Task 1: Complete Exercise 1 from the Week 2 PowerPoint (Dice Roller)

1. Create a new dotnet core Razor Pages web application
2. Create a new Razor page in the application called **DiceRoller.cshtml**
3. Use the starter code as provided in the power point slides (slide 54) to replace the razor code block
4. Use Razor, C#, and HTML to deliver the following
   1. Display the current random number
   2. If the current random number equals the winning random number then display a **winning**message indicating the win
   3. If the current random number does not equal the winning number then display a **losing**message to indicate the loss
   4. Display how “far off” the random number was from the winning number. E.g. If the winning number is 4 and the randomNumber is 1 you would display a line about the user being “3 off”.
   5. Use elements from bootstrap to create a clear user interface

Task 1 Deliverables

1. Provide a copy of the source code of **DiceRoller.cshtml**
2. Provide a screenshot of the Webpage showing that it is running in your browser

**CODE:**

@page

@model IndexModel

@{

ViewData["Title"] = "Home page";

int randomNumber = new Random().Next(1, 7); //GENERATES NUMBER BETWEEN 1 and 6

int winningNumber = 2;

}

<div class="text-center">

<h3 class="display-4">Welcome to Dice Roller example</h3>

</div>

<p class="text-center">The exercise tells us if the random generated number is our Winning Number or not.</p>

<br />

@if (randomNumber == winningNumber)

{

<div class="text-center">

<p>Displays the Winning Number: @randomNumber</p><span>Message: </span><span class="bg-success">We Won</span>

</div>

}

else

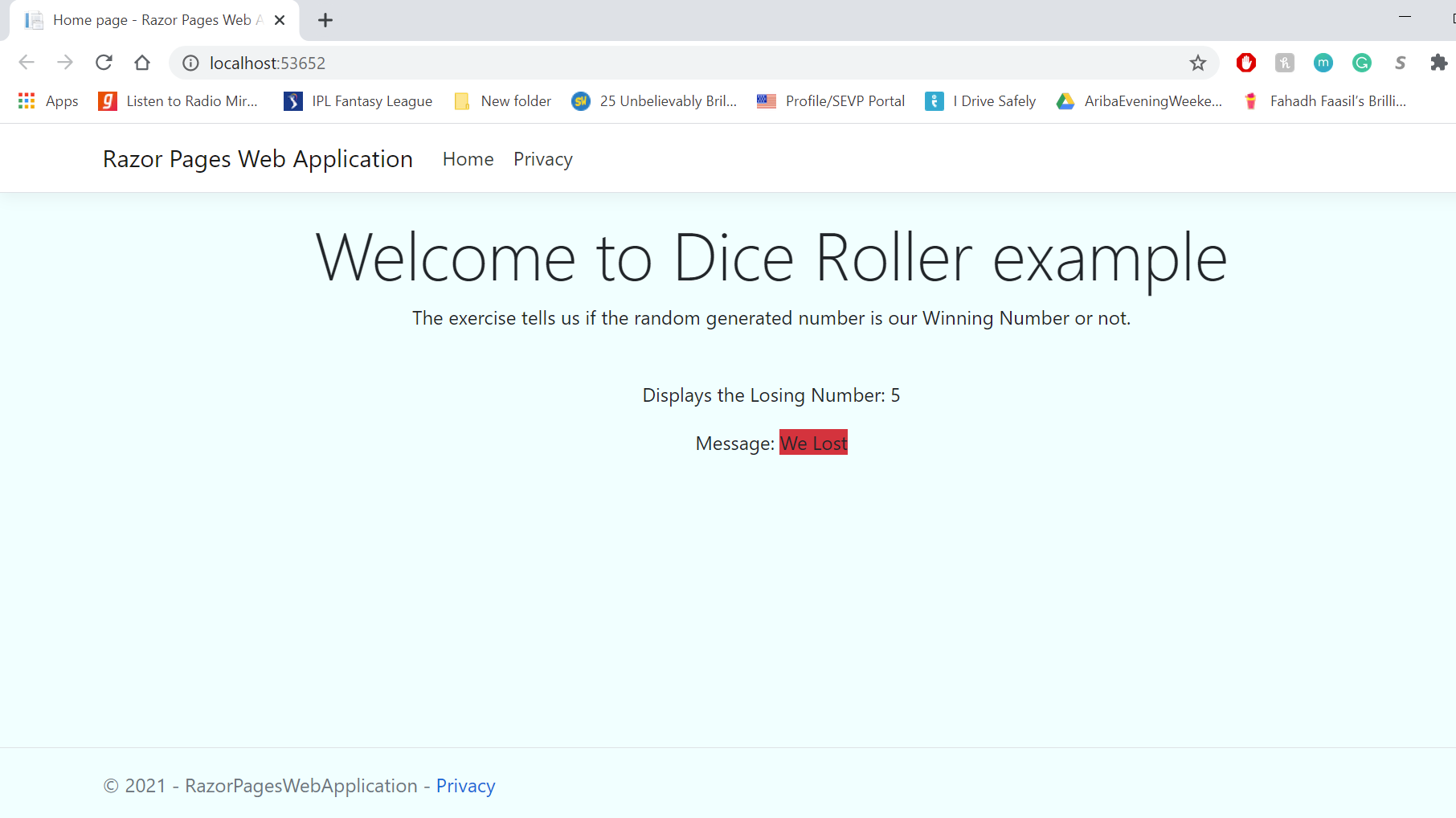
{

<div class="text-center">

<p>Displays the Losing Number: @randomNumber</p><span>Message: </span><span class="bg-danger">We Lost</span>

</div>

}



Task 2: Complete Exercise 2 from the Week 2 Power Point (broken project)

\* NOTE THIS PROJECT DOES NOT USE THE BOOTSTRAP UI AND IS INTENDED FOR THIS TASK ONLY. DO NOT COPY THE LOOK AND FEEL FROM THIS PROJECT INTO OTHER PROJECTS FOR THIS CLASS \*

1. Locate the Zip File on the Module 2 Notes
2. Before you unzip the file right-click on the file. Select properties and select unblock
3. Locate and open the Solution File (Called ExploreCalifornia.Monolith or ExploreCalifornia.Monolith.sln depending on your settings)
4. Build the application. You should see that there are a number of errors.
5. Work through each error (note almost all errors can be resolved by visually reviewing the code. The last 2 will take a little extra effort / googling. You will not need to add any new files, or write or delete more than a couple of characters.
6. Once the application is correctly fixed then when run the debugger it will look like the output on slide 56

Task 2 Deliverables

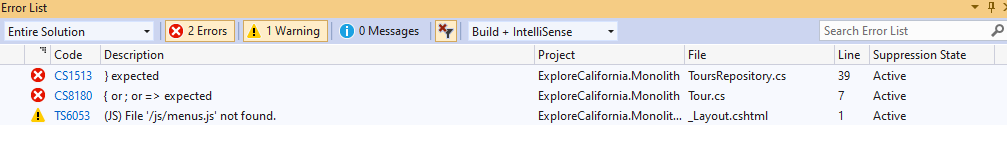
1. You only need to provide a copy of each line that you corrected pasted into a word document.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

I have successfully extracted the zip file and located the ‘ExploreCalifornia.Monolith’.

Then I have succefully opened it on Visual Studio.

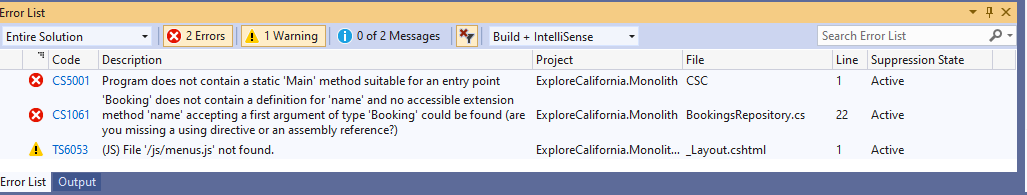
I have Observed following errors while Running the Website.



* Fixed error1: CS1513 by adding a parenthesis “}” at line 39.
* Fixed error2: CS8180 by adding semicolon “;” at line 7.

public string Description { get; set; }

When I ran the program second time, it gave following 2 errors:



* Fixed error2: CS1061 by removing “.name” after the booking as it does not contain a definition for ‘name’ and no accessible extension method accepting the first argument ‘Booking’.

new { TourId = booking.TourId, Name = booking, Email = booking.Email, Transport = booking.Transport });

* Fixed error 1: CS5001 by removing extra ‘I’ in “Maiin” and Build code, then Run the solution.

public static void Main(string[] args)

* Fixed the Last Error: by removing @\*……\*@ in the index.cshtml, as it is not part of the syntax.

