

Zadanie rekrutacyjne - i like that

Zadanie polega na stworzeniu podstrony HTML5 z dodatkiem SCSS, JavaScript oraz jQuery (i ew. wtyczek do jQuery). Dobrze będzie wykorzystać np. flex z CSS.

Oceniana również będzie jakość i porządek kodu.

Powodzenia!

Na kroku pierwszym użytkownik dodaje frazy (ograniczenie do max 3 fraz), zatwierdzenie enterem lub przyciskiem “Add it!”:

The screenshot shows a web interface for a recruitment task. At the top left is the logo 'm rch'. To its right is a progress bar with three steps: 'Step 1' (active), 'Step 2', and 'Step 3'. Below the progress bar is a large text input field with the placeholder text 'Type something...'. To the right of the input field is a blue button labeled 'Add it!'. Below the input field, there is a small line of text: 'E.g. popular words & phrases that your community use, what people call you, things you say, your band or DJ name, initials, squad names etc.'

Przykład gdy dodanych jest kilka fraz i tekst podczas wpisywania:

Tats all you got?

Add it! x x x x xGenerate

Responsywny design powinien zostać stworzony na zasadzie “fluid design” oraz w stylu web aplikacji, czyli nawet na bardzo dużym ekranie powinien się rozciągać i powiększać różne elementy typu rozmiar fontów itp. (z ograniczeniem np. do 4k).

Czcionka: SharpSans - dołączona.

Gdy dodana jest minimum jedna fraza, pojawia się przycisk Generate.

Jako bonus: Hover tego przycisku byłby taki, że granatowy ozdobnik, który znajduje się na jego dole, mógłby zmniejszyć swoją wysokość o 2-3px. CSS/SCSS powinien zostać tak tutaj stworzony, aby w teorii dało się używać i jednego i drugiego stylu przycisku w innych modułach (np. dodatkowa klasa btn-primary). Chodzi o zwizualizowanie “wciskania przycisku”. Ilustruje to dołączone wideo “**hover przycisku primary.mov**”.

Frazy powinny zapisywać się np. w liście/arrayu w JS.

Po kliknięciu przycisku Generate, przechodzimy do kolejnego kroku (wskazana jakaś animacja przejścia, np. fade):



W powyższym przykładzie mamy t-shirt a na nim obszar roboczy. Każda z dodanych fraz zostaje domyślnie dodana jedna pod drugą w odstępach np. 20px.

Domyślnie kolory hex losujemy jakimś prostym algorytmem.

Należy tutaj wykorzystać bibliotekę <https://interactjs.io/> w celu możliwości przesuwania fraz myszką wewnątrz obszaru roboczego na koszulce. (proszę zerknąć na przykłady dragging, snapping w dokumentacji biblioteki).

Dla ułatwienia - ten krok nie musi być responsywny.

Przycisk jest zawsze przyklejony do dołu strony.

W prawej kolumnie znajduje się akordeon z frazami (wyjątkowo, ten element należy wykonać w czystym JavaScript/jQuery, tj. bez dodatkowych wtyczek jQuery typu accordion).

Po kliknięciu na daną frazę na obszarze roboczym na koszulce, lub rozwinięciu danego akordeonu z prawej kolumny, dana fraza na koszulce się zaznacza (wtyczka interactJS wszystko to obsługuje). Tak samo klikając na dany akordeon, obecnie aktywna fraza się odznacza, zaznacza ta powiązana z danym akordeonem i tylko on jest już widoczny. Przykład poniżej:



Gdy dana fraza jest aktywna, pojawia nam się opcja zmiany koloru poprzez slider (najprościej wykorzystać jakąś wtyczkę JS/jQuery, wskazówka: HSL UI-Slider).

Finalnie, klikając przycisk “Send Ajax Request with Data”, następuje wysłanie requesta POST poprzez ajax (z jQuery) na jakiś fikcyjny URL. Jego payload powinien zawierać tablicę w formacie JSON. Każdy element tablicy (1 fraza = 1 element) powinien mieć zagnieżdżone takie pola:

- text
- position_x
- position_y
- color

Po wykonaniu ajaxa, zmieniamy tekst buttona na “Done” oraz ustawiamy atrybut disable, który ostylowany powinien zostać w ten sposób:

