**JS zadania na podsumowanie**

Poniższe zadania będą opisem efektu, jaki klient chce otrzymać na stronie. Twoim zadaniem będzie stworzenie na podstawie jego opisu oraz podpowiedzi, strony i skryptu JS, który spełni jego oczekiwania i zademonstruje rozwiązanie problemu.

**Zad. 1**

**Story:**

W panelu administracyjnym do mojego bloga posiadam tabelkę, do której za pomocą API obecnie pobieram wszystkie komentarze do artykułów, a następnie mogę usuwać komentarze, które są uznawane za nieprzyzwoite. W moim panelu administracyjnym brakuje mi możliwości wyszukiwania i wyróżniania komentarzy nieprzyzwoitych.

**Podpowiedź***:*

Klient nie udostępnił nam swojego API, więc skorzystaj z API <https://jsonplaceholder.typicode.com/comments> aby wyświetlić komentarze w tabeli. Stwórz tabelę z nagłówkami **name**, **email**, **body** oraz **action**.

W polu action dodaj przycisk **usuń** i **zaloguj** w konsoli (usuń odnosi się do komentarza, jaki chcesz usunąć). Dodaj możliwość wyszukiwania w tabeli za pomocą inputów wyszukujących po polach name, email i body.

Dodaj możliwość zaznaczania kilku wierszy i przycisk do usuwania zaznaczonych.

**Opcjonalnie***:*

Dodaj checkboxa wyróżnij nieprzyzwoite, który w body lub name zaznaczy na żółto wiersz, który zawiera nieprzyzwoity komentarz. W tym wypadku zamiast wyszukiwać nieprzyzwoity wyraz wybierz jeden wyraz ze zbioru udostępnionego przez jsonplaceholder.

**Zad.2**

**Story***:*

Podczas zamówienia mebli mamy w firmie ponad sto kolorów, obecnie klient wybiera kolor z **selecta** co jest dla niego niewygodne. Mamy przez to mniejszą konwersję sprzedaży. Chcielibyśmy dać możliwość wyszukiwania koloru.

**Podpowiedź:**

Stwórz inputa, w którym po wpisywaniu przez użytkownika filtrowane będą dostępne opcje. Po kliknięciu na input pod nim wyświetlają się wszystkie możliwe opcje i nadal można pisać. Kolory nie muszą być wygenerowane, na potrzeby developmentu stwórz wyszukiwanie po liście elementów.

**Zad. 3**

**Story**:

Na mojej stronie fotograficznej obecnie pokazuję galerię wykonanych zdjęć (kafelki na całej stronie obok siebie o stałych rozmiarach). Chciałbym aby za każdym odświeżeniem strony do galerii ładowały się inne zdjęcia, a rozmiary kafelków generowały się dynamicznie. Po najechaniu myszą na zdjęcie wyświetlał się nazwa autora zdjęcia.

**Podpowiedź:**

Wykorzystaj Grida i jego zaawansowane właściwości do dopasowania kwadratów, tak aby zajmowały jak największą przestrzeń na stronie. Do załadowania przykładowych zdjęć wykorzystaj api unsplash [Unsplash API Documentation | Free HD Photo API](https://unsplash.com/documentation#get-a-random-photo)

**Zad. 4**

**Story:**

W naszej wewnętrznej aplikacji mamy wiele formularzy, które generujemy na bieżąco za pomocą edytora. Generowanie formularza za pomocą edytora zajmuje nawet kilka godzin, a potrzebujemy ich jak najszybciej. Chcielibyśmy otrzymać rozwiązanie, które pozwoli na szybkie tworzenie dynamicznych formularzy na podstawie parametrów konfiguracyjnych.

**Podpowiedź:**

Stwórz funkcję, która zamieni podanego jej JSONa w formularz, który następnie pojawi się na stronie. JSON powinien zawierać nazwę pola jego typ i walidację. Stwórz funkcję, która na podstawie walidacji stworzy do formularza przycisk i będzie decydować czy może go wysłać dalej czy nie i wskaże które pola są nieprawidłowo wypełnione.

**Zad. 5**

**Story:**

Nasza firma chciałaby otrzymać możliwość oglądania produktu za pomocą lupki tak jak w sklepie lego [Koloseum 10276 | Creator Expert | Oficjalnym sklepie LEGO® PL](https://www.lego.com/pl-pl/product/colosseum-10276). Po kliknięciu zdjęcie powiększy się i będzie można przesuwać się po nim.

**Podpowiedź:**

Wykorzystaj manipulowanie background position oraz obserwowanie pozycji myszy po kliknięciu na obrazek

**Zad. 6**

**Story:**

Klient zapisuje spotkania ze swoimi kontrahentami w bazie danych w odpowiednich godzinach. Brakuje mu możliwości wyświetlania spotkań na jednym widoku. Przygotuj tablicę obiektów, która będzie przechowywała godzinę, dzień, nazwę i dane kontrahenta w tabeli spotkań. Zainspiruj się widokiem poniżej. Obok nazwy dnia powinna wyświetlać się data. Domyślnie wyświetla się bieżący tydzień. Na dole dodaj przyciski do przełączania pomiędzy tygodniami.  


**Zad. 7**

**Arabic number conversion tool**

Create a web-based solution that converts numeric input into the English phrase of that number.

**For example:**

| 7 === seven  42 === forty-two  1999 === one thousand nine hundred and ninety-nine  2001 === two thousand and one  17999 === seventeen thousand nine hundred and ninety-nine  342251 === three hundred and forty-two thousand two hundred and fifty-one  1300420 === one million three hundred thousand four hundred and twenty |
| --- |

**Note**: Numbers greater than 100 and less than 1000 always include the word "and" between the hundreds figure and the other part, numbers greater than 1000 include "and" between any hundreds figure and the figure less than 100.

Treat this task like a project to create a real-life application, focus on other aspects too, not just the code:

* Fork this project.
* Use the best stack matching the simplicity of the task.
* Don't use external libraries for the conversion.
* Make the solution pleasant to look at and user friendly in as many aspects as you can.
* Commit the important milestones and not just the final result.
* Don't forget to write tests. We expect the tests to check also the examples above.
* Try to support numbers as high as you can.
* (Optional) Try to support British English counting where numbers between 1000 and 2000 are said out using "hundreds". E.g. 1999 === nineteen hundred and ninety-nine

Thank you for your time and happy coding! 🧑‍💻