

# Technologie internetowe

## lista zadań nr 7

Wszystkie zadania realizujemy wykorzystując ASP.NET lub analogiczny framework MVC o podobnych możliwościach.

1. Poeksperymentuj z różnymi typami, które może zwracać akcja kontrolera. Przygotuj przykład, w którym zostaną przedstawione 4 różne typy zawartości (np. string, JsonResult).  
[1p]
2. Bazując na przykładzie *MvcExample*, przygotować jedną stronę, na której będzie prezentowana lista osób w stronicowanej tabeli, czyli, dokładniej, widok ma zawierać pewien podzbiór listy elementów oraz *pagera* z numerami stron, za pomocą którego wybieramy kolejne podzbiory. Kolekcja ma być prostą kolekcją w pamięci i należy zadbać, aby utworzyć odpowiedni model prezentacji.  
[3p]
3. Bazując na przykładzie *RazorTemplates* utwórz serwis o 3 podstronach (Strona główna, Galeria i Kontakt) spełniający następujące postulaty:
  - Wszystkie ścieżki powinny mieć swoje nazwy, czyli należy odpowiednio uzupełnić kod w pliku *Startup.cs*.
  - Menu serwisu powinno być widokiem częściowym (*partial view*) o następujących pozycjach: Strona główna, Galeria, Kontakt. Dodatkowo linki do podstron powinny być utworzone w oparciu o nazwy ścieżek zdefiniowanych w poprzednim punkcie.
  - Podstrony *Strona główna* i *Kontakt* mogą pozostać puste, natomiast podstrona *Galeria* powinna zawierać 4 obrazki i podstrona powinna być zaimplementowana jako *ViewComponent*.

Przy rozwiązywaniu tego zadania należy wykorzystywać gdzie to możliwe *tag helpers*.

- [3p]
4. Bazując na przykładzie *FormValidation* utwórz serwis, w którym będą 2 podstrony: jedna do formularzem do zebrania danych o samochodzie, a druga potwierdzająca udaną operację, czyli zwykła strona w komunikatem *Operacja udana*. Formularz powinien być oparty o klasę modelu i zawierać następujące pola wraz z odpowiednią walidacją:
    - numer rejestracyjny (2 lub 3 litery, 4 – 6 cyfr),
    - data pierwszej rejestracji (poprawna data),
    - marka pojazdu (litery),
    - rok produkcji (liczba),
    - rodzaj paliwa (jeden z napisów P, ON, LPG, EE)

Jeśli dla jakiegoś pola nie uda się zrealizować walidacji przez atrybuty, należy tę walidację zrealizować w kontrolerze i w przypadku wyniku negatywnego dodać odpowiedni błąd za pomocą metody *ModelState.AddModelError()*. Po poprawnym wypełnieniu formularza należy przekierować na wspomnianą wcześniej stronę potwierdzającą powodzenie operacji.

[3p]