MVC Basics, Guessing Game   ASP.NET Fundamentals

Add a new view containing a number-guessing game.

The page should include a form that allows you to input a number. When the page is loaded for the first time, the controller should generate a random number between 1 and 100, that it will save in a session so it remembers it for the page even if it is refreshed.

When you submit a number into the form, it should compare your number to the one the controller generated. If it is the correct one, you should get a message congratulating you on your success, and a new number should be generated.

Otherwise, it should tell you whether your guess is either too high or too low, and let you take another guess.

Required Features:

- A randomly generated number, hidden from the user of the page.
- The ability to guess, through a form, which number was generated.
- A message displaying how well you did.

Optional:

- Add a counter for how many times the user has guessed, and display it after every failed guess. This should reset when the correct number is guessed.
- Add a high score list that is stored in a cookie, and displayed on the page.

Code Requirements:

- The guessing game View should be accessible through a custom route, using the "/GuessingGame" pattern, regardless of what the controller is named.
- The random number should be stored in the Session state.
- Guessing should be handled through overloaded Actions, referring to the same View.
    - When the page is loaded through a GET request (such as through the URL), the app should reset, and start over with a new number.
    - When the page is loaded through a POST request, it should make a guess, unless the value isn't provided or invalid, in which case it should display an error message.

MVC Basics, Guessing Game   ASP.NET Fundamentals

Resources:

- https://app.pluralsight.com/library/courses/building-aspdotnet-core-mvc-web-applications/table-of-contents

Subjects Covered:

- State management.
    - Session
    - Cookies
- Overloaded actions with GET and POST