1. Q1 Past Year Question and Answer
    i. Original Question
    ii. Answers
2. Q2 Past Year Question and Answer
    i. Original Question
    ii. Answer
3. Q3 Past Year Question and Answer
    i. Question
    ii. Answer
    iii. Complete Code
4. Q4 Past Year Question and Answer
    i. Original Question
    ii. Answer

Q1 Past Year Question and Answer

Original Question

Part (a)

Differentiate between HTTP and HTTPS. (6 marks)

Part (b)

(i) What is the main purpose of a stateful protocol?

(ii) Explain how the protocol meets the purpose.

Part (c)

With the aid of a diagram, briefly describe the interaction process between web server and a web client.

Part (d)

Identify FOUR (4) advantages of designing dynamic web pages compared to static web pages.

Answers

Part (a)

HTTP (HyperText Transfer Protocol) is used for transmitting web pages over the internet but does not encrypt the data, making it vulnerable. It uses port 80 for communication.
HTTPS (HyperText Transfer Protocol Secure) is HTTP with encryption and uses SSL/TLS to secure the data transfer, ensuring that data is encrypted and secure from eavesdroppers. It uses port 443 for communication.

HTTP	HTTPS
Unencrypted	Encrypted
Uses port 80	Uses port 443
Vulnerable to attacks	Secure data transfer
No SSL/TLS	Uses SSL/TLS for encryption

Part (b)

(i) The main purpose of a stateful protocol is to maintain the state or context of a session between the client and server. This allows the server to remember information about the ongoing interaction, such as user-specific data, session history, and context, to provide a personalized and continuous experience for the user.

For example, in a shopping cart application, the server needs to remember the items added to the cart, the user's login status, and other session-specific information to provide a seamless shopping experience.

> ⓘ Note

> Stateless protocols do not maintain the state of the interaction, treating each request as independent, which can be inefficient for maintaining context and session information.
> For example, HTTP is a stateless protocol, and each request is processed independently without any knowledge of previous requests.

(ii) Stateful protocols meet this purpose by establishing and maintaining a session or connection between the client and server, allowing them to exchange information and maintain the state of the interaction throughout the session.

Part (c)

1. Client sends a request: The web client (browser) sends an HTTP/HTTPS request to the web server.
2. Server processes the request: The web server receives the request, processes it, and retrieves the requested resource (e.g., HTML page, data from a database).
3. Server sends a response: The web server sends the HTTP/HTTPS response back to the web client, including the requested resource.
4. Client renders the response: The web client receives the response and renders the content on the user's screen.

Part (d)

1. Personalization: Dynamic web pages can be tailored to individual users, displaying personalized content based on user preferences, behaviors, and interactions.
2. Interactivity: Dynamic web pages can include interactive elements such as forms, animations, and real-time updates, enhancing user engagement and providing a richer user experience.
3. Content Management: Dynamic web pages allow for easy content management and updates, enabling website owners to add, edit, and remove content dynamically without changing the underlying structure of the web page.

4. Integration with Databases: Dynamic web pages can retrieve and display data from databases, enabling dynamic content generation based on user input or database queries.

Q2 Past Year Question and Answer

Original Question

Part (a)

(i) Write PHP codes to store the following information shown in Table 1 into an associative array. Use `$state` as the variable name.

Table 1: Main Campus & Branches

State Abbreviation	State Name
JH	Johor
KL	Kuala Lumpur
PH	Pahang
PN	Penang
PR	Perak
SB	Sabah

(ii) Write PHP code to display the `$state` array excluding "KL" (Kuala Lumpur).

Part (b)

(i) In PHP, what is the purpose of the `implode` and `explode` functions?

(ii) Given a string: `$branches = "JH-PH-PN-PR-SB";` Convert it to an array and display it.

Part (c)

The following PHP codes are written:

```
$msg = "I like C++ programming";
echo "Calculate a product of two numbers" . "<br>";
calProduct(2, 3);
```

(i) Based on the given function `calProduct()` , write PHP codes in the function to multiply two numbers, and then display the product (result).

(ii) Write PHP codes to replace the word "C++" to "PHP".

Answer

Part (a)

(i) To store the information shown in Table 1 into an associative array, we use the following PHP code:

```
$state = array(
    "JH" => "Johor",
    "KL" => "Kuala Lumpur",
    "PH" => "Pahang",
    "PN" => "Penang",
    "PR" => "Perak",
    "SB" => "Sabah"
);
```

Explanation:

- The `$state` array is an associative array that stores the state abbreviation as the key and the state name as the value.

(ii) To display the `$state` array excluding "KL" (Kuala Lumpur), we use the following PHP code:

```
foreach ($state as $abbreviation => $name) {
    if ($abbreviation != "KL") {
        echo "$name ($abbreviation)<br>";
    }
}
```

Explanation:

- The `foreach` loop iterates through each element in the `$state` array.
- The `if` statement checks if the state abbreviation is not equal to "KL".
- If the condition is met, it displays the state name and abbreviation.
- Therefore, the output will be:

  ```
  Johor (JH)
  Pahang (PH)
  Penang (PN)
  Perak (PR)
  Sabah (SB)
  ```

Part (b)

(i) The purpose of the `implode` and `explode` functions in PHP is as follows:

- `implode` : Purpose of this function is used to join array elements into a single string using a specified delimiter. For example:

```
$array = array("JH", "PH", "PN", "PR", "SB");
$string = implode("-", $array);
// $string will be "JH-PH-PN-PR-SB"
```

Use implode when you have an array of items that you want to represent as a single string (e.g., converting an array of words into a comma-separated list).

- `explode` : This function is used to split a string into an array based on a specified delimiter. For example:

```
$string = "JH-PH-PN-PR-SB";
$array = explode("-", $string);
// $array will be array("JH", "PH", "PN", "PR", "SB")
```

Use explode when you have a string that you want to break into an array of elements (e.g., parsing a CSV line into its component values).

(ii) Given a string `$branches = "JH-PH-PN-PR-SB";` To convert it to an array and display it, we use the following PHP code:

```
$branches = "JH-PH-PN-PR-SB";
$branch_array = explode("-", $branches);
print_r($branch_array);
```

Explanation:

- The `explode` function is used to split the string `$branches` into an array based on the delimiter "-".

Part (c)

(i) Based on the given function `calProduct()` , to multiply two numbers and display the product, we use the following PHP code:

```
function calProduct($num1, $num2) {
    $result = $num1 * $num2;
    echo "The product of $num1 and $num2 is: $result";
}

$msg = "I like C++ programming";
echo "Calculate a product of two numbers" . "<br>";
calProduct(2, 3);
```

> **Caution**

> The function `calProduct()` should be defined before calling it in the code.

Explanation:

- The `calProduct()` function takes two parameters `$num1` and `$num2`, multiplies them, and stores the result in the variable `$result`.
- The product of the two numbers is then displayed using `echo`.
- In this case, the function `calProduct(2, 3)` is called with the arguments 2 and 3, resulting in the output "The product of 2 and 3 is: 6".

(ii) To replace the word "C++" with "PHP", we use the following PHP code:

```php
$msg = "I like C++ programming";
$msg = str_replace("C++", "PHP", $msg);
echo $msg;
```

Explanation:

- The `str_replace` function is used to replace all occurrences of the word "C++" with "PHP" in the string `$msg`.

> **Note**
>
> If you only want to replace the first occurrence, you can use `str_replace("C++", "PHP", $msg, 1)`.

Q3 Past Year Question and Answer

Question

Part (a)

Based on the following PHP codes, create a sticky form for the Name and Email fields.

```
$name = $_POST["name"];
$email = $_POST["email"];


<form method="" action="">
    Name: <input type="text" name="name">
    Email: <input type="text" name="email">
    <input type="submit" name="submit">
</form>
```

Part (b)

Answer the following questions based on the table named "campus" shown in Table 2. The table has been created and stored in the MySQL database.

Table 2: campus

CampusID	CampusName
C1	Main Campus
C2	Johor
C3	Pahang
C4	Penang
C5	Perak
C6	Sabah

(i) Write PHP codes to connect the MySQL database with the following information and disconnect it if the connection fails. (7 marks)

```
Username = tarumt
Password = 123
```

```
Database = tarumtDB
Host = localhost
```

(ii) Write SQL codes to search the Campus table with a specific CampusID ($CampusID) submitted by a user. Assign it to a PHP variable $sql_searchCampus. (3 marks)

(iii) Based on the codes written in Question 3b) (i) and Question 3b) (ii), write PHP codes to query the database (tarumtDB). Assign the query result to a PHP variable named $result_searchCampus. (2 marks)

(iv) Write PHP codes to display the campus name (Campus Name) if the record is found. Otherwise, show the message "Record not found". (5 marks)

(v) Write PHP codes to update "KL" to "Kuala Lumpur". Assign the codes of updating to a PHP variable named $sql_updateCampus, and the query result to a PHP variable named $result_updateCampus. (5 marks)

Answer

Part (a)

```php
<?php
// Check if the form is submitted
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // Get the form values
    $name = htmlspecialchars($_POST["name"]);
    $email = htmlspecialchars($_POST["email"]);
} else {
    // Initialize variables if the form is not submitted
    $name = "";
    $email = "";
}
?>


<form method="post" action="">
    Name: <input type="text" name="name" value="<?php echo $name; ?
    Email: <input type="text" name="email" value="<?php echo $email
    <input type="submit" name="submit" value="Submit">
</form>
```

> **Note**
>
> Sticky form is a web form that retains the information entered by the user even after the form is submitted. This is achieved by pre-filling the form fields with the previously entered values.

Exaplantion:

- The PHP code checks if the form is submitted using `$_SERVER["REQUEST_METHOD"] == "POST"` .
- If the form is submitted, it retrieves the values of the Name and Email fields using `$_POST["name"]` and `$_POST["email"]` , respectively.
- The `htmlspecialchars()` function is used to prevent XSS attacks by converting special characters to HTML entities. This helps to sanitize the input data. Example: `<` becomes `&lt;` .

Part (b)

(i) Connect to the MySQL database and disconnect if the connection fails.

```
<?php
$host = "localhost";
$username = "tarumt";
$password = "123";
$database = "tarumtDB";

// Create connection
$conn = mysqli_connect($host, $username, $password, $database);

// Check connection
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}
?>
```

Explanation:

- The PHP code establishes a connection to the MySQL database using the `mysqli_connect()` function.
- The connection details such as the host, username, password, and database name are provided as parameters to the function.
- If the connection fails, the `die()` function is called to display an error message along with the error description obtained from `mysqli_connect_error()`.
- The `mysqli_connect()` function returns a MySQLi object representing the connection to the database.
- The connection object is stored in the variable `$conn` for further database operations.
- The connection will be automatically closed when the script finishes executing.

(ii) Search the Campus table with a specific CampusID ($CampusID) submitted by a user.

```
<?php
$CampusID = $_POST['CampusID'];
$sql_searchCampus = "SELECT * FROM campus WHERE CampusID = '$Campus
?>
```

Explanation:

- Obtain CampusID from user input: Assume $CampusID is obtained from user input using the $_POST superglobal.
- SQL query: Construct the SQL query string to search for the specific CampusID in the "campus" table. The WHERE clause is used to filter the records based on the provided CampusID.

(iii) Query the database (tarumtDB) and assign the query result to a PHP variable named $result_searchCampus.

```php
<?php
$result_searchCampus = mysqli_query($conn, $sql_searchCampus);
?>
```

Explanation:

- Search Query: Construct the SQL query to search for the specific CampusID in the campus table and execute it using mysqli_query(). Assign the result to $result_searchCampus.

(iv) Display the campus name (Campus Name) if the record is found. Otherwise, show the message "Record not found".

```php
<?php
if (mysqli_num_rows($result_searchCampus) > 0) {
    while($row = mysqli_fetch_array($result_searchCampus)) {
        echo "Campus Name: " . $row["CampusName"] . "<br>";
    }
} else {
    echo "Record not found";
}
?>
```

Explanation:

- Check if any records are found. If records are found, fetch the results using mysqli_fetch_array() and display the campus name. If no records are found, display "Record not found".

(v) Update "KL" to "Kuala Lumpur". Assign the codes of updating to a PHP variable named $sql_updateCampus, and the query result to a PHP variable named $result_updateCampus.

```php
<?php
$sql_updateCampus = "UPDATE campus SET CampusName = 'Kuala Lumpur'
```

```
$result_updateCampus = mysqli_query($conn, $sql_updateCampus);
?>
```

Explanation:

- Construct the SQL query to update the campus name from "KL" to "Kuala Lumpur". Execute the query and assign the result to $result_updateCampus. Check if the update was successful and display an appropriate message.

Complete Code

```
<?php
// Re-establish the database connection if necessary (assuming the
$conn = mysqli_connect($host, $username, $password, $database);

// Check connection
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}

// SQL query to update "KL" to "Kuala Lumpur"
$sql_updateCampus = "UPDATE campus SET CampusName = 'Kuala Lumpur'

// Execute the update query and assign the result to $result_update
$result_updateCampus = mysqli_query($conn, $sql_updateCampus);

// Check if the update was successful
if ($result_updateCampus) {
    echo "Record updated successfully";
} else {
    echo "Error updating record: " . mysqli_error($conn);
}

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

Q4 Past Year Question and Answer

Original Question

Part (a)

There are two methods of passing values to a PHP script: HTML's hidden input type, and URL's appended value.

(i) Write a code to show the use of HTML's hidden input type to pass a value discount when a submit button is clicked.

(ii) Write a code to carry a parameter named status and its value discount by appending them to confirm.php when a submit hyperlink is clicked.

Part (b)

What is the difference between using POST and GET methods in the following code?

```
<form action="confirm.php" method="post"/"get">
<input type="text" name="name">
<input type="submit" name="button" value="Submit">
</form>
```

Part (c)

Explain TWO (2) main differences between cookie and session.

Part (d)

In jQuery event, hover() is the combination of mouseenter() and mouseleave(). Write a jQuery to show the respective alert messages "ENTER" and "LEAVE" when a mouse enters and leaves the following HTML element.

```html
<p class="special"> AMIT2043 </p>
```

Part (e)

Describe the operation of AJAX technology.

Answer

Part (a)

(i) Use of HTML's hidden input type to pass a value discount when a submit button is clicked.

```html
<form action="process.php" method="post">
  <input type="hidden" name="discount" value="20">
  <input type="submit" value="Submit">
</form>
```

(ii) Carry a parameter named status and its value discount by appending them to confirm.php when a submit hyperlink is clicked.

```html
<a href="confirm.php?status=discount">Submit</a>
```

Part (b)

POST method sends data in the body of the request, making it more secure for sensitive information and allowing larger amounts of data to be sent. It is not visible in the URL.

GET method appends data to the URL, making it visible and easily shareable, but is limited in length and less secure for transmitting sensitive data.

POST

- Sends data in the body of the request

- More secure for sensitive information
- Allows larger amounts of data to be sent
- Not visible in the URL

GET

- Appends data to the URL
- Visible and easily shareable
- Limited in length
- Less secure for transmitting sensitive data

Part (c)

1. Storage Location: Cookies are stored on the client's browser, whereas sessions are stored on the server.
2. Lifetime: Cookies can persist for a specified duration even after the browser is closed, whereas sessions typically expire once the user closes the browser or after a predefined period of inactivity.

Part (d)

jQuery to show the respective alert messages "ENTER" and "LEAVE" when a mouse enters and leaves the following HTML element.

```
<p class="special">AMIT2043</p>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jqu
<script>
  $(document).ready(function() {
    $(".special").hover(function() {
      alert("ENTER");
    }, function() {
      alert("LEAVE");
    });
  });
</script>
```

Explanation:

- The `hover()` function in jQuery is a combination of `mouseenter()` and `mouseleave()` events.
- When the mouse enters the `.special` element, the first function is executed to show the "ENTER" alert message.
- When the mouse leaves the `.special` element, the second function is executed to show the "LEAVE" alert message.

Part (e)

AJAX (Asynchronous JavaScript and XML) allows web pages to be updated asynchronously by exchanging small amounts of data with the server behind the scenes. This means that parts of a web page can be updated without reloading the entire page, providing a smoother and faster user experience. It involves using the XMLHttpRequest object to send and receive data asynchronously.

AJAX operations typically involve the following steps:

1. An event triggers an AJAX request, such as a button click or page load.
2. The XMLHttpRequest object is created and configured with the necessary parameters, such as the URL and request type.
3. The request is sent to the server asynchronously.
4. The server processes the request and sends a response back to the client.