COMP3322 Modern Technologies on WWW

Online Quiz II

Date: April 21, 2020

Student Number: _____

Instructions

- **This is an open-note quiz. Candidates are permitted to bring to the quiz hardcopies of all lecture slides, on which handwritten notes are allowed.**
- **Candidates are not allowed to search the Internet for answers during the examination. However, they can use the built-in developer tools in the browsers to assist their coding.**
- **Total of 60 points. This quiz consists of 4 questions. Candidates are required to answer all questions.**
- **If you are not clear about what a question is asking, be sure to write down any assumptions you have made in answering the question.**
- **To answer the questions, you can**
    - **use a text editor to type in the answers and submit the file to Moodle within 6 hours – this is the preferred option.**
    - **type the answers in a Microsoft Word document and submit the file to Moodle within 6 hours.**
    - **use some PDF editing software, type the answers on the PDF, and submit the file to Moodle within 6 hours.**
    - **print the quiz papers, write the answers on the papers, scan them to a single PDF file, and submit the file to Moodle within 6 hours – this is the least preferred option.**
- **Please make sure that your submission includes all your answers and the answers are arranged in the question order. The quality of the images is acceptable and clear.**

The Honor Pledge (If you use options 1/2, please copy the Honor Pledge paragraph to your file.)

- **In taking this quiz, I understand that I may not work with anyone else, including conferring with others (student, or anyone else); exchanging information, answer or ideas; or in aiding or being aided by others in the completion of this assignment. I understand that failure to follow the rules is considered cheating and may result in initiating disciplinary actions. I certify that I have personally prepared the answers to this assessment in accordance with the above stated rules.**

Signature of the examinee: _____

Question 1 (10 points)

a) (4) Write a fragment of PHP code that uses **a cookie** named 'visit' to keep track of how many times a client has visited the web site on "www.code.co/mysite/" and its sub-directories since the browser started. This fragment of code will be **placed in all .php files** under this web site. For example, "www.code.co/mysite/", "www.code.co/mysite/about/", "www.code.co/mysite/contact/", etc. Please use pure cookie and don't use session to achieve that.

b) (1) Write a fragment of PHP code that removes the cookie installed by the above code fragment.

c) (4) Rewrite part (a) with the use of the session. The fragment of PHP code uses the session concept to keep track of how many times a client has visited the web site on "www.code.co/mysite/" and its sub-directories since the browser started. This fragment of code will be **placed in all .php files** under this web site.

d) (1) Write a fragment of PHP code that clears this visiting count (but not the session) once the code is executed.

Question 2 (8 points)

Below is a program that makes use of AJAX to retrieve a set of Response headers from the Web server. The set of headers is expected to be returned if the file “info.txt” would be requested with an HTTP GET method. The getAllResponseHeaders() method returns a string containing all response headers of the XMLHttpRequest object.

```
<!DOCTYPE html>
<html>
<body>

<h1>AJAX</h1>
<button type="button" onclick="loadDoc()">Load Content</button>
<p id="demo"></p>

<script>

function loadDoc() {
  var xhttp = new XMLHttpRequest();
  xhttp.onreadystatechange = function() {
    if (this.readyState == 4 && this.status == 200) {
      document.getElementById("demo").innerHTML = this.getAllResponseHeaders();
    } else if (this.readyState == 4) {
      document.getElementById("demo").innerHTML = this.status+" "+this.statusText;
    }
  };
  xhttp.open("HEAD", "info.txt", true);
  xhttp.send();
}

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

Rewrite the loadDoc() function by using ***async/await fetch approach***. The getAllResponseHeaders() method is not available under Fetch API. We can make use of the Headers.entries() method of the Response object to retrieve the set of Response headers. You can use the below getHeaders() method to replace the getAllResponseHeaders() method in your async/await fetch program.

```
//This function accepts one argument, which is the Response object
//It returns all the headers
function getHeaders(response) {
  let headers = "";
  // Display the key/value pairs
  for (var pair of response.headers.entries()) {
    headers += pair[0]+ ': '+ pair[1]+' ';
  }
  return headers;
}
```

Question 3 (22 points)

a) (10) Write a PHP program called ***getmark.php*** that pulls out data from a database which stores the assessment results of a course. The assessment results are stored in a table called ***gradebook*** which has the following structure. You can consider the column headers are the field names of the table. The table may contain zero to at most 200 records.

stdName	stdNumber	assign1	assign2	midterm	exam
Jim Ching	3015000000	30	61	72	67
Amy Lee	3015111111	85	72	65	81
Ben Hui	3015222222	null	50	55	61
Candy Sum	3015333333	70	41	60	63
David Wong	3015444444	69	50	null	75
Edvin Low	3015555555	79	74	69	72
Felix Lam	3015666666	84	81	70	90
Gary Yuen	3015777777	62	null	null	70
Henry Ho	3015888888	58	72	77	74
Ivan Luk	3015999999	72	78	69	81

Your PHP program connects to the MySQL database with the name "***c1111a***" at the server "***sophia.cs.hku.hk***" and the username "***c1111a***" with the password "***sudoku***".

If it successfully retrieves the data from the database, your PHP program returns all assessment results to the requested client in JSON format. If it experiences any error, your PHP program returns an HTTP status code 405 Method Not Allowed to the client. [Hint: you can use the PHP function http_response_code(405) to generate the 405 status code response.]

```
<?php




?>
```

b) (10) **Using jQuery** to write an event handler ***getmark()*** which is associated with the "Get Mark" button. Upon clicking on the button, this event handler sends an AJAX GET request to the ***getmark.php*** program to retrieve all assessment results of the course. If it successfully retrieves the data from the server, it displays the results in a table with the header row "stdName", "stdNumber", & "Final Score".
For each student, the value of the "Final Score" field is calculated by the following formula:

$$Final\ Score = assign1 \times 15\% + assign2 \times 15\% + midterm \times 20\% + exam \times 50\%$$

A null value represents a zero mark for that assessment. The table is displayed within the block with id="show".

Get Mark

stdName	stdNumber	Final Score
Jim Ching	3015000000	61.55
Amy Lee	3015111111	77.05
Ben Hui	3015222222	49
Candy Sum	3015333333	60.15
David Wong	3015444444	55.35
Edvin Low	3015555555	72.75
Felix Lam	3015666666	83.75
Gary Yuen	3015777777	44.3
Henry Ho	3015888888	71.9
Ivan Luk	3015999999	76.8

If it experiences any errors, it displays the returned response status message to the block.

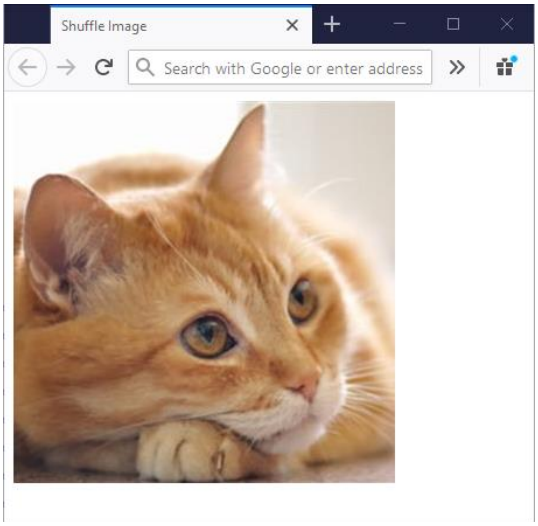
Complete the getmark() function.

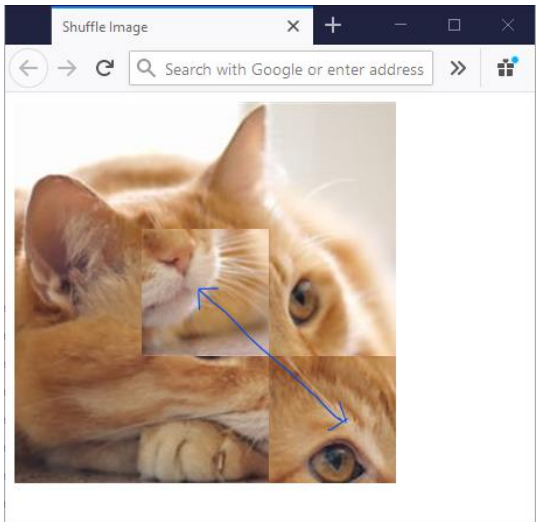
```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Get Mark</title>
  <script src='jquery-3.3.1.js'></script>
  <style>
    table, th, td {
      border: 1px solid blue;
      text-align: center;
    }
  </style>
</head>
<body>
  <p><button onclick="getmark(event);">Get Mark</button></p>
  <div id="show"></div>

  <script>
    function getmark(event) {









    }

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

Question 4 (20 points)

You are going to implement a simple web app that displays an image in a block. The block contains 9 canvases. Your web app divides the image into 9 pieces and places each piece on one canvas. The canvases do not have a border, padding, and margin, and they are nicely packed inside the block. The resulting picture reassembles the original image (as shown in the below diagram).

Each canvas is associated with an onClick handler. When the user clicks on a canvas, the web app randomly selects another canvas and switches the place with the onclick canvas. As shown in the example diagram, when the user clicked on the middle canvas, the program decided to switch its place with the bottom right canvas.





Below is the framework of the web app program. You are going to implement part (a) and part (b) of the program.

```
<!DOCTYPE html>
<html>
<head>
  <title>Shuffle Image</title>
  <script src='jquery-3.3.1.js'></script>
  <style>
    #sbox {
      width: 327px; height: 327px;
    }
    canvas { float: left; }
  </style>
</head>
<body>
  <div id="sbox">
    <canvas id="c1" width="109" height="109"></canvas>
    <canvas id="c2" width="109" height="109"></canvas>
    <canvas id="c3" width="109" height="109"></canvas>
    <canvas id="c4" width="109" height="109"></canvas>
    <canvas id="c5" width="109" height="109"></canvas>
    <canvas id="c6" width="109" height="109"></canvas>
    <canvas id="c7" width="109" height="109"></canvas>
    <canvas id="c8" width="109" height="109"></canvas>
    <canvas id="c9" width="109" height="109"></canvas>
  </div>
  <script>
    //This function returns a number between 0 to limit-1 and the returned
    //number should not be equal to the first parameter (index).
    function randomLoc(index, limit) {
      do {
```

```
        var r = Math.floor(Math.random()*Math.floor(limit));
      } while (r == index);
      return r;
    }

    draw();

    //Implement the draw() function
    //Split the image into 9 pieces and draw each piece on individual canvas.
    function draw() {
          part (a)
    }

    //Use jQuery to add the onclick event handler to each canvas
          part (b)

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

a) (10) Implement the draw() function using ***jQuery*** and/or ***native Javascript***. The function performs the following tasks:

1) It loads the image file "cat.jpg" to the web app.

2) It divides the image into 9 pieces and draws each piece on a canvas starting from the top left to the right and then going downward.

0	1	2
3	4	5
6	7	8

```
function draw() {









}
```

b) (10) Use ***jQuery*** to add the onClick event handler to each canvas. The event handler performs the following tasks upon the click event:
1) It tries to find which canvas triggered this event.
2) It uses the provided ramdonLoc() function to randomly selected another canvas. The randomLoc() function accepts two parameters. The first argument indicates the location of the onClick canvas within the block. For example, the middle canvas with "id=c5" would have a value of 4 and the bottom-right canvas with "id=c9" would have a value of 8. The second argument indicates the upper limit of the random number generator. In our case, you should always set the limit to 9.
3) It switches the positions of the onClick canvas with the randomly selected canvas.

```
//Use jQuery to add the onclick event handler to each canvas
```

Question X (Your answers in here will not affect your score)

a) How long did you take to complete this online quiz?

b) Did you use the developer tools in the quiz?

c) Please use a five-point scale to rate the difficulty level of the questions. With a '5' for very difficult, '4' for difficult, '3' for okay, '2' for easy, and '1' for very easy.

d) Do you think you have improved in your Web application development skills and knowledge after taking the course? Please use a five-point scale to provide your feedback. With a '5' for strongly agree, '4' for agree, '3' for neutral, '2' for disagree, and '1' for strongly disagree.

e) Any suggestions to help us to improve the course as well as to enhance your learning.

	Q1	Q2	Q3	Q4	Total	Total
Score					/60	/12