DEPARTMENT OF INFORMATICS

INF 272

SEMESTER TEST (1) **DATE: 2022-06-02**

Examiners	:	Dr. JP van Deventer	**Time** :	180 min
	:	Prof. MC Matthee		
	:	Mr. Z. Mahlaza		
	:	Mr. J Pienaar		
	:	Mr E van der Merwe		
Moderator / External Examiner	:	Mr. R Hanslo	**Marks** :	30

Student Number								Surname	Initials
--	--	--	--	--	--	--	--	-- MEMO --	MEMO

Question Section	Module outcomes (as found in the Study Guide)					Marks allocated	Maximum mark
	MO1	MO2	MO3	MO4	MO5		
Section A	XX				XX	10	10
Section B	XX				XX	10	10
Section C	XX				XX	10	10
Total						30	30

INSTRUCTIONS
1. This paper consists of three sections with several main questions (sub-sets of instructions) each.
2. Each section relates to a small semi-complete program that needs to be updated or finalised.
3. Each question relates to a file in one of the programs you need to update or finalise.
4. Each sub-sets of instructions relates to activities and tasks in one of the files.
5. Answer all the questions – there are no optional questions.
6. Please read all questions, instructions, and sub-sets of tasks very carefully.
7. After completing work on a relevant question, please upload ONLY the edited file to the correct upload area.

The University of Pretoria commits itself to produce academic work of integrity. I affirm that I am aware of and have read the Rules and Policies of the University, more specifically the Disciplinary Procedure and the Tests and Examinations Rules, which prohibit any unethical, dishonest, or improper conduct during tests, assignments, examinations and/or any other forms of assessment. I am aware that no student or any other person may assist or attempt to assist another student, or obtain help, or attempt to obtain help from another student or any other person during tests, assessments, assignments, examinations and/or any other forms of assessment.

SECTION A – HTML, CSS AND JAVASCRIPT (10)

Download the zipped folder named **2022 INF272 ST01 Section A.zip**. Unzip the folder on a local drive. Please refer to the stipulations on file preparation that may be found on the Semester Test General Information page as linked under the Assessment page on ClickUP.

In the folder you will find the following files:

- Index.html – containing the HTML for this project.
- style.css – the Cascading Style Sheet file.
- script.js – the JavaScript file.

Your task is to complete the code in the **JavaScript file, the HTML file and the CSS file** in order to get this application to function as described below. The program is a simulation of a game of darts (simplified).

When the user clicks on the Play button, one of the concentric rings 'lights' up or brightens – to simulate the target of the dart. This is achieved by changing the background colour to a lighter shade. Once the red circle is hit, it brightens, and the words Bullseye appears below the buttons as well as the number of times the user hits the bullseye on the dartboard. The user has to reset the board after each throw of the dart (or each click of the Play button).

The steps required to achieve this have been broken down into separate questions. For each question, please do the following:

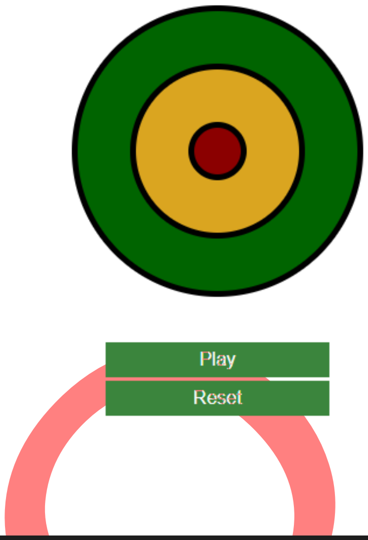
- Read the instructions carefully.
- Look for comments (hints) in the provided sample solution. The comments will guide you in terms of where the code modifications are required.
- Apply the necessary code changes to the specified file in the required application.
- Only upload the modified file associated with the question to ClickUP.

(**IMPORTANT: DO NOT UPLOAD ZIP, TAR, RAR or SLN files**)

SECTION A - QUESTION 1.1 and 1.2 (Correcting the layout) (2 MARKS)

When rendering the HTML file in the zipped folder provided, three circles display underneath each other. You have to change the CSS and HTML files such that the dartboard looks like the one to the right. Note that if you cannot manage this, you will still be able to complete the questions that follow up on this question.

After the completion of this question, please upload the modified and updated HTML file (under question 1.1) and CSS file (under question 1.2) to the space provided for this question.



SUGGESTED SOLUTION (1.1)

```
<head>
    <meta charset="UTF-8">
    <title>Section 1</title>
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet" href="style.css">
</head>

<body>
    <container class="clearfix">
        <div class="green" id="0">
            <div class="yellow" id="1">
                <div class="red" id="2">
                </div>
            </div>
        </div>


    </container>
    </br>
    </br>

    <div class="playTheGame">
        <button type="button" id="play">Play</button>
        </br>
        </br>

    </div>
    <div class="resetGame">
        <button type="button" id="reset">Reset</button>

    </div>

    <div>
    </br>
</br>
        <p id="message"></p>
    </div>

    <script src="script.js"></script>
</body>
```

SUGGESTED SOLUTION (1.2)

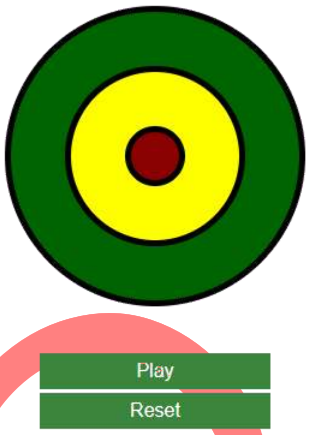
```css
.yellow {
  border-radius: 50%;
  display: flex;
  height: 150px;
  width: 150px;
  background: goldenrod;
  margin: 0 auto;
  align-items: center;
  border-style: solid;
  border-width: 5px;
  box-sizing: border-box;
}
.red {
  border-radius: 50%;
  height: 50px;
  width: 50px;
  background: darkred;
  margin: 0 auto;
  border-style: solid;
  border-width: 5px;
  box-sizing: border-box;
}
.green {
  display: flex;
  border-radius: 50%;
  height: 250px;
  width: 250px;
  background: darkgreen;
  margin: 0 auto;
  border-style: solid;
  border-width: 5px;
  box-sizing: border-box;
  align-items: center;
}
.playTheGame {
  display:flex;
  justify-content:center;
  align-items:center;
}
.resetGame {
  display:flex;
  justify-content:center;
  align-items:center;
}
#message {
  text-align:center;
  margin: auto;
  font-family: 'Lucida Sans', 'Lucida Sans Regular', 'Lucida Grande', 'Lucida Sans Unicode', Geneva,
Verdana, sans-serif;
  font-size: large;
  color: red;
}
button {
  background-color: #3b853d; /* Green */
  border: none;
  color: white;
  width:12em;
  height: 30px;
  /* padding: 12px 32px; */
  text-align: center;
  text-decoration: none;
  display: inline-block;
  font-size: 16px;
}
```

SECTION A - QUESTION 2 (8 MARKS)
(Creating the game and providing feedback to the user)

Add code to the script.js file to create the game:

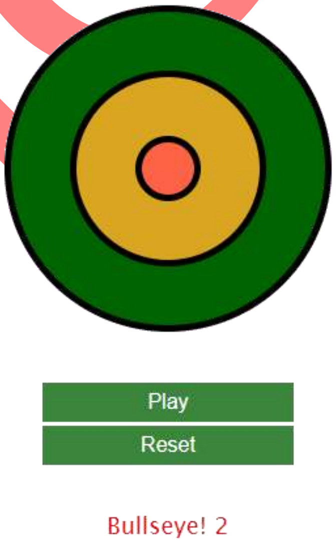
- When Play is clicked, a random circle brightens (either green, yellow, or red). This is done by changing the background colour to a lighter shade. We suggest the following: for the yellow bulb, change it to “yellow”, the red bulb to “tomato” and the green bulb to “light green”. See the example to the right where the yellow circle was hit
- When Reset is clicked, all colours change back to the darker shades and the comment below the buttons is removed.

[6 Marks]



When the red circle is hit, the word “Bullseye!” appears below the buttons as well as the number of times the user hit Bullseye at that stage. Please refer to the example provided.

[2 marks]



After completing Question 2 please upload the JavaScript file to the space provided.

SUGGESTED SOLUTION

```
window.addEventListener("load", function () {

    let startPlay = document.getElementById("play");


   //add event listener to button
    startPlay.addEventListener('click', function () {
        play();
    });

    this.document.getElementById("reset").addEventListener('click',function(){
        reset();
    })

});
var count = 0;
function play() {
    let randomNum = Math.floor(Math.random() * 3);
    typeof(points);
```

```javascript
    console.log(randomNum);
    switch (randomNum) {
        case 0:
            {
                document.getElementById("0").style.background = "lightgreen";
                break;
            }
        case 1:
            {
                document.getElementById("1").style.background = "yellow";
                break;
            }
        case 2:
            {
                document.getElementById("2").style.background = "tomato";
                count += 1;
                document.getElementById("message").innerHTML = "Bullseye!" + " " + count;
                break;
            }
            default:
                {

                }
    }
};

function reset() {

    document.getElementById("0").style.background = "darkgreen";
    document.getElementById("1").style.background = "goldenrod";
    document.getElementById("2").style.background = "darkred";
    document.getElementById("message").innerHTML = " ";

}
```

SECTION A TOTAL: 10
TEST TOTAL: 30

SECTION B – JQUERY AND BOOTSTRAP (10)

This section will require students to work with two separate small incomplete programs. Download the zipped folders as stipulated below. Unzip the folders on a local drive. Please refer to the stipulations on file preparation that may be found on the Semester Test General Information page as linked under the Assessment page on ClickUP.

- **2022 INF272 ST01 Section B1.zip – Bootstrap question.**
- **2022 INF272 ST01 Section B2.zip – jQuery question.**

Your task is to complete the code base in order to get this application to function as described below. The steps required to achieve this have been broken down into separate questions. For each question, please do the following:
- Read the instructions carefully.
- Look for comments (hints) in the provided sample solution. The comments will guide you in terms of where the code modifications are required.
- Apply the necessary code changes to the specified file in the required application.
- Only upload the modified file associated with the question to ClickUP.

(**IMPORTANT:** **DO NOT UPLOAD ZIP, TAR, RAR or SLN files)**

SECTION B - QUESTION 1 (Bootstrap) (5 MARKS)

Only upload the **Index.cshtml** file after completing this question. This file may be found in the Views folder.

A company is interested in selling gift baskets/boxes for the winter period. Their homepage currently shows all their offerings in a grid. They have decided to make improvements to their websites since their clients use devices with various screen sizes.

They have tasked you with using bootstrap to make their homepage more responsive.

Create a grid layout in **Views** / **Home** / **Index.cshtml** with cards that lists the various winter packages/boxes offered by the company.

If you want to hardcode the packages/boxes in the view, then you can use the data provided in the file named '**Q1WinterBoxesData.csv**'. If you want to use Razor, then there is no need for additional changes since the **HomeController**'s constructor already creates the boxes and the **Index** action already passes the list of boxes to the view.

Your grid layout must abide by the following restrictions:

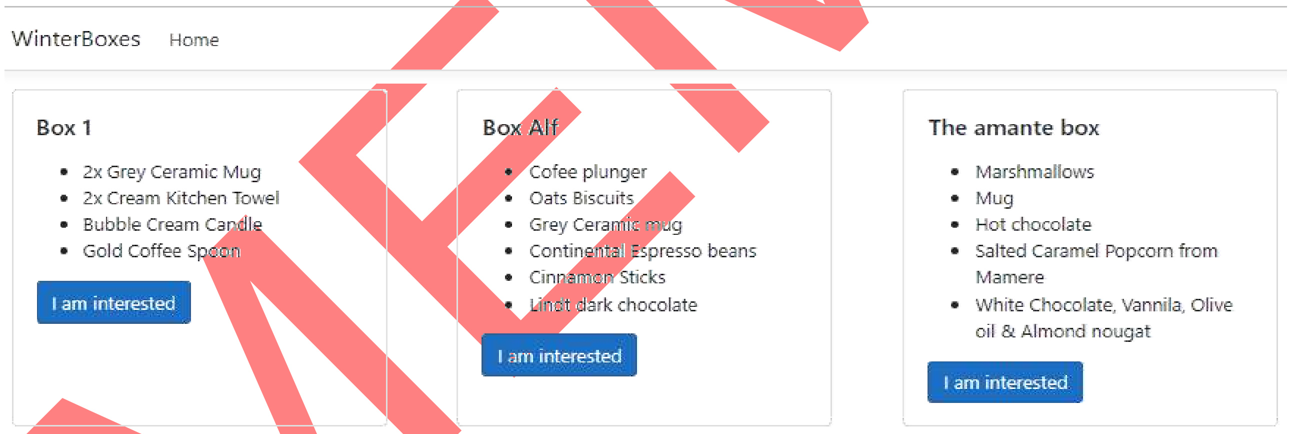
Q1 Expected Output

(a) Each card must have a width of "20rem" and it must show the winter package/box's name as the title, the body must list the items found in the box, and there must be a button at the bottom labelled "I am interested"

[**2 Marks**].

(b) When the screen is displayed in the large (≥992px) and extra-large (≥1200px) sizes, it must display three cards in a row. For smaller screen sizes, you can use your own discretion

[**2 Marks**].


WinterBoxes Home

Box 1
- 2x Grey Ceramic Mug
- 2x Cream Kitchen Towel
- Bubble Cream Candle
- Gold Coffee Spoon

I am interested

Box Alf
- Cofee plunger
- Oats Biscuits
- Grey Ceramic mug
- Continental Espresso beans
- Cinnamon Sticks
- Lindt dark chocolate

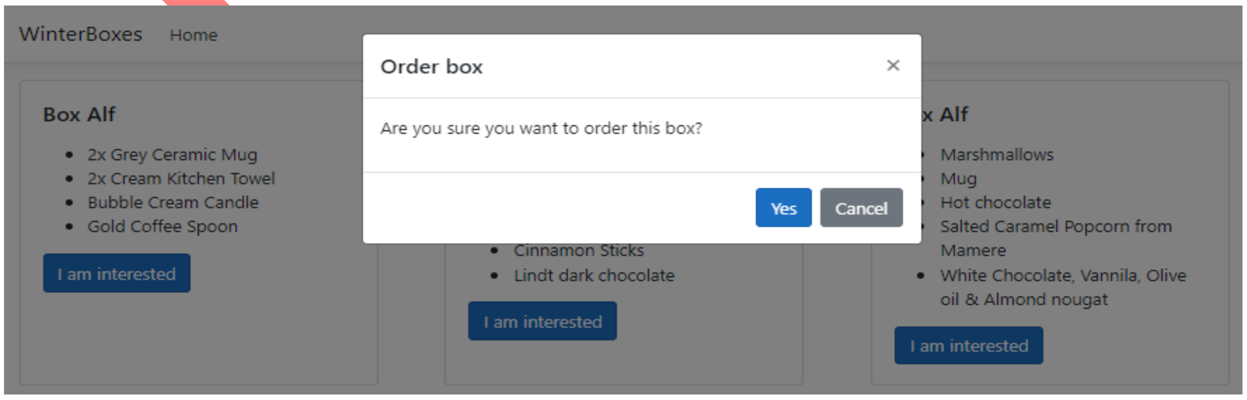
I am interested

The amante box
- Marshmallows
- Mug
- Hot chocolate
- Salted Caramel Popcorn from Mamere
- White Chocolate, Vannila, Olive oil & Almond nougat

I am interested

(c) When the button labelled "I am interested" is clicked, you must open a modal window whose title is "Order box" and the body must read "Are you sure you want to order this box?". There is no need to execute any action when the modal's buttons are clicked.

[**1 Mark**].


WinterBoxes Home

Order box ×

Are you sure you want to order this box?

Yes Cancel

Box Alf
- 2x Grey Ceramic Mug
- 2x Cream Kitchen Towel
- Bubble Cream Candle
- Gold Coffee Spoon

I am interested

- Cinnamon Sticks
- Lindt dark chocolate

I am interested

x Alf
- Marshmallows
- Mug
- Hot chocolate
- Salted Caramel Popcorn from Mamere
- White Chocolate, Vannila, Olive oil & Almond nougat

I am interested

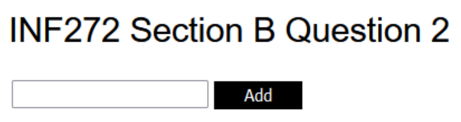
SUGGESTED SOLUTION

```
@model List<Box>

<div class="row">
    @foreach (var box in Model)
    {
        <div class="col-xl-4 col-lg-4 col-md-6 mb-3">
            <div class="card h-100" style="width: 20rem;">
                <div class="card-body">
                    <h5 class="card-title">@box.BoxName</h5>
                    <p class="card-text">
                        <ul>
                            @foreach (var item in box.Items)
                            {
                                <li>@item</li>
                            }
                        </ul>
                    </p>
                    <button type="button" class="btn btn-primary" data-toggle="modal" data-
                                          target="#exampleModal">
                        I am interested
                    </button>
                </div>
            </div>
        </div>
    }
</div>

<div class="modal" tabindex="-1" role="dialog" id="exampleModal">
    <div class="modal-dialog" role="document">
        <div class="modal-content">
            <div class="modal-header">
                <h5 class="modal-title">Order box</h5>
                <button type="button" class="close" data-dismiss="modal" aria-label="Close">
                    <span aria-hidden="true">&times;</span>
                </button>
            </div>
            <div class="modal-body">
                <p>Are you sure you want to order this box?</p>
            </div>
            <div class="modal-footer">
                <button type="button" class="btn btn-primary">Yes</button>
                <button type="button" class="btn btn-secondary" data-dismiss="modal">Cancel</button>
            </div>
        </div>
    </div>
</div>
```

SECTION B - QUESTION 2 (jQuery) (5 MARKS)

Only upload the **index.html** file after completing this question. This file may be found in the Views folder.

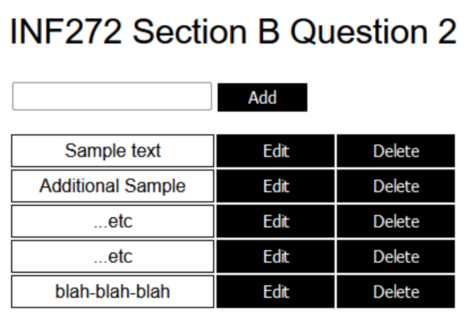
Complete the missing jQuery code required for the following functionality. Refer to the internal documentation in the HTML to identify where the missing code should be added / edited.



INF272 Section B Question 2

Add

When index.html is opened for the first time you will be presented with a textbox as well as a button. The textbox can accept any string-based input. After a string has been typed in the textbox, and the add button is clicked, then the text is added to a table as seen below. The table includes an Edit and a Delete button as generated by jQuery by means of a loop as well as an indexing array.



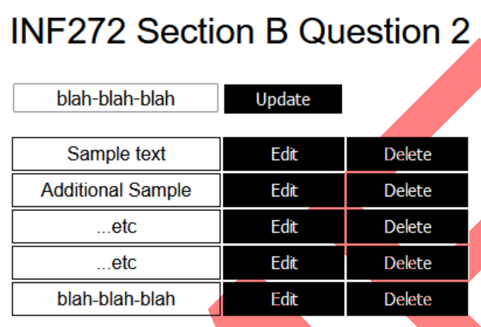
INF272 Section B Question 2

Add

Sample text	Edit	Delete
Additional Sample	Edit	Delete
...etc	Edit	Delete
...etc	Edit	Delete
blah-blah-blah	Edit	Delete

As may be seen, five samples of text have been added by clicking the Add button. When for example the Delete button is clicked, that particular row will be deleted. If for example the Edit button is clicked, then the text in the row will be added to the textbox so that it can be edited.

When the Edit button is clicked, not only does the text to be edited become active in the textbox but the Add button is temporarily replaced by an Update button.



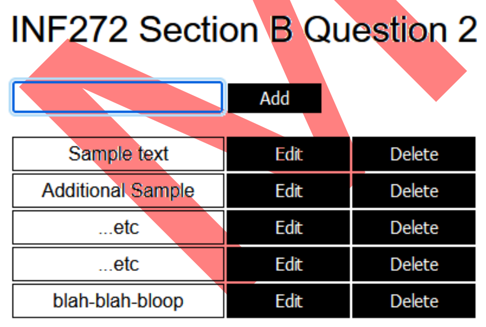
INF272 Section B Question 2

blah-blah-blah Update

Sample text	Edit	Delete
Additional Sample	Edit	Delete
...etc	Edit	Delete
...etc	Edit	Delete
blah-blah-blah	Edit	Delete

In this instance, the final row was selected by means of clicking the Edit button. The text in the row has been moved to the textbox and the Add button has been temporarily replaced by the (previously hidden) Update button.

The user can now edit the text loaded into the textbox. After the text has been edited, then, when the update button is clicked, the text selected to be edited will be replaced with the new text.



INF272 Section B Question 2

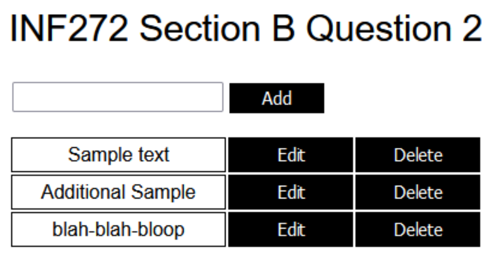
Add

Sample text	Edit	Delete
Additional Sample	Edit	Delete
...etc	Edit	Delete
...etc	Edit	Delete
blah-blah-bloop	Edit	Delete

As may be seen, the text has been updated and the last row has been replaced with a new instance of text.

Please note, that this process can include any of the indicated rows. The last row was only selected as an example.

In the following example, you would see that two rows have been deleted by clicking on the associated Delete buttons.



INF272 Section B Question 2

Add

Sample text	Edit	Delete
Additional Sample	Edit	Delete
blah-blah-bloop	Edit	Delete

Instructions: Complete the required jQuery to facilitate the aforementioned functionality. The missing section will be stipulated within the index.html file by means of internal documentation.

Please complete the missing jQuery after considering the jQuery that may still be found in the index.html file.

After updating the missing jQuery, please upload only the index.html file to the required upload area.

SUGGESTED SOLUTION

```
        $("#tblTable").on("click", ".edit", function () {
            var row = $(this).closest('tr');
            var td = $(row).find("td");
            $('#hfRowIndex').val($(row).index());
            $('#txtText').val($(td).eq(0).html());
            $('#btnAdd').hide();
            $('#btnUpdate').show();
        });
    });
    // -----------------------------------------------------
    // Update values

    $('#btnUpdate').on('click', function () {
        var name, country, id;
        id = $("#txtText").val();
        $('#tblTable tbody tr').eq($('#hfRowIndex').val()).find('td').eq(0).html(id);
        $('#btnAdd').show();
        $('#btnUpdate').hide();

    });
```

SECTION B TOTAL: 10
TEST TOTAL: 30

SECTION C – MVC AND RAZOR (10)

This section will require students to work with one incomplete program. Download the zipped folder as stipulated below. Unzip the folders on a local drive. Please refer to the stipulations on file preparation that may be found on the Semester Test General Information page as linked under the Assessment page on ClickUP.

- **2022 INF272 ST01 Section C.zip**

The program has two Views named **Question01.cshtml** and **Question02.cshtml**. The files may be found in the **View** / **Home** folder.

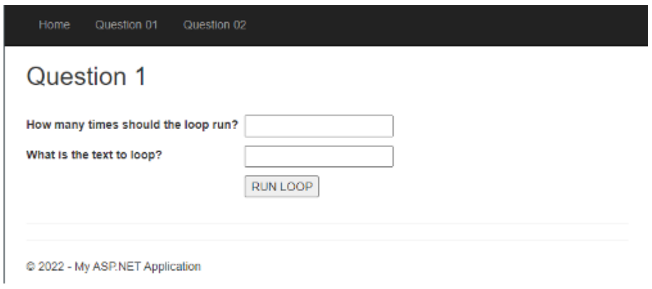
Your task is to complete the code base in order to get this application to function as described below. The steps required to achieve this have been broken down into separate questions. For each question, please do the following:
- Read the instructions carefully.
- Look for comments (hints) in the provided sample solution. The comments will guide you in terms of where the code modifications are required.
- Apply the necessary code changes to the specified file in the required application.
- Only upload the modified file associated with the question to ClickUP.

(**IMPORTANT:** **DO NOT UPLOAD ZIP, TAR, RAR or SLN files)**

SECTION C - QUESTION 1 (5 MARKS)

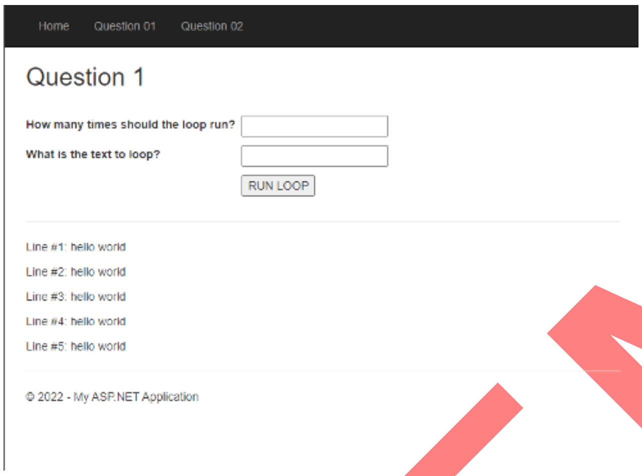
Only upload the **Question01.cshtml** file after completing this question. This file may be found in the Views folder.

Complete the code required for the following functionality. Refer to the internal documentation in the HTML to identify where the missing code should be added / edited.


Home
Question 01
Question 02
Question 1
How many times should the loop run?
What is the text to loop?
RUN LOOP
© 2022 - My ASP.NET Application

When running the MVC project for the first time, you will be presented with the stipulated view. The view will contain two textboxes as well as one button labelled "RUN LOOP".

The first textbox indicates the number of times a loop should be executed, and the second textbox indicates the text that should be displayed as may be seen in the following example.


Home
Question 01
Question 02
Question 1
How many times should the loop run?
What is the text to loop?
RUN LOOP
© 2022 - My ASP.NET Application

```
Line #1: hello world
Line #2: hello world
Line #3: hello world
Line #4: hello world
Line #5: hello world
```

In this instance, you can see that the user indicated that the loop should be executed five times and the line "hello world" was repeated. The words "Line 1#: " should be added as part of the loop. The user should not add this detail.

Instruction: Create a loop that would make use of an If statement as well as a For loop that would enable the indicated functionality. The functionality will be executed when the user clicks the RUN LOOP button.

Complete the missing Razor Block after considering the details found in the Question01.cshtml file. Refer to the internal documentation to identify where the relevant code block should be added.

After updating the missing Razor Block, please upload only the Question01.cshtml file to the required upload area.

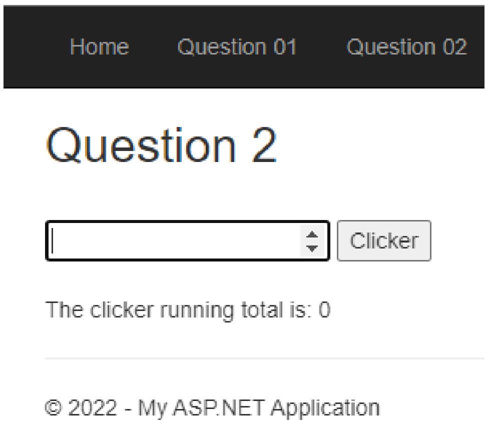
SUGGESTED SOLUTION

```
@{
    var loopCount = 0;

    if (IsPost)

    {
        var timesToRunLoop = Request["timesToLoop"];

        var textToShowInLoop = Request["textToLoop"];

        loopCount = timesToRunLoop.AsInt();

        for (var loopRun = 0; loopRun < loopCount; loopRun++)

        {
            var loopNumberToDisplay = loopRun + 1;

            <p>Line #@loopNumberToDisplay: @textToShowInLoop </p>
        }
    }
}
```

SECTION C - QUESTION 2 (5 MARKS)

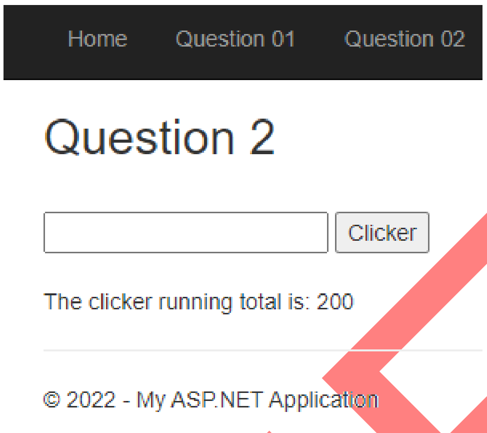
Only upload the **Question02.cshtml** file after completing this question. This file may be found in the Views folder.

Complete the code required for the following functionality. Refer to the internal documentation in the HTML to identify where the missing code should be added / edited.


Home
Question 01
Question 02
Question 2
Clicker
The clicker running total is: 0
© 2022 - My ASP.NET Application

When running the MVC project for the first time and selecting the Question 2 menu item, you will be presented with the stipulated view. The view will contain one textbox that can only accept numbers (both positive and negative) as well as a button labelled “Clicker”. As may be seen a label is presented stating that “The clicker running total is: 0”.

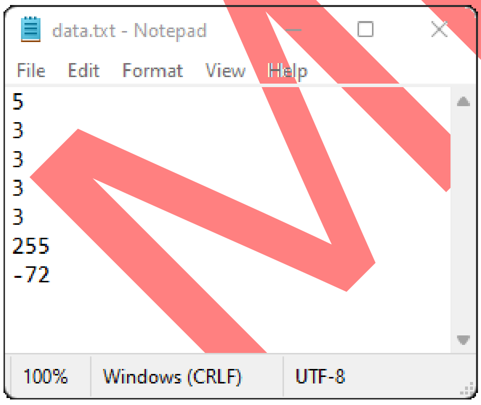
The value associated with the running total is stored sequentially in a text file called data.txt that may be found in the App_Data folder. This file is created for you by default when the MVC project is executed for the first time.


Home
Question 01
Question 02
Question 2
Clicker
The clicker running total is: 200
© 2022 - My ASP.NET Application

When a number is added to the textbox and the Clicker button is selected, the data is saved to the text file and a new line is added at the end of the file so that a new number can be added after it.

Data from the file is then obtained from the temporary text file and read into an array that is used in a loop to add all the values together and to display it as a running total.

In this example, a total number value of numbers adding up to two hundred has been added to the textbox and the clicker was used to add the numbers. The values that were stored sequentially were added up and the final result was two hundred.



```
5
3
3
3
3
255
-72
```

As may be seen, the numbers stored in data.txt is stored sequentially and when added up, produces a total value of two hundred.

Instruction: By making use of a Razor Syntax code block, complete the missing code that is necessary to (1) store the data sequentially in the created text file, (2) access the sequential data and then finally (3) add the numbers together as indicated in the aforementioned examples.

Refer to the internal documentation to identify where the relevant code block should be added.

After updating the missing Razor Block, please upload only the Question02.cshtml file to the required upload area.

SUGGESTED SOLUTION

```
@{
    ViewBag.Title = "Question 2";
}

<h2>Question 2</h2>

<br />

@{
    var dataFile = Server.MapPath("~/App_Data/data.txt");

    var dataToSave = "";

    File.AppendAllText(@dataFile, dataToSave);

    var runningTotal = 0;

    // Required for Question 2
    // ---------------------------------------------
    if (IsPost)
    {
        var number = Request["num"];

        dataToSave = number + Environment.NewLine;

        File.AppendAllText(@dataFile, dataToSave);

    }
    // Required for Question 2
    // ---------------------------------------------
}

<form id="form" method="post">
    <div>
        <table>
            <tr>
                <td><input name="num" type="number" autofocus value="" /></td>
                <td>&nbsp;<input type="submit" value="Clicker" /></td>
            </tr>
        </table>
    </div>
</form>

@{
    Array dataFromFile = null;

    dataFromFile = File.ReadAllLines(dataFile);
}

    <!-- Required for Question 2
    -------------------------------------------->

@foreach (string dataItem in dataFromFile)
{
    runningTotal = runningTotal + dataItem.AsInt();
}

<br />
<p>The clicker running total is: @runningTotal</p>

    <!-- Required for Question 2
    -------------------------------------------->
```

SECTION C TOTAL: 10
TEST TOTAL: 30