Question 1

Question 1a

Identify and explain four (4) web protocols that will be used in the website design. For each protocol, explain its function and how it contributes to the website's functionality.

1. HTTP (Hypertext Transfer Protocol):
    - Function: HTTP is the protocol used for transmitting web pages over the internet. It allows web browsers to communicate with web servers.
    - Contribution: Enables users to access the coffee shop's website, browse the menu, and view other content.
2. HTTPS (Hypertext Transfer Protocol Secure):
    - Function: HTTPS is the secure version of HTTP, which encrypts data transmitted between the web browser and server.
    - Contribution: Ensures secure communication for sensitive transactions like online orders and reservations, protecting customer data.
3. FTP (File Transfer Protocol):
    - Function: FTP is used for transferring files between a client and a server on a computer network.
    - Contribution: Allows the website administrators to upload and manage website files on the server, such as updating the menu or adding new content.
4. SMTP (Simple Mail Transfer Protocol):
    - Function: SMTP is used for sending emails.
    - Contribution: Facilitates email communication for order confirmations, reservation notifications, and customer service inquiries.

Question 1b

Explain the process of how a web browser retrieves information from a web server.

1. DNS Lookup: The browser sends a request to a DNS server to resolve the domain name (e.g., www.coffeeshop.com) to an IP address.
2. TCP Connection: The browser establishes a TCP connection with the web server

using the resolved IP address.
3. HTTP Request: The browser sends an HTTP request to the web server, requesting the specific resource (e.g., an HTML page).
4. Server Response: The web server processes the request and sends an HTTP response back to the browser, containing the requested resource.
5. Rendering: The browser receives the response, parses the HTML, CSS, and JavaScript, and renders the webpage for the user to view.

Question 1c

Describe three (3) functions of a web server in the context of the coffee shop website.

1. Serving Web Pages: The web server hosts the website's HTML, CSS, JavaScript, and image files, delivering them to users when requested.
2. Processing Requests: The web server processes client requests, such as retrieving the menu or processing an online order.
3. Database Interaction: The web server interacts with the database to store and retrieve data, such as customer orders, reservations, and user accounts.

Question 1d

Explain how an ISP plays a role in accessing the coffee shop website.

- Internet Service Provider (ISP): The ISP provides internet connectivity to both the coffee shop's web server and the users attempting to access the website. ISPs route data between the user's device and the web server, enabling access to the website's content.

Question 1e

Write a CSS code snippet that implements this hover effect, using the following specifications:

- The background color of the link should change to a light blue (#ADD8E6) when the user hovers over it.
- The text color of the link should change to white when the user hovers over it.
- The hover effect should be applied to all links in the navigation menu.
- Your code snippet should be written in three (3) different styles: inline, internal, and

external, and should be properly labeled to indicate which style is which.

Inline CSS:

```
<a href="#" style="background-color: #f0f0f0; color: black;" onmouseover="this.style.backg
```

Internal CSS:

```
<!DOCTYPE html>
<html>
<head>
    <style>
        .nav a {
            background-color: #f0f0f0;
            color: black;
            padding: 10px;
            text-decoration: none;
        }
        .nav a:hover {
            background-color: #ADD8E6;
            color: white;
        }
    </style>
</head>
<body>
    <div class="nav">
        <a href="#">Home</a>
        <a href="#">Menu</a>
        <a href="#">Reservations</a>
        <a href="#">Contact</a>
    </div>
</body>
</html>
```

External CSS:

styles.css

```css
.nav a {
    background-color: #f0f0f0;
    color: black;
    padding: 10px;
    text-decoration: none;
}
.nav a:hover {
    background-color: #ADD8E6;
    color: white;
}
```

index.html

```html
<!DOCTYPE html>
<html>
<head>
    <link rel="stylesheet" type="text/css" href="styles.css">
</head>
<body>
    <div class="nav">
        <a href="#">Home</a>
        <a href="#">Menu</a>
        <a href="#">Reservations</a>
        <a href="#">Contact</a>
    </div>
</body>
</html>
```

This CSS code snippet implements the hover effect for the navigation menu links using inline, internal, and external styles.

Question 2

Question 2a

Briefly explain the concept of server-side scripting and its importance in web application development.

- Concept of Server-Side Scripting: Server-side scripting involves writing code that runs on a web server to generate dynamic web content. This code interacts with databases, processes user input, and performs other server-side tasks to create customized responses for each user request.
- Importance in Web Application Development:
    - Dynamic Content Generation: Server-side scripting allows for the creation of dynamic web pages that can display different content based on user interactions, preferences, or other conditions.
    - Data Processing: It enables the processing of form submissions, authentication, and other backend operations that require interaction with databases or external APIs.
    - Security: Sensitive operations, such as user authentication and data validation, can be securely handled on the server side, reducing the risk of client-side manipulation.

Question 2b

State the main features of each of the following server-side scripting languages: Java, Perl, PHP, ASP, and ColdFusion. For each language, provide an example of a web application that could be developed using it.

1. Java:
    - Features: Object-oriented, platform-independent, robust, secure, and widely used for enterprise-level applications.
    - Example Application: An e-commerce platform with a complex backend for managing products, orders, and user accounts.
2. Perl:
    - Features: Text manipulation, regular expressions, flexible and powerful

for scripting and automation tasks.
    - Example Application: A web-based tool for text processing and analysis, such as a log file analyzer.
3. PHP:
    - Features: Easy to learn, widely supported by web hosts, extensive library support, and good integration with databases.
    - Example Application: A content management system (CMS) like WordPress for managing website content.
4. ASP (Active Server Pages):
    - Features: Integrates well with Microsoft technologies, supports multiple languages (VBScript, JScript), and designed for building dynamic web applications.
    - Example Application: An intranet portal for employee management and communication within a company.
5. ColdFusion:
    - Features: Rapid development, strong database integration, easy to learn, and built-in functionalities for web development.
    - Example Application: An online event management system for scheduling and managing events and registrations.

Question 2c

Describe how CGI (Common Gateway Interface) works and its role in server-side scripting. Describe the steps involved in a typical CGI request/response cycle.

- Role in Server-Side Scripting: CGI enables web servers to execute external programs, typically written in languages like Perl, Python, or C, to generate dynamic content. It acts as a bridge between the web server and external applications.

Typical CGI Request/Response Cycle:

1. Client Request: The client (web browser) sends an HTTP request to the web server.
2. Server Processing: The web server identifies the request as a CGI request and invokes the appropriate CGI script or program.
3. Script Execution: The CGI script is executed, processing the request, often

involving database operations or other computations.

4. Output Generation: The script generates an HTTP response, typically in the form of HTML, which is sent back to the web server.
5. Server Response: The web server sends the generated HTML content back to the client, which is then rendered by the web browser.

Question 2d

Write ONLY the MISSING PHP codes (line 2, 3, 4, 5, 12, 13, 22, 31, 32, 34) for successful testing of the authentication, encryption, and connection to the database.

```php
<?php
// Set up database connection
$db_host = 'localhost';
$db_user = 'root';
$db_pass = 'password';
$db_name = 'database';

$conn = mysqli_connect($db_host, $db_user, $db_pass, $db_name);

// Start session
session_start();

// Check if form has been submitted
if(isset($_POST['submit'])) {

    // Sanitize and store user input
    $email = mysqli_real_escape_string($conn, $_POST['email']);
    $password = mysqli_real_escape_string($conn, $_POST['password']);

    // Query database for user
    $query = "SELECT * FROM users WHERE email='$email'";
    $result = mysqli_query($conn, $query);

    // Check if user exists
    if(mysqli_num_rows($result) == 1) {
        // Get user's data
        $row = mysqli_fetch_assoc($result);
        $db_password = $row['password'];

        // Verify password
        if(password_verify($password, $db_password)) {
            // Set session variable
            $_SESSION['user_id'] = $row['id'];
            // Redirect to dashboard
            header("Location: dashboard.php");
            exit();
        } else {
            // Invalid password
            echo "Invalid password";
```

```
        }
    } else {
        // User does not exist
        echo "User does not exist.";
    }
}
?>

<!DOCTYPE html>
<html>
<head>
    <title>Login</title>
</head>
<body>
<h1>Login</h1>
<form method="post" action="<?php echo $_SERVER['PHP_SELF']; ?>">
    <label>Email:</label>
    <input type="email" name="email" required><br>
    <label>Password:</label>
    <input type="password" name="password" required><br>
    <button type="submit" name="submit">Login</button>
</form>
</body>
</html>
```

This completes the missing PHP codes for the provided script, ensuring the connection to the database, user input sanitization, password encryption and verification, and session management.

Question 3

Question 3a

What is an IPv4 address and why is it important for the attendance system of this company?

- **IPv4 Address**: An IPv4 (Internet Protocol version 4) address is a numerical label assigned to each device connected to a computer network that uses the Internet Protocol for communication. It consists of four numbers separated by periods (e.g., 192.168.1.1), where each number can range from 0 to 255.
- **Importance for Attendance System**: The IPv4 address is important for the attendance system because it can be used to identify and track the devices from which employees log into the application. This helps ensure that attendance records are accurately linked to specific employees and can help detect unauthorized access or anomalies.

Question 3b

How can you retrieve the IPv4 address of a user who logs into the web application using PHP?

- **Retrieve IPv4 Address**:

```php
$user_ip = $_SERVER['REMOTE_ADDR'];
echo "User's IP Address: " . $user_ip;
```

Question 3c

Explain how cookies work and why they are necessary for this attendance system.

- **How Cookies Work**: Cookies are small pieces of data stored on the user's device by the web browser. They are created by the server and sent to the client, where they are stored and then sent back to the server with each subsequent request. Cookies can store user preferences, session information, and other data.
- **Necessity for Attendance System**:
    - **Session Management**: Cookies can be used to remember the login session of employees, ensuring they do not have to log in repeatedly during a single session.
    - **Personalization**: Cookies can store user-specific data, such as preferences or settings, to provide a customized user experience.
    - **Tracking**: Cookies help in tracking user activity and interactions within the application, which is essential for accurately recording attendance.

Question 3d

Write a PHP function that sets a cookie to remember a user's login session.

```php
<?php
function setLoginSession($user_id) {
    $cookie_name = "user_session";
    $cookie_value = $user_id;
    $expiry_time = time() + (86400 * 30); // Cookie valid for 30 days
    setcookie($cookie_name, $cookie_value, $expiry_time, "/"); // "/" means cookie is avai
    echo "Cookie set successfully!";
}
?>
```

Question 3e

Write a SQL query that inserts attendance data into a table named "attendance" with columns "employee_id", "date", and "status".

```sql
INSERT INTO attendance (employee_id, date, status) VALUES ('EMP12345', '2024-07-02', 'Pres
```

Question 3f

Explain what AJAX is and how it can be used to display attendance data in real-time to the HR department.

- **AJAX (Asynchronous JavaScript and XML)**: AJAX is a technique for creating fast and dynamic web pages. It allows web applications to send and retrieve data from a server asynchronously without interfering with the display and behavior of the existing page. This means that parts of a web page can be updated without reloading the whole page.
- **Use in Attendance System**:
    - **Real-Time Data Update**: AJAX can be used to fetch the latest attendance data from the server and update the HR department's dashboard in real-time. This ensures that HR has access to the most current data without needing to refresh the page.

- Example Implementation:

```
// JavaScript code to use AJAX for fetching attendance data
function fetchAttendanceData() {
    var xhr = new XMLHttpRequest();
    xhr.open("GET", "fetch_attendance.php", true);
    xhr.onreadystatechange = function() {
        if (xhr.readyState == 4 && xhr.status == 200) {
            document.getElementById("attendanceTable").innerHTML = xh
        }
    };
    xhr.send();
}

// Call fetchAttendanceData function periodically, e.g., every minute
setInterval(fetchAttendanceData, 60000);
```

By using AJAX, the web application can provide a seamless and efficient user experience, ensuring that HR staff have up-to-date attendance information at all times.

Question 4

Question 4a

Explain the importance of web application security and identify some common security risks that web applications face.

- Importance of Web Application Security:
    - Protects Sensitive Data: Ensures that sensitive information such as personal details, payment information, and business data are kept safe from unauthorized access and breaches.
    - Maintains Trust: Security is crucial for maintaining user trust and

confidence in the application, which is vital for customer retention and reputation.
    - Compliance: Helps in complying with legal and regulatory requirements like GDPR, HIPAA, etc., which mandate data protection measures.
    - Prevents Financial Loss: Protects against financial losses due to fraud, data breaches, and cyber-attacks.
    - Business Continuity: Ensures that web applications remain available and functional, preventing downtime caused by security incidents.
- Common Security Risks:
    - SQL Injection: Attackers can manipulate a web application's database query by injecting malicious SQL code.
    - Cross-Site Scripting (XSS): Attackers can inject malicious scripts into web pages viewed by other users, potentially stealing information or spreading malware.
    - Cross-Site Request Forgery (CSRF): Attackers can trick users into performing actions they did not intend by exploiting their authenticated sessions.
    - Insecure Direct Object References (IDOR): Attackers can access unauthorized data by manipulating URLs or form parameters to reference objects directly.
    - Security Misconfiguration: Improper configuration of web servers, databases, and applications can lead to vulnerabilities and exploits.

Question 4b

How can jQuery be used to enhance the security of a web application? Provide examples.

- Input Validation: jQuery can be used to validate user inputs on the client side before they are sent to the server, reducing the risk of malformed or malicious data.
    - Example: Checking if an email input follows the correct format.

```
$(document).ready(function(){
    $("#submitForm").click(function(){
        var email = $("#email").val();
        var emailPattern = /^[a-zA-Z0-9._-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]
        if(!emailPattern.test(email)) {
            alert("Please enter a valid email address.");
            return false;
        }
    });
});
```

- **Preventing CSRF**: jQuery can be used to include CSRF tokens in AJAX requests, ensuring that requests are genuine.
    - **Example**: Including a CSRF token in an AJAX request.

```
$.ajaxSetup({
    headers: {
        'X-CSRF-Token': $('meta[name="csrf-token"]').attr('content')
    }
});

$("#submitForm").click(function(){
    $.post("/submit", { data: "example" }, function(response){
        console.log(response);
    });
});
```

- **Securing AJAX Requests**: jQuery can help in making secure AJAX calls, ensuring that data transmission is protected.
    - **Example**: Using HTTPS for secure AJAX requests.

```
$.ajax({
    url: "https://example.com/api",
    type: "POST",
    data: { name: "John" },
    success: function(response) {
        console.log(response);
    },
    error: function(error) {
        console.log(error);
    }
});
```

- **Dynamic Content Loading**: jQuery can load content dynamically without exposing sensitive data in the URL.
    - **Example**: Loading user-specific data securely.

```
$("#loadUserData").click(function(){
    $.get("/user/data", function(data){
        $("#userData").html(data);
    });
});
```

Question 4c

Write the HTML5 codes for query.html (as shown in Figure 1) taking into consideration the additional information provided. You may assume that NO table tag has been used in this form.

```
<!DOCTYPE html>
<html>
<head>
    <title>Client Details E-Commerce</title>
</head>
<body>
    <h1>Client Details E-Commerce</h1>
    <h2>JUNE/JULY 2023</h2>
    <form action="myclient.php" method="POST">
        <p>
            <label for="detcontact">Contact Date:</label>
            <input type="date" id="detcontact" name="detcontact">
        </p>
        <p>
            <label for="txtCID">Client Id:</label>
            <input type="text" id="txtCID" name="txtCID">
        </p>
        <p>
            <label for="txtFname">First Name:</label>
            <input type="text" id="txtFname" name="txtFname">
        </p>
        <p>
            <label for="txtLname">Last Name:</label>
            <input type="text" id="txtLname" name="txtLname">
        </p>
        <p>
            <label for="rdoGender">Gender:</label>
            <input type="radio" id="rdoGender" name="rdoGender" value="M"> Male
            <input type="radio" id="rdoGender" name="rdoGender" value="F"> Female
        </p>
        <p>
            <label for="slPreferences">Preferences:</label>
            <select id="slPreferences" name="slPreferences">
                <option value="Books">Books</option>
                <option value="Car">Car</option>
                <option value="Clothing">Clothing</option>
            </select>
            <label for="txtOther">Other Particulars:</label>
            <input type="text" id="txtOther" name="txtOther">
```

```
        </p>
        <p>
            <label for="chkOfficer">Administrator:</label>
            <input type="checkbox" id="chkOfficer" name="chkOfficer"> Above Client's info
        </p>
        <p>
            <button type="submit">Submit Form</button>
            <button type="reset">Reset Form</button>
        </p>
    </form>
</body>
</html>
```

This HTML code includes all the specified elements with the appropriate IDs and names, and it ensures that the form data is submitted to `myclient.php` via the POST method.