IP II

UE 2022 SOLVED

i. What server support Ajax?

a) WWW

b) SMTP

c) HTTP

d) POP3

The correct answer is c) HTTP. Ajax stands for Asynchronous JavaScript and XML, and it's a technique for creating fast and dynamic web pages. Ajax works by sending and receiving data asynchronously with the server using the XMLHttpRequest object in JavaScript. HTTP is the protocol used for communicating between the client and server in Ajax applications.

ii. What is the correct syntax of the declaration which defines the XML version?

a. <xml version=”A.0”/>

b. <?xml version="A.0"?>

c. <?xml version="A.0"/>

d. <xml version =yes>

The correct answer is b. <?xml version="A.0"?>. This declaration is required at the beginning of an XML document and specifies the version of XML in use. The correct syntax is to use the question mark <? at the beginning of the declaration, followed by the keyword "xml," a space, the attribute "version" with its value in quotes, and finally, the question mark and closing > symbol.

iii. Which formats are supported by Ajax to send and receive information?

a) JSON, XML, HTML, SQL, MySQL, and Text Files

b) JSON, XML, HTML, and Text Files

c) JSON, XML, HTML, NoSQL, and Oracle

d) JAVA, JSON, XML, HTML, Oracle, and Text Files

The correct answer is B. JSON, XML, HTML, and Text Files are the formats supported by Ajax to send and receive information. Ajax stands for Asynchronous JavaScript and XML, which means it uses JavaScript and XML to send and receive

data asynchronously with the server. However, it's not limited to XML, and JSON and HTML are also popular formats for sending and receiving data using Ajax. Text files can also be used for simple data exchange. Other formats mentioned in the options, such as SQL, MySQL, NoSQL, Oracle, and JAVA are not formats supported by Ajax for data exchange.

iv. Which statement IS true

A. All XML elements must be in lower case

B. All XML elements must have a closing tag

C. All XML documents must have a DTD

D. All the statements are true

The correct answer is B. All XML elements must have a closing tag. While XML elements can be written in lowercase, it's not a requirement. XML elements are case-sensitive and can be written in any case. All XML documents don't have to have a DTD (Document Type Definition) either. A DTD is optional, and its purpose is to define the structure of an XML document, including the allowed elements and their attributes.

v. Which Of the following is used to display the output in PHP?

A. Echo

B. Both (a) and (d)

C. write

D. Print

The correct answer is A. Echo is used to display the output in PHP. The echo statement is one of the most commonly used functions in PHP and is used to output strings and variables to the browser or the command line.

vi. Which of the following role of web service architecture
provides a central place where developers can publish new
services or find existing ones?

A. Service Provider

B. Service Registry

C.Service Requestor

D.Service Consumer

The correct answer is B. Service Registry provides a central place where developers can publish new services or find existing ones. In a web service architecture, there are four main roles: Service Provider, Service Requestor, Service Registry, and Service Consumer. The Service Registry acts as a directory of available services and provides a way for Service Consumers to find them.

vii. If there is no error, then what will the error () method return?

A. True

B. Empty String

C. 0

D. False

The correct answer is B. If there is no error, the error() method will return an empty string. The error() method is used to retrieve the error message from the last operation on an XML parser. If there is no error, the method returns an empty string.

viii. Which is the correct symbol to insert a comment in JSON?

A. //

B. /*…*/

C. JSON doesn't support comments

D. <!--…-->

The correct answer is C. JSON doesn't support comments. Unlike JavaScript, which supports comments, JSON doesn't support comments. Any comments included in a JSON file will cause a parsing error.

ix. How do you represent a JSON array of strings?

A. { "days": { "SUN" , "MON" , "TUE" , "WED “,"THU" , "FRI" , "SAT" } }

B. { "days"= ["SUN" , "MON" , "TUE" , "WED”,"THU" , "FRI" , "SAT"] }

C. { "days" : ["SUN" , "MON" , "TUE" , "WED”,"THU" , "FRI" , "SAT'] }

D. { "days"= { "SUN" , "MON" , "TUE" , "WED”,"THU" , "FRI" , "SAT' } }

The correct answer is C. To represent a JSON array of strings, you would use the following syntax: { "days" : ["SUN" , "MON" , "TUE" , "WED","THU" , "FRI" , "SAT"] }. In this example, the "days" key is associated with an array of strings.

x. Which of the following function displays the information about PHP and its configuration?

A. php_info()

B. info()

C. phpinfo()

D. phpinfor()

The correct answer is C. The phpinfo() function displays the information about PHP and its configuration. The phpinfo() function outputs a large amount of information about the current state of PHP, including information about PHP compilation options and extensions, the PHP version, server information, and more.

2. Match the item in Column A with its corresponding item in Column B.

Column A	Column B.
i . onreadystatechange	A. request received
ii. Http Status =400	B. XML
iii. readyState =2	C. Server not found
iv. REST	D. Provider of a web service on an online directory
v. Web Services Definition Language	E. XML document depends on other external files
vi. One among the principles of REST architecture	F. Internal server error
vii. XML stand alone =YES	G. Fixed width
viii. Styling language for xml	H. Processing request
ix..container-fluid	I. Set with an JavaScript event handler that fires at each state change
x. Helper classes	J. Uniform Interface
	K. XML/JSON/TEXT
	L. XML based interface for describing the functionalities of the web service
	M. XML document does not depending on other external files
	N. XSLT
	O. Full width
	P. Make customization as quickly as possible.

Answer

i. onreadystatechange - I. Set with a JavaScript event handler that fires at each state change

ii. Http Status =400 - F. Internal server error

iii. readyState =2 - A. request received

iv. REST - J. Uniform Interface

v. Web Services Definition Language - L. XML based interface for describing the functionalities of the web service

vi. One among the principles of REST architecture - P. Make customization as quickly as possible.

vii. XML stand alone =YES - M. XML document does not depend on other external files

viii. Styling language for xml - N. XSLT

ix. container-fluid - O. Full width

x. Helper classes - E. XML document depends on other external files

3. Answer the following questions:-

a) HTTP is a stateless protocol. Explain

Answer

a) HTTP is a stateless protocol because it does not maintain any information about the state of the communication between client and server. Each request from the client to the server is treated as a new request, and the server does not remember any information about previous requests. Therefore, the server does not have any knowledge of the client's previous requests or actions.

b) Explain the two ways to make HIT P a state full protocol.

Answer

b) There are two ways to make HTTP a stateful protocol:

- ✧ **Cookies:** The client can send a cookie to the server, and the server can use the information in the cookie to maintain the state of the communication. The cookie can include information such as session ID, user preferences, and other information that the server can use to identify the client and remember their actions.

- ✧ **Session variables:** The server can use session variables to store information about the client's actions and use it to maintain the state of the communication. Session variables are stored on the server and are associated with a specific client. The server uses a session ID to identify the client and retrieve their session variables.

c) Explain the differences between Simple Object Access Protocol (SOAP) and Representational State Transfer (REST) web services.

Answer

c) SOAP and REST are two different web service architectures.

SOAP (Simple Object Access Protocol) is a messaging protocol that uses XML to exchange data between two endpoints. SOAP uses a predefined set of rules for defining the structure of messages and for processing the messages at the receiving end. SOAP is often used for enterprise-level applications that require more complex messaging and security features.

REST (Representational State Transfer) is an architectural style for designing web services. REST uses HTTP methods (GET, POST, PUT, DELETE) to perform CRUD (Create, Read, Update, Delete) operations on resources. REST does not use a predefined structure for messages, but instead, uses a uniform interface to access and manipulate resources.
REST is more lightweight than SOAP and is often used for simpler web services and mobile applications.

d)PHP is "a weakly typed language". Explain what this statement means as regards to php variables.
Answer
d) PHP is a weakly typed language, which means that PHP does not require the programmer to declare the data type of a variable before using it. In PHP, variables are dynamically typed, which means that the data type of a variable is determined at runtime based on the value assigned to it. This allows for more flexible programming, but it can also lead to errors if the programmer is not careful with variable types. For example, a variable that was assigned a string value can later be assigned a numeric value without the need for explicit type conversion.

4. You have been hired as a front end developer at one of the E- Commerce company. The first day of your work at the company you realize that company's website is not okay. You decided to use bootstrap framework to revamp the website to make it more attractive and user friendly. Among the features of the bootstrap you have decided to use are carousel and pagination. You have decided to divide the contents of the website into five pages ie. pagel .html, page2 .html, page3 .html, page4 .html and page5 .html. Among the images that describe the company are image l .jpg, image2.jpg, image3.jpg, image4.jpg and image5.jpg. You thought it will be better if the images will be in a slide show. In your design show how you will use bootstrap carousel and pagination to create a slide show of the mentioned images and pagination of the mentioned pages respectively.

Hint: use carousel-indicator class to show the captions of each slide, carousel-inner class to add slides to the carousel, carousel-control-prev and carousel-control-next class to allow the user to go back and forward between the slides. (15 Marks)

Answer

Sure, here's an example code snippet that demonstrates how to use Bootstrap's carousel and pagination features to create a slide show of the mentioned images and pagination of the mentioned pages respectively:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Bootstrap Carousel and Pagination Example</title>
  <!-- Link to Bootstrap CSS file -->
  <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
</head>
<body>
  <!-- Navigation Bar -->
  <nav class="navbar navbar-expand-sm bg-dark navbar-dark">
    <ul class="navbar-nav">
      <li class="nav-item active">
        <a class="nav-link" href="page1.html">Page 1</a>
      </li>
```

```
    <li class="nav-item">
      <a class="nav-link" href="page2.html">Page 2</a>
    </li>
    <li class="nav-item">
      <a class="nav-link" href="page3.html">Page 3</a>
    </li>
    <li class="nav-item">
      <a class="nav-link" href="page4.html">Page 4</a>
    </li>
    <li class="nav-item">
      <a class="nav-link" href="page5.html">Page 5</a>
    </li>
  </ul>
</nav>

<!-- Carousel -->
<div id="myCarousel" class="carousel slide" data-ride="carousel">
  <!-- Indicators -->
  <ol class="carousel-indicators">
    <li data-target="#myCarousel" data-slide-to="0" class="active"></li>
    <li data-target="#myCarousel" data-slide-to="1"></li>
    <li data-target="#myCarousel" data-slide-to="2"></li>
    <li data-target="#myCarousel" data-slide-to="3"></li>
    <li data-target="#myCarousel" data-slide-to="4"></li>
  </ol>

  <!-- Slides -->
  <div class="carousel-inner">
    <div class="carousel-item active">
      <img src="image1.jpg" alt="Image 1" style="width:100%;">
      <div class="carousel-caption">
        <h3>Image 1 Caption</h3>
      </div>
    </div>
```

```
    <div class="carousel-item">
      <img src="image2.jpg" alt="Image 2" style="width:100%;">
      <div class="carousel-caption">
        <h3>Image 2 Caption</h3>
      </div>
    </div>
    <div class="carousel-item">
      <img src="image3.jpg" alt="Image 3" style="width:100%;">
      <div class="carousel-caption">
        <h3>Image 3 Caption</h3>
      </div>
    </div>
    <div class="carousel-item">
      <img src="image4.jpg" alt="Image 4" style="width:100%;">
      <div class="carousel-caption">
        <h3>Image 4 Caption</h3>
      </div>
    </div>
    <div class="carousel-item">
      <img src="image5.jpg" alt="Image 5" style="width:100%;">
      <div class="carousel-caption">
        <h3>Image 5 Caption</h3>
      </div>
    </div>
  </div>

  <!-- Controls -->
  <a class="carousel-control-prev" href="#myCarousel" data-slide="prev">
<!-- Page Navigation -->
<nav aria-label="Page navigation example">
 <ul class="pagination justify-content-center">
  <li class="page-item"><a class="page-link" href="pagel.html">1</a></li>
  <li class="page-item"><a class="page-link" href="page2.html">2</a></li>
  <li class="page-item"><a class="page-link" href="page3.html">3</a></li>
```

```
    <li class="page-item"><a class="page-link" href="page4.html">4</a></li>
    <li class="page-item"><a class="page-link" href="page5.html">5</a></li>
  </ul>
</nav>

<!-- Image Carousel -->
<div id="myCarousel" class="carousel slide" data-ride="carousel">
  <ol class="carousel-indicators">
    <li data-target="#myCarousel" data-slide-to="0" class="active"></li>
    <li data-target="#myCarousel" data-slide-to="1"></li>
    <li data-target="#myCarousel" data-slide-to="2"></li>
    <li data-target="#myCarousel" data-slide-to="3"></li>
    <li data-target="#myCarousel" data-slide-to="4"></li>
  </ol>
  <div class="carousel-inner">
    <div class="carousel-item active">
      <img class="d-block w-100" src="image1.jpg" alt="First slide">
      <div class="carousel-caption d-none d-md-block">
        <h5>Image 1 Caption</h5>
      </div>
    </div>
    <div class="carousel-item">
      <img class="d-block w-100" src="image2.jpg" alt="Second slide">
      <div class="carousel-caption d-none d-md-block">
        <h5>Image 2 Caption</h5>
      </div>
    </div>
    <div class="carousel-item">
      <img class="d-block w-100" src="image3.jpg" alt="Third slide">
      <div class="carousel-caption d-none d-md-block">
        <h5>Image 3 Caption</h5>
      </div>
    </div>
    <div class="carousel-item">
```

```
    <img class="d-block w-100" src="image4.jpg" alt="Fourth slide">
    <div class="carousel-caption d-none d-md-block">
      <h5>Image 4 Caption</h5>
    </div>
  </div>
  <div class="carousel-item">
    <img class="d-block w-100" src="image5.jpg" alt="Fifth slide">
    <div class="carousel-caption d-none d-md-block">
      <h5>Image 5 Caption</h5>
    </div>
  </div>
 </div>
 <a class="carousel-control-prev" href="#myCarousel" data-slide="prev">
  <span class="carousel-control-prev-icon"></span>
  <span class="sr-only">Previous</span>
 </a>
 <a class="carousel-control-next" href="#myCarousel" data-slide="next">
  <span class="carousel-control-next-icon"></span>
  <span class="sr-only">Next</span>
 </a>
</div>
```

5. Answer the following questions: -

a. Create an Ajax-based product catalog that obtains its data from JSON files located on the server. The data should be separated into four JSON files. The first file should be a summary file, containing a list of products. Each product should have a title, an image file name for a thumbnail image and a price. The second file should contain a list of descriptions for each product. The third file should contain a list of file names for the full-size product images. The last file should contain a list of the thumbnail image file names. Each item in a catalogue should have a unique ID that should be included with the entries for that product in every file. Next, create an Ajax-enabled web page that displays the product information in a table. The catalog should initially display a list of product names with their associated thumbnail images and prices. When the. mouse hovers over a thumbnail image, the larger product image should be displayed. When the user moves the mouse away from that image, the original thumbnail should be redisplayed. You should provide a button that the user can click to display the product description.

Answer

To create an Ajax-based product catalog that obtains its data from JSON files located on the server, we will need to create four JSON files - one for summary data, one for product descriptions, one for full-size product images, and one for thumbnail images. We will then create an HTML page that uses jQuery and Ajax to load and display the data.

Here's an example of how we can structure the four JSON files:

summary.json:

```
{
  "products": [
    {
      "id": "1",
      "title": "Product 1",
      "thumbnail": "product1-thumb.jpg",
      "price": "10.00"
    },
    {
      "id": "2",
      "title": "Product 2",
```

```
      "thumbnail": "product2-thumb.jpg",
      "price": "20.00"
    },
    {
      "id": "3",
      "title": "Product 3",
      "thumbnail": "product3-thumb.jpg",
      "price": "30.00"
    }
  ]
}
```


descriptions.json:


```
{
  "products": [
    {
      "id": "1",
      "description": "Product 1 description"
    },
    {
      "id": "2",
      "description": "Product 2 description"
    },
    {
      "id": "3",
      "description": "Product 3 description"
    }
  ]
}
```


full_images.json:


```
{
  "products": [
    {
      "id": "1",
      "image": "product1-full.jpg"
```

```
    },
    {
      "id": "2",
      "image": "product2-full.jpg"
    },
    {
      "id": "3",
      "image": "product3-full.jpg"
    }
  ]
}
```

thumbnail_images.json:

```
{
  "products": [
    {
      "id": "1",
      "thumbnail": "product1-thumb.jpg"
    },
    {
      "id": "2",
      "thumbnail": "product2-thumb.jpg"
    },
    {
      "id": "3",
      "thumbnail": "product3-thumb.jpg"
    }
  ]
}
```

Now, let's create the HTML page to display the product information in a table:

```
<!DOCTYPE html>
<html>
<head>
  <title>Product Catalog</title>
  <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
```

```
  <script>
    $(document).ready(function() {
      // Load summary data from JSON file
      $.getJSON("summary.json", function(data) {
        $.each(data.products, function(key, value) {
          // Create table row for each product
          var row = "<tr id='" + value.id + "'>";
          row += "<td><img src='" + value.thumbnail + "' alt='" + value.title +
"'></td>";
          row += "<td>" + value.title + "</td>";
          row += "<td>$" + value.price + "</td>";
          row += "</tr>";
          $("table").append(row);
        });

        // Load full-size product images from JSON file
        $.getJSON("full_images.json", function(data) {
          $.each(data.products, function(key, value) {
            // Add full-size image as a hidden element
            $("#" + value.id).append("<input type='hidden' value='" + value.image +
"'>");
          });
        });

        // Load product descriptions from JSON file
        $.getJSON("descriptions.json", function(data) {
          $.each(data.products, function(key, value) {
            // Add description as
```

Huyu chatgpt kaishia hapa ss cjui ndo mwisho au inaendelea..

b. Write a PHP script that obtains a URL and its description from a user through html form and stores the information into a database using MySQL. Use URL and Url table as the name of the database and table respectively. The first field of the table should contain an actual URL, and the second, which is named Description, should contain a description of the URL. Use www.udom.ac.tz as the first URL, and input UDOM as its description. The second URL should be www.ratiba.udom.ac.tz, and the description should be UDOM Time Table System. The third URL should be www.civeclassroom.udom.ac.tz, and its description should be CIVE digital classroom. The fourth URL should be www.civeipt.udom.ac.tz, and its description should be CIVE Industrial Practical Training. After each new URL is submitted, print the contents of the database in a table format.
Use server side validation to make sure that the fields for URL and Description should accept empty values and should not contain spaces before and at the end.
Answer

```
<?php
// connect to the database
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "URL";

$conn = new mysqli($servername, $username, $password, $dbname);

// check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

// initialize variables
$url = "";
$description = "";
$errors = array();

// validate form data on submission
```

```
if (isset($_POST['submit'])) {
    // sanitize user input
    $url = trim($_POST['url']);
    $description = trim($_POST['description']);

    // check for empty URL
    if (empty($url)) {
        array_push($errors, "URL is required");
    } else {
        // check if URL already exists in the database
        $result = mysqli_query($conn, "SELECT * FROM Url WHERE url='$url'");
        if (mysqli_num_rows($result) > 0) {
            array_push($errors, "URL already exists");
        }
        // check for spaces at the beginning and end of the URL
        if (preg_match('/^\s+|\s+$/', $url)) {
            array_push($errors, "URL should not contain spaces at the beginning or end");
        }
    }

    // check for empty description
    if (empty($description)) {
        array_push($errors, "Description is required");
    } else {
        // check for spaces at the beginning and end of the description
        if (preg_match('/^\s+|\s+$/', $description)) {
            array_push($errors, "Description should not contain spaces at the beginning or
end");
        }
    }

    // if there are no errors, insert the data into the database
    if (count($errors) == 0) {
        $sql = "INSERT INTO Url (url, description) VALUES ('$url', '$description')";
```

```
        mysqli_query($conn, $sql);

        // print contents of the database in a table format
        $result = mysqli_query($conn, "SELECT * FROM Url");
        echo "<table>";
        echo "<tr><th>URL</th><th>Description</th></tr>";
        while ($row = mysqli_fetch_array($result)) {
            echo "<tr><td>" . $row['url'] . "</td><td>" . $row['description'] . "</td></tr>";
        }
        echo "</table>";
    }
}
// close the database connection
mysqli_close($conn);
?>

<!-- HTML form to accept user input -->
<form method="post" action="<?php echo
htmlspecialchars($_SERVER["PHP_SELF"]);?>">
    <label for="url">URL:</label>
    <input type="text" id="url" name="url" value="<?php echo $url; ?>">
    <br><br>
    <label for="description">Description:</label>
    <input type="text" id="description" name="description" value="<?php echo
$description; ?>">
    <br><br>
    <input type="submit" name="submit" value="Submit">
</form>

<!-- display validation errors, if any -->
<?php if (count($errors) > 0): ?>
    <div>
        <?php foreach ($errors as $error): ?>
            <p><?php echo $error; ?></p>
```

```
        <?php endforeach; ?>
    </div>
```

6. Answer the following questions: -

a. XML plays an important role in the current world of web development. It is completely beneficial for those who want to use web technology to distribute information across the web.

i. Create a web service in xml format which contain information of the students such as: Students firstname, last name, RegNo, Age, Gender, Courses, and Program.

ii. Using PHP write a script.that will read the XML web service in part (i) and display the results in a web browser in JSON format.

Answer

i. Here's an example XML web service that contains information about students:

```
<?xml version="1.0"?>
<students>
  <student>
    <firstname>John</firstname>
    <lastname>Doe</lastname>
    <regno>12345</regno>
    <age>20</age>
    <gender>Male</gender>
    <courses>
      <course>Mathematics</course>
      <course>Computer Science</course>
    </courses>
    <program>Bachelor of Science</program>
  </student>
  <student>
    <firstname>Jane</firstname>
    <lastname>Smith</lastname>
    <regno>54321</regno>
    <age>19</age>
    <gender>Female</gender>
    <courses>
      <course>English</course>
      <course>History</course>
    </courses>
```

```
        <program>Bachelor of Arts</program>
    </student>
</students>
```

ii. Here's a PHP script that will read the XML web service in part (i) and display the results in a web browser in JSON format:

Answer

```
<?php
// Load the XML file
$xml = simplexml_load_file('students.xml');

// Initialize an empty array to store the student information
$students = array();

// Loop through each student and store their information in the array
foreach ($xml->student as $student) {
    $student_info = array(
        'firstname' => (string)$student->firstname,
        'lastname' => (string)$student->lastname,
        'regno' => (string)$student->regno,
        'age' => (int)$student->age,
        'gender' => (string)$student->gender,
        'courses' => array(),
        'program' => (string)$student->program
    );
    foreach ($student->courses->course as $course) {
        $student_info['courses'][] = (string)$course;
    }
    $students[] = $student_info;
}

// Convert the array to JSON format and output it
echo json_encode($students);
?>
```

When you run this script in a web browser, it will display the student information in JSON format, like this:


```
[
    {
        "firstname": "John",
        "lastname": "Doe",
        "regno": "12345",
        "age": 20,
        "gender": "Male",
        "courses": [
            "Mathematics",
            "Computer Science"
        ],
        "program": "Bachelor of Science"
    },
    {
        "firstname": "Jane",
        "lastname": "Smith",
        "regno": "54321",
        "age": 19,
        "gender": "Female",
        "courses": [
            "English",
            "History"
        ],
        "program": "Bachelor of Arts"
    }
]
```

b. Suppose that there is a web service named adventure.php, located on your web server in the same directory as your code. This service outputs XML data describing a situation and options a player can pick (you can choose your own adventure style game). In this problem you will use Ajax to contact the web service (using a GET request), examine its XML data, and display the questions and options in the game. The XML data returned by the web service consists of an overall document element called adventure that contains a situation tag, an answers tag and an end tag. The answers tag contains a series of one or more answer elements inside
it. The format of the XML matches the following.

```
<adventure>
<situation>The dragon is looking at you</situation>
<answers>
<answer>run screaming</answer>
<answer>draw your sword</answer>
<answer>be very polite</answer>
<answer>sneeze</answer>
< / answers>
<end reason="ooh! The dragon accidentally stepped on you" / >
< / adventure>
```

Assume the code you are writing will go into adventure.js, linked from the following script.

```
<body>
<h1>Choose your own adventure</hl>
<div id="container">
< fieldset id=”situation”> </fieldset>
< p id="situationparagraph"></p> </fieldset>
< fieldset id="answer"></fieldset>
</div>
</body>
```

When the page loads send an AJAX request to the adventure.php service and pass it a parameter named situation with the value begin. Insert the questions you get
back into the situation paragraph. Add a button to the answer fieldset for each answer tag you receive. Set these buttons so that if they are clicked your page will send an Ajax request to adventure.php with the situation parameter set to the text on the

button. This request will return the same sort of data which should be dealt with in the same way. The buttons displayed should only ever be the current options, no 01 options should show. If instead of receiving XML back that contains a question tag you receive XML with an end tag, place the text of the reason attribute of the end tag in the situation paragraph and make sure no buttons are showing.

Answer

Here's an example implementation of the adventure game using Ajax:

```
<body>
  <h1>Choose Your Own Adventure</h1>
  <div id="container">
    <fieldset id="situation">
      <legend>Current Situation:</legend>
      <p id="situation-paragraph"></p>
    </fieldset>
    <fieldset id="answers">
      <legend>Choose Your Next Move:</legend>
    </fieldset>
  </div>
  <script src="adventure.js"></script>
</body>
```

JavaScript (adventure.js):

```
// Function to send an Ajax request to the adventure.php web service
function sendRequest(situation) {
 // Create a new XMLHttpRequest object
 const xhr = new XMLHttpRequest();

 // Set up the request
 xhr.open('GET', `adventure.php?situation=${situation}`);

 // Set up the response handler
 xhr.onload = function() {
  if (xhr.status === 200) {
   // Parse the XML response
   const responseXML = xhr.responseXML;
```

```
    const situationText = responseXML.querySelector('situation').textContent;
    const endReason = responseXML.querySelector('end')?.getAttribute('reason');
    const answerElements = responseXML.querySelectorAll('answer');

    // Display the situation text
    const situationParagraph = document.getElementById('situation-paragraph');
    situationParagraph.textContent = situationText;

    // Remove any previous answer buttons
    const answersFieldset = document.getElementById('answers');
    answersFieldset.innerHTML = '';

    if (endReason) {
      // If the response includes an end tag, display the end reason and stop the game
      situationParagraph.textContent = endReason;
    } else {
      // If the response includes answer tags, display buttons for each answer
      answerElements.forEach(answerElement => {
        const answerButton = document.createElement('button');
        answerButton.textContent = answerElement.textContent;
        answerButton.addEventListener('click', () =>
sendRequest(answerElement.textContent));
        answersFieldset.appendChild(answerButton);
      });
    }
  }
 };

 // Send the request
 xhr.send();
}

// Start the game by sending an Ajax request with situation='begin'
sendRequest('begin');
```

The adventure game will start by displaying the initial situation text and a set of buttons for possible answers. When the player clicks on an answer button, an Ajax request is sent to the web service with the corresponding situation text as a parameter. The web service will respond with either another situation text and a new set of answer buttons, or an end reason indicating that the game is over. The game will continue until the web service returns an end reason.

7. The University of Dodoma through its College of Informatics and Virtual Education (CIVE) has decided to create a web system to manage the whole process of Industrial Practical Training (IPT). The system was developed using PHP and MySQL database. The system has two parts, which are source codes and database. The database part is hosted at the server with IP Address 192.168.10.10 and the source code is hosted at the server with IP Address 192.168.10.20. After the system has been deployed it was observed that the performance was poor as the system loads very slowly especially when the information are retrieved from the server. It was observed that the problem was due to huge amount of data that are loaded from the server. As an expert in web development, you have been consulted to give your advice. In your advice you suggested AJAX technology to be used to load data from the server to the client side so as to improve performance. Assume you want to load the following information from the table student (FName, SName, Program, Gender and RegNo). Write a script hosted at the source code server to show how you connect to the database, load data from the server using AJAX and display to the client in a table format. Use JQuery methods to send the AJAX requests.
Use the following credentials to access the database server. Udom_IPT as a database name, Password: 123 abc, username: udom cive.
Answer
To connect to the database and retrieve the required data using AJAX, the following steps can be followed:
Step 1: Create a PHP file on the source code server (192.168.10.20) named "get_data.php". This file will be responsible for connecting to the database, executing the query to retrieve the required data, and returning the result in JSON format.
get_data.php:

```
<?php
// Set database credentials
$db_host = "192.168.10.10";
$db_name = "Udom_IPT";
$db_user = "udomcive";
$db_pass = "123abc";

// Connect to the database
$conn = new mysqli($db_host, $db_user, $db_pass, $db_name);
```

```
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

// Query to retrieve student data
$sql = "SELECT FName, SName, Program, Gender, RegNo FROM student";
$result = $conn->query($sql);

// Store the result in an array
$data = array();
if ($result->num_rows > 0) {
    while ($row = $result->fetch_assoc()) {
        $data[] = $row;
    }
}

// Close the database connection
$conn->close();

// Return the result in JSON format
echo json_encode($data);
?>
```

Step 2: Create an HTML file on the source code server (192.168.10.20) named "index.html". This file will contain the table where the data will be displayed.

index.html:

```
<!DOCTYPE html>
<html>
<head>
    <title>Student Data</title>
    <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
    <script>
        $(document).ready(function() {
            // Send AJAX request to get data from server
            $.ajax({
```

```
            url: "get_data.php",
            dataType: "json",
            success: function(data) {
                // Create table header
                var header = "<tr><th>First Name</th><th>Last Name</th><th>Program</th><th>Gender</th><th>Registration No</th></tr>";
                $("#student_table thead").html(header);

                // Create table rows
                var rows = "";
                $.each(data, function(index, student) {
                    rows += "<tr>";
                    rows += "<td>" + student.FName + "</td>";
                    rows += "<td>" + student.SName + "</td>";
                    rows += "<td>" + student.Program + "</td>";
                    rows += "<td>" + student.Gender + "</td>";
                    rows += "<td>" + student.RegNo + "</td>";
                    rows += "</tr>";
                });
                $("#student_table tbody").html(rows);
            }
        });
    });
  </script>
</head>
<body>
  <h1>Student Data</h1>
  <table id="student_table">
    <thead></thead>
    <tbody></tbody>
  </table>
</body>
</html>
```

This script will use JQuery's AJAX method to send a request to the server and retrieve the required data from the database. Once the data is retrieved, it will be displayed in a table format on the client side. This approach will significantly reduce the amount of data that needs to be loaded from the server and improve the performance of the IPT system.