

Projektowanie aplikacji ADO.NET + ASP.NET

Zestaw 7

AJAX, JavaScript

2019-01-22

Liczba punktów do zdobycia: **15/73**
Zestaw ważny do końca semestru

1. (15p) (Mini projekt - sklep internetowy w ASP.NET MVC)

To zadanie podwyższa poprzeczkę, jaką postawiono w odpowiadającym zadaniu z jednego z poprzednich zestawów. W tamtym zadaniu, w WebForms, wiadomo było bowiem jak zorganizować sobie warstwę prezentacji danych. Tu - ta kwestia pozostaje otwarta. Do wyboru jest więc oparcie się o gotowe formanty (WebGrid) lub oprogramowanie całości we własnym zakresie.

Z poprzednim zadaniem można natomiast współdzielić część związaną z utrwalaniem danych oraz logowaniem użytkowników.

Ponownie więc należy zaimplementować widok towarów w sklepie internetowym - każdy towar ma mieć nazwę, opis, cenę i link do obrazka.

Należy zwrócić uwagę na poprawne przekazywanie parametrów **sortowania** i **wybierania strony** danych do warstwy danych w taki sposób, żeby zapytanie wysyłane ostatecznie do serwera bazy danych było możliwe optymalne (tzn. wybierało faktycznie jedną **stronę** danych w zadanym **porządku**).

Widok towarów powinien obsługiwać co najmniej

- dla zwykłego użytkownika (nawet bez logowania) - prezentację (użytkownik może przeglądać towary i dodawać do koszyka)
- dla administratora (po zalogowaniu) - dodatkowo modyfikację, dodawanie i usuwanie (administrator sklepu internetowego nie tylko przeląda listę towarów ale też może każdy z nich wyedytować oraz dodawać nowe)

O ile w przypadku WebForms nawigacja dla edycji/dodawania jest stosunkowo oczywista, o tyle na wykładzie mówiono o tym, że taka implementacja w której tabela pokazuje wiersze w trybie do-odczytu, a tylko jeden w trybie do-edycji, tu w MVC może być stosunkowo trudna. Łatwiej jest przekierować żądanie do osobnej strony, adresowanej dedykowaną ścieżką, typu `/ware/add` dla dodawania elementy czy `/ware/edit/123` dla edycji elementu o identyfikatorze 123.

Takie podejście rodzi dwa problemy, które należy rozwiązać. Po pierwsze, jeśli identyfikatory obiektów pojawiają się w pasku adresowym, należy upewnić się że aplikacja nie jest podatna na atak Query-string Tampering (użytkownik bez uprawnień może dostać się do zasobu do którego nie powinien przez modyfikację paska adresowego). Po drugie - po przejściu do osobnej strony (do dodawania, do edycji) użytkownik może nie wiedzieć jak wrócić

do widoku listy. Dlatego - jak powiedziano na wykładzie - należy tu zastosować dodatkowy element interfejsu użytkownika, służący do *nawigacji okraskowej* (ang. breadcrumb).

Proszę tym razem zaimplementować również wyszukiwanie towarów.

Każdy towar można dodać do koszyka z poziomu widoku towarów, a następnie przejść do osobnego widoku, w którym widać tylko te elementy które wcześniej dodano do koszyka.

Koszyk zaimplementować jako osobną klasę **ShoppingCart**, która wewnętrznie używa kontenera **Session** do tymczasowego przechowania wybranych towarów na czas sesji pracy użytkownika.

Wiktor Zychła