Question 1 (25 Marks)

a) The World Wide Web (WWW) has become an integral part of modern society, revolutionizing the way people access information, communicate, and conduct business.

i. Explain the concept of the World Wide Web. (2 Marks)

The World Wide Web is a system of interlinked hypertext documents and resources, accessible via the internet. It allows users to view and navigate through multimedia content using web browsers by following hyperlinks between web pages. The WWW operates on protocols such as HTTP, enabling communication between servers and clients.

ii. Explain how the World Wide Web has changed the way people access information, communicate, and conduct business. (3 Marks)

The World Wide Web has fundamentally transformed:

- **Access to information**: It provides immediate access to vast amounts of information, enabling quick research and the dissemination of knowledge.
- **Communication**: Platforms like social media, emails, and instant messaging have become integral, enabling real-time global communication.
- **Business**: E-commerce platforms, online banking, and digital marketing have revolutionized the way businesses reach customers, sell products, and offer services, making global markets more accessible.

iii. Identify and discuss two (2) challenges associated with the World Wide Web. (4 Marks)

1. **Security and Privacy**: With the proliferation of online activities, data breaches, and identity theft have become major concerns. Users often face risks like phishing attacks and unauthorized data collection.
2. **Digital Divide**: Access to the World Wide Web is not universal. Many individuals in rural or underdeveloped regions face limited access, creating a gap in opportunities and education.

b) You are developing an ecommerce website called "FashionFusion" that specializes in selling a variety of fashion products online.

i. Suggest an appropriate URL for FashionFusion's website for the e-commerce part, that includes the subdomain, domain name, and path to a specific product category. (2 Marks)

Suggested URL: `https://shop.fashionfusion.com/clothing/women/dresses`

Explanation:

- `shop.`: Subdomain for the e-commerce platform.
- `fashionfusion.com`: Domain name for the main website.
- `/clothing/women/dresses`: Path that leads to a specific product category on the website.

ii. Draw a tree diagram showing the hierarchies of the root domain, top level domain, domain name, and subdomains for FashionFusion's e-commerce website. (6 Marks)

You would need to illustrate this, but here is a description of what to include:

- Root domain: `. (dot)`
- Top-level domain (TLD): `.com`
- Second-level domain (SLD): `fashionfusion`
- Subdomains: `shop` (for the e-commerce site), `support` (for customer support)

Tree Diagram Example:

```
Root
 |
 `-- .com
      |
      `-- fashionfusion
           |
           `-- shop (e-commerce)
           `-- support (customer support)
```

iii. Identify and explain two key characteristics of domain names and their significance for FashionFusion’s ecommerce website. Discuss how these characteristics contribute to identifying and accessing fashion products on the internet. (8 Marks)

1. **Uniqueness**: Domain names are unique, meaning no two websites can have the same name. This is crucial for branding and ensuring that customers can reliably access FashionFusion's website without confusion with other brands.
2. **Memorability**: A domain name should be easy to remember and type. `fashionfusion.com` is short, relevant to the fashion industry, and helps users easily recall the website when searching for fashion products online.

These characteristics help improve customer experience by allowing for easy navigation, recognition, and repeated visits to the site.

Question 2 (25 Marks)

a) You are a software engineer tasked with designing a new online banking system for a financial institution. The system needs to handle a large volume of transactions securely while providing a seamless user experience. To achieve this, you decide to implement a three-tier architecture.

i. Explain the role of the web browser, web server, and database server in the 3-tier client-server architecture. (6 Marks)

- **Web Browser**: The web browser acts as the client in this architecture. It is responsible for sending requests to the web server (through URLs) and displaying the results of those requests to the user. The web browser handles the presentation layer by rendering HTML, CSS, and JavaScript to create an interactive user interface for the online banking system.
- **Web Server**: The web server is the middle layer, responsible for receiving requests from the web browser, processing them, and sending the necessary commands to the database server. It handles business logic, such as processing user login credentials, verifying account balances, and performing transactions.
- **Database Server**: The database server stores and manages the data for the online banking system. This includes customer details, account balances, transaction records, and more. The database server processes queries sent from the web server and returns the requested data to be processed and sent back to the web

browser.

---

ii. Identify and discuss two challenges associated with the client-server architecture in the context of developing the online banking system. (4 Marks)

1. Security Risks: Since sensitive financial data is being transmitted between clients and servers, there is always a risk of attacks such as SQL injection, man-in-the-middle attacks, or data breaches. Implementing robust encryption and security protocols is critical.
2. Scalability Issues: As the number of users and transactions increases, the web and database servers might experience performance bottlenecks. Ensuring that the architecture can scale horizontally (adding more servers) or vertically (increasing server capacity) is important for maintaining a seamless user experience.

---

b) As part of developing the online banking system for the financial institution, you need to implement validations to ensure data integrity and security. These validations can be performed either at the client side or the server side.

i. Give two (2) differences between client-side scripting and server-side scripting. (4 Marks)

1. Location of Execution:
    - Client-side scripting: Executes in the user's web browser. Examples include JavaScript.
    - Server-side scripting: Executes on the web server before the response is sent to the client. Examples include PHP, Python, and Node.js.
2. Performance:
    - Client-side scripting: Reduces load on the server by handling some processing on the client's machine, leading to faster responses for certain tasks (e.g., form validations).
    - Server-side scripting: Ensures that processing is done securely on the

server, but can result in slower performance due to server load.

---

ii. Discuss where it is more appropriate to use client-side scripting and server-side scripting for validations in the online banking system. (4 Marks)

- Client-side scripting: Appropriate for initial form validations, such as checking that required fields are filled or that input formats (e.g., email addresses) are correct. This improves user experience by providing immediate feedback without needing to reload the page.
- Server-side scripting: Should be used for critical validations, such as verifying user credentials, account balances, and transaction requests. These validations are crucial for maintaining data integrity and security and should be done on the server to prevent manipulation on the client side.

---

iii. PHP can be utilized for developing the online banking system on the server-side. Identify and discuss five (5) key characteristics of PHP that make it suitable for server-side scripting in this context. (5 Marks)

1. Ease of Integration: PHP easily integrates with various databases such as MySQL, PostgreSQL, and Oracle, making it suitable for handling the database operations needed for an online banking system.
2. Security Features: PHP provides built-in mechanisms to prevent security vulnerabilities such as SQL injection, cross-site scripting (XSS), and cross-site request forgery (CSRF), making it ideal for a secure application.
3. Cost-Effective: PHP is open-source and has a vast ecosystem of free libraries and frameworks, which reduces development costs for building and maintaining the system.
4. Wide Hosting Support: Most web hosting providers support PHP out of the box, making it easier to deploy and manage the online banking system.
5. Fast Performance: PHP has a fast execution time, especially with the use of caching mechanisms like OPcache, ensuring that the system can handle a large volume of transactions efficiently.

---

iv. PHP allows us to create our own customized functions called user-defined functions. Give two (2) rules for creating a user-defined function. (2 Marks)

1. **Unique Function Name**: The name of the function must be unique within the scope in which it is declared to avoid conflicts with other functions or built-in functions.
2. **Proper Syntax**: A PHP function must start with the keyword `function`, followed by the function name, parentheses `()`, and a pair of curly braces `{}` where the function's code block resides. Example:

```
function calculateBalance($amount, $transaction) {
    return $amount + $transaction;
}
```

Question 3 (25 Marks)

a) Imagine you are a web developer tasked with building an online shopping platform for a retail company. The platform will allow users to browse products, add items to their shopping cart, and proceed to checkout to make purchases. As part of the development process, you need to implement session management and cookies to enhance user experience and streamline the shopping process.

i. Define PHP session. (2 Marks)

A PHP session is a way to store information (such as user data) on the server for later use. Unlike cookies, which store data on the user's browser, session data is stored on the server and associated with a unique session ID, which is passed between the server and the client (usually stored in a cookie or URL). Sessions help maintain state between different web pages.

---

ii. Explain two (2) ways you can use session management for the online shopping platform. (4 Marks)

1. **Shopping Cart**: Session management can be used to keep track of the items a user adds to their cart as they navigate through the website. This ensures that the cart data is preserved even if the user visits different pages before checkout.
2. **User Authentication**: Sessions can be used to manage user login status. After logging in, the user's session can store their credentials securely on the server, allowing them to navigate through the platform without needing to re-enter login details for every action.

---

iii. What PHP function is used to start the session? (1 Mark)

The function used to start a PHP session is `session_start()`.

---

iv. Define PHP cookies. (2 Marks)

Cookies in PHP are small pieces of data stored on the user's computer by their web browser. They are typically used to remember information about the user, such as login details or site preferences, and are sent with every HTTP request made to the server.

---

v. The syntax `setcookie(name, value, expire, path, domain, secure, httponly)` is used to set/create cookies in PHP. Discuss any 3 attributes from the `setcookie()` function, and explain their significance in creating and managing cookies. (6 Marks)

1. **name**: This is the name of the cookie, which is used as a key to retrieve its value. It's important because it identifies the cookie's purpose, such as storing session IDs or user preferences.
2. **expire**: This sets the expiration date and time for the cookie. If no expiration is set, the cookie will expire when the browser is closed. This is significant for determining how long the cookie should persist on the user’s machine.
3. **secure**: This ensures that the cookie is only sent over secure HTTPS connections. It is crucial for protecting sensitive data (such as session IDs) from being intercepted by attackers during transmission.

---

b) Websites and web systems are vulnerable to various types of attacks. Briefly explain the following types of attacks:

i. SQL Injection (2 Marks)

SQL Injection is an attack where an attacker inserts malicious SQL code into a query, allowing them to manipulate the database. This can lead to unauthorized access to data, data deletion, or bypassing authentication.

ii. Cross-Site Scripting (XSS) (2 Marks)

XSS is an attack where an attacker injects malicious scripts into web pages viewed by other users. These scripts can steal session cookies, redirect users to malicious sites, or perform actions on behalf of the user.

iii. Cookie/Session Hijacking (2 Marks)

In this attack, an attacker intercepts a user's session ID (often stored in a cookie) and uses it to impersonate the user, gaining unauthorized access to their account or session.

iv. Shell Script Upload (2 Marks)

Shell Script Upload occurs when an attacker uploads a malicious shell script (a type of command-line script) to a server, which can then be executed to gain control of the server or escalate privileges.

v. Brute Force/Dictionary Attack (2 Marks)

In a brute force or dictionary attack, an attacker attempts to gain unauthorized access to a system by systematically trying every possible combination of usernames and passwords (brute force) or using a predefined list of possible passwords (dictionary attack).

---

d) The form code provided is designed to collect details from the user, including employee ID, name, address, phone, and department, which will be submitted to a PHP script (`employee_reg.php`) to insert into the database.

The question asks to:

- Include the `connect.php` file to establish a database connection.
- Collect form data submitted by users.
- Construct an SQL statement to insert the collected form data into the `employees`

table.

PHP Code for `employee_reg.php`

```php
<?php
// Step 1: Include the connection file to connect to the database
include 'connect.php';  // Assumes connect.php contains the database connection code

// Step 2: Collect data from the form
$emp_id = $_POST['emp_id'];  // Employee ID
$name = $_POST['name'];      // Name
$address = $_POST['address']; // Address
$phone = $_POST['phone'];    // Phone number
$department = $_POST['department']; // Department

// Step 3: Construct an SQL statement to insert the data into the 'employees' table
$sql = "INSERT INTO employees (id, name, address, phone, department)
        VALUES ('$emp_id', '$name', '$address', '$phone', '$department')";

// Step 4: Execute the SQL query
if (mysqli_query($conn, $sql)) {
    echo "New employee record inserted successfully!";
} else {
    echo "Error: " . $sql . "<br>" . mysqli_error($conn);
}

// Close the database connection
mysqli_close($conn);
?>
```

Explanation:

1. **Include the connection file**: The `include 'connect.php';` line assumes there is a file called `connect.php` in the same directory. This file should contain the necessary PHP code to connect to the database (using `mysqli_connect()`).
2. **Collect form data**: The `$_POST` superglobal is used to retrieve the form data

submitted through the `POST` method. The input fields for employee ID, name, address, phone, and department are captured.

3. **Construct and execute the SQL statement**: An `INSERT INTO` SQL query is built to insert the form data into the `employees` table. The query inserts the collected form values into the corresponding columns.
4. **Error handling and success message**: The `mysqli_query()` function is used to execute the SQL statement. If the insertion is successful, a success message is displayed. If there is an error, the error message is printed using `mysqli_error()`.
