Arial, sans-serif;

max-width: 800px;

margin: 0 auto;

padding: 20px;

}

.container {

background: #f9f9f9;

padding: 20px;

border-radius: 5px;

box-shadow: 0 0 10px rgba(0,0,0,0.1);

}

.login-form {

max-width: 300px;

margin: 50px auto;

}

.form-group {

margin-bottom: 15px;

}

input[type="text"],

input[type="password"] {

width: 100%;

padding: 8px;

box-sizing: border-box;

}

button {

background: #4CAF50;

color: white;

padding: 10px 15px;

border: none;

cursor: pointer;

}

.error {

color: red;

margin-bottom: 15px;

}

.dashboard {

text-align: center;

}

PHP PART

<?php

session\_start();

// Simulated user database (replace with real database in production)

$valid\_users = [

'admin' => password\_hash('admin123', PASSWORD\_DEFAULT),

'user' => password\_hash('user123', PASSWORD\_DEFAULT)

];

// Handle logout

if (isset($\_GET['action']) && $\_GET['action'] === 'logout') {

session\_destroy();

header("Location: ?");

exit;

}

// Handle login form submission

if ($\_SERVER['REQUEST\_METHOD'] === 'POST' && isset($\_POST['login'])) {

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

$password = $\_POST['password'] ?? '';

if (isset($valid\_users[$username]) {

if (password\_verify($password, $valid\_users[$username])) {

$\_SESSION['user\_id'] = $username;

header("Location: ?");

exit;

}

}

$login\_error = "Invalid username or password";

}

// Check if user is logged in

$is\_logged\_in = isset($\_SESSION['user\_id']);

?>