

Wstęp.

Routing i podstawowe widoki silnika Razor w ramach ASP .Net.

List zadań

Dla wszystkich zadań wynik wizualnie ma się pojawić na stronie WWW (oczywiście numer portu może być inny niż w przykładach w zadaniu). W ramach rozwiązania powinny powstać odpowiednie metody routingu umożliwiające użycie adresów URL jak w podanych przykładach.

1. Stworzyć w aplikacji internetowej MVC kontroler `ToolController`, który rozwiązuje równanie kwadratowe $ax^2+bx+c=0$ (analogicznie jak dla listy 5, zadanie 1). Współczynniki równania mają być podane w adresie dla akcji `Solve`, czyli przykładowo $2x^2+3x+4=0$ ma być podane jako:
`http://localhost:12345/Tool/Solve/2/3/4`
Zastosować różne kolory/fonty/itp. na stronie w zależności od tego, czy są dwa, jedno, brak rozwiązań czy też równanie jest tożsamościowe. Najlepiej wykorzystać do klasy HTML oraz reguły i style CSS. Starać się **nie korzystać** z możliwości silnika Razor (oprócz przekazania danych do widoku), czyli wszelkie przetwarzanie powinno być wykonane w ramach kodu C# akcji kontrolera.
2. W tej samej aplikacji W kontrolerze `GameController` stworzyć grę-zgadywanekę. W kontrolerze w **zmiennych statycznych** (które będą pamiętane między żądaniami użytkownika) przechowywać wylosowaną liczbę z zakresu od 0 do (n-1) oraz wartość samą n. Ustawienie zakresu (n), np. na wartość 30 ma być wykonywane przez adres:
`http://localhost:12345/Set,30`
Wylosowanie nowego numeru z zakresu przez:
`http://localhost:12345/Draw`
Natomiast zgadywanie przez gracza liczby poprzez:
`http://localhost:12345/Guess,12`
Po każdej próbie na stronie pojawia się informacja, czy zgadujący podał za małą, za dużą liczbę lub zgadł. Program powinien również podawać, która to próba zgadnięcia liczby (licznik powinien być zerowany po wylosowaniu liczby).
Postarać się o estetykę odpowiedzi na stronach WWW analogicznie do zadania 1.

Data I: Spotkanie 10 (100 punktów)

Data II: Spotkanie 11 (80 punktów)

Data III: Spotkanie 12 (50 punktów)