

## Wstęp

Celem listy zadań jest dodanie autentykacji i autoryzacji do rozwijanej aplikacji sklepu webowego. Do kontynuacji można wybrać albo realizację MVC albo strony Razorowe. Do wykonania tej listy należy użyć zestawu **Identity** omówionego na wykładzie.

Lista zadań wygląda na skomplikowaną, lecz w rzeczywistości są to drobne zadania, wiele z nich do zrobienia w kilka minut.

## Lista zadań

1. Dodać możliwość założenia konta w sklepie.
2. Dodać do aplikacji rolę administratora, która jako jedyna będzie mogła modyfikować kategorie i artykuły w sklepie, ale nie będzie miała dostępu do koszyka (potrzeba zrobić politykę - policy).
3. Niezalogowany użytkownik może dodawać towary do koszyka, jednak aby złożyć zamówienie (dodatkowy przycisk w koszyku) musi się zalogować.
4. Menu powinno odzwierciedlać możliwości użytkownika strony (niezalogowany/zalogowany/administrator)
5. Po poprawnym przejściu do strony zamówienia powinna się pojawić tabelka podsumowująca koszyk (już bez możliwości modyfikacji jego zawartości) oraz przycisk „potwierdź zamówienie”. Oprócz tego na tej stronie powinien być formularz do wpisania danych osobowych i adresu dostawy oraz wyboru sposobu płatności (lista rozwijalna możliwości, to tylko symulacja możliwości).
6. Po wciśnięciu potwierdzenia zamówienia powinna się pojawić strona potwierdzająca zamówienie pokazująca dane osobowe i wybrany sposób płatności. Koszyk powinien być wyczyszczony. Dla chętnych: zapisać zamówienie do bazy danych (wymaga migracji ze stworzeniem 3 tabel).
7. Dodać kontroler API do operowania na artykułach i kategoriach (operacje GET/POST/PUT/DELETE). Dla uproszczenia można założyć, że nie będziemy tworzyć artykułów z obrazkiem **za pomocą tego kontrolera** (czyli jakby użytkownik tworzył artykuł bez podania pliku z obrazkiem).
8. Przetestować działanie kontrolera API za pomocą Swagger-a lub Postman-a.
9. Czy i jak można ograniczyć dostęp tylko dla administratora do funkcji kontrolera API **modyfikujących** bazę danych? Dla chętnych – zaimplementowanie zaproponowanego rozwiązania.
10. W sklepie dodać **leniwe doładowywanie** kolejnych pozycji z artykułami.
  - a. Najprostsza wersja to pokazanie pierwszych  $n$  ( $n > 1$ ) towarów na liście oraz pod nimi przycisk „pokaz kolejne”. Dla przycisku stworzyć w JavaScriptcie funkcję, która AJAX-em skomunikuje się z kontrolerem API i doczyta kolejne  $n$  towarów

(w formacie JSON, tylko dane towarów) i stwórz odpowiednie elementy HTML.

- b. Wersja ciekawsza: doczytywanie uruchamia się automatycznie, gdy użytkownik przesunie stronę w dół do ostatniego widocznego artykułu (wówczas najczęściej wczytujemy więcej niż  $n$  artykułów na początku, bo trzeba wypełnić trochę więcej niż wysokość strony).

Oczywiście, celem demonstracji działania ostatniego punktu należy w bazie stworzyć dość dużą liczbę artykułów. Sprawdzić, czy poprawnie działają linki dodawania artykułu do koszyka dla „doczytanych” artykułów.

Data I: Spotkanie 13 (max 100 punktów)

Data II: Spotkanie 14 (max 80 punktów)

Data III: Spotkanie 15 (max 50 punktów)