

Wprowadzenie do aplikacji Internetowych

laboratorium 3

Cel zajęć:

Celem laboratorium jest przećwiczenie zagadnień związanych z programowaniem z wykorzystaniem JS. Skoncentrujemy się na aspektach zarządzania zdarzeniami oraz modyfikacji modelu DOM.

Zadania na rozgrzewkę (bez oceniania).

Zadanie 1. Po naciśnięciu na przycisk wyświetl okno dialogowe typu prompt pozwalające na wprowadzenie swojego imienia. Podane imię ma być wstawione do przygotowanej sekcji.

Zadanie 2. Przełączanie pomiędzy dwoma zdjęciami. Niech na stronie wyświetlone jest zdjęcie z gór w czerwonym obramowaniu. Po naciśnięciu przycisku zdjęcie zamieni się na zdjęcie morze w niebieskim obramowaniu. Kolejne naciśnięcie to ponowny powrót do wersji z górami itp.

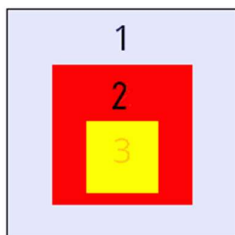
Zadanie 3. Napisz skrypt, w którym za pomocą 2 przycisków modyfikujesz zawartość listy elementów. Jeden przycisk dodaje a drugi usuwa elementy z listy. Niech usuwane będzie zawsze pierwszy element listy.

Zadania punktowane.

Sekcja podstawowa (max 10 pkt)

Zadanie 4. Napisz skrypt, który dodaje lub usuwa obsługę przycisku testowego. Naciśnięcie przycisku inkrementuje licznik przyciśnięć. Odpicie zdarzenia resetuje licznik. **(1pkt)**

Zadanie 5. Stwórz stronę zawierającą 3 elementy np. divy/zdjęcia spozycjonowane tak jak na rysunku poniżej. **(2pkt)**



Niech naciśnięcie któregośkolwiek z nich wyświetla odpowiedni komunikat w odpowiedniej sekcji na stronie.

Żółtego - „naciśnąłeś żółty o wartości 5”

Czerwonego – „naciśnąłeś czerwony o wartości 2”

Niebieski – „naciśnięś niebieski o wartości 1”

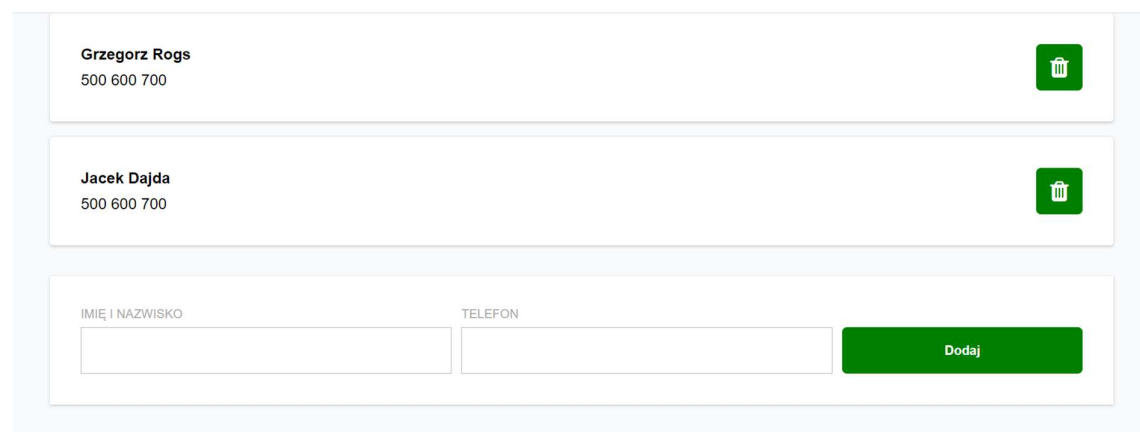
Jeśli suma naciśniętych wartości przekroczy 30 należy wyłączyć możliwość obsługi zdarzenia click przez obiekt 2. Jeśli 50 usuwamy dodatkowo obsługę zdarzenia przez obiekt 3.

Niech na stronie będą jeszcze 2 przyciski: Stop/StartPropagation, Reset .

Stop/StartPropagation – włącza i wyłącza propagację (niech tekst na przycisku odpowiada stanowi przycisku)

Reset – wraca do stanu początkowego – reset licznika punktów oraz możliwość obsługi zdarzeń typu click.

Zadanie 6. Napisz skrypt, w którym za pomocą przycisk Dodaj będzie dodawał nową pozycję do książki telefonicznej (imię, nr telefonu). Dane podajemy za pomocą formularza. Dodawana pozycja zawiera również przycisk (w formie ikony – kosz) pozwalający na usuwanie elementów z książki adresowej. Wymagana jest walidacja wprowadzanych wartości tak aby numer telefonu to były tylko cyfry w formacie 9 cyfr. Imię musi rozpoczynać się od dużej litery i zawierać tylko litery (cyfry i inne znaki wykluczone). Przykład poniżej. **(2pkt)**



The screenshot shows a web application interface. At the top, there is a list of contacts. Each contact entry consists of a white box containing the name and phone number, and a green trash icon to its right. The first contact is 'Grzegorz Rogs' with phone number '500 600 700'. The second contact is 'Jacek Dajda' with phone number '500 600 700'. Below the list is a form to add a new contact. It has two input fields: 'IMIĘ I NAZWISKO' and 'TELEFON'. To the right of the 'TELEFON' field is a green button labeled 'Dodaj'.

Zadanie 7. W katalogu Zadanie 7 znajdziesz plik city.json zawierający kolekcję miast Polski.

Celem ćwiczenia jest weryfikacja umiejętności przetwarzania plików zewnętrznych z wykorzystaniem Fetch API. Aby można było przetwarzać pliki zewnętrzne (pliki json-owe) potrzebujemy serwera który będzie je hostował. Proponuje użycie json-server. W celu jego zainstalowania potrzeba posiadania jakiegoś menedżera pakietów: npm lub yarn.

Aby zainstalować JSON Server, należy otworzyć konsolę i wpisać poniższe polecenie:

```
npm install -g json-server
```

Flaga -g spowoduje, że serwer zostanie zainstalowany globalnie w systemie, co pozwoli uruchomić go z każdego miejsca.

Aby uruchomić JSON Server, proszę otworzyć wiersz poleceń i wpisać:

json-server db.json

W tym momencie serwer udostępnia pliki db.json i będzie można przy użyciu Fetch API przeczytać jego zawartość.

Napisz funkcje które:

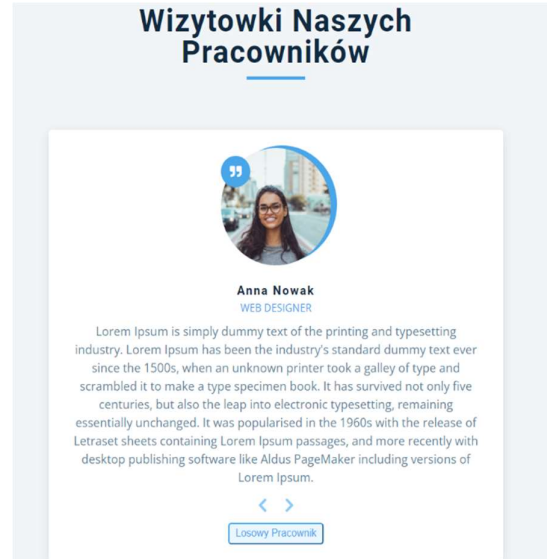
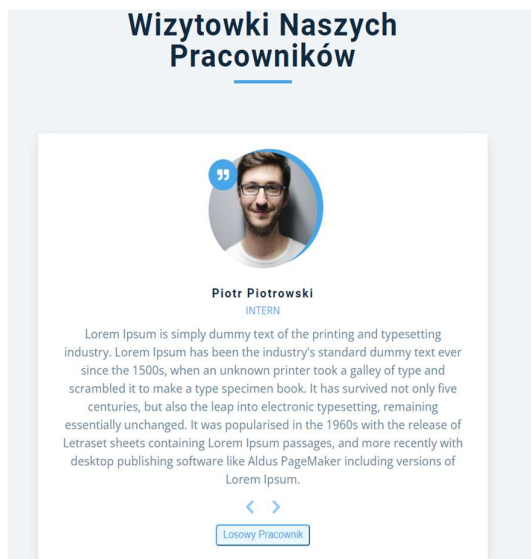
- a). wyświetli na stronie tylko miasta z województwa małopolskiego
- b). wyświetli miasta które w swojej nazwie posiadają dwa znaki 'a'
- c). wyświetli piate pod kątem gęstości zaludnienia miasto w Polsce.
- d). dla wszystkich miast powyżej 100000 dodać (na końcu) city do nazwy.
- e) wyliczyć czy więcej jest miast powyżej 80000 mieszkańców czy poniżej wraz z informacją o ich liczbie.
- f). jaka jest średnia powierzchnia miast z powiatów zaczynających się na literkę „P”

Wyniki wyświetlić na dedykowanych sekcjach na stronie. (2pkt)

Zadanie 8. Napisz stronę, która wyświetla ikonę balonika (używając emotikonu 🎈). Kiedy naciśniesz strzałkę w górę, powinno się napompować (wzrosnąć) o 10 procent, a po naciśnięciu strzałki w dół powinno opróżnić się (skurczyć) o 10 procent. Możesz kontrolować rozmiar tekstu (emotikony są tekstem), ustawiając właściwość CSS font-size (style.fontSize) na jego elemencie nadrzędnym. Pamiętaj, aby w wartości uwzględnić jednostkę — na przykład piksele (10px). Upewnij się, że klawisze „strzałka w górę” oraz „w dół” zmieniają tylko rozmiar balonika, bez przewijania strony. Przy spuszczeniu powietrza istnieje dolna granica wielkości balonika. (chodzi o to aby balonik nie zniknął z ekranu). Kiedy balonik przekroczy ustalony rozmiar pęka. W tym przypadku eksplodowanie oznacza, że jest ono zastępowane emotikonem 💣. Od tego momentu nie jest możliwe pompowanie lub spuszczenie

Jeśli na baloniku (i tylko na nim) naciśniesz Ctrl i prawy klawisz myszki – wywołasz stworzone przez Ciebie menu kontekstowe, które powinno wyświetlić informacje na temat aktualnej wielkości balonika. (2pkt)

Zadanie 9. Napisz slider wizytówek. Wizytówka składa się ze zdjęcia, stanowiska i przykładowego opisu. Przyciski < i > pozwalają na przeglądanie naszych pracowników. Dodatkowo istnieje przycisk do losowego wyboru wizytówki do wyświetlenia (1pkt)



Sekcja rozszerzona (max 9 pkt)

Zad 10. Przetwarzanie plików zewnętrznych (4 pkt)

Mamy trzy pliki typu json. Pliki zawierają kolekcje obiektów opisanych za pomocą kategorii (dostępne są kategorie produktów przemysłowych i spożywczych) oraz nazwy konkretnych produktów. W obu plikach mogą pojawić się te same kategorie jak również produkty czyli dane mogą się powtarzać. Nie jest wymagane aby każda kategoria miała przypisany produkt. Niech lista kategorii liczy przynajmniej 8 pozycji.

Przykładowe pliki : Plik produktyA.json zawiera:

```
„Samochody” : [
  „nazwa”: „Fiat”,
  „nazwa”: „BMW”,
  ....
]
„Hulajnogi” : [ ]
„Słodycze” : [
  „nazwa”: „Czekolada”,
  „nazwa”: „Cukierki”,
  ....
]
„Soki” : [
  „nazwa”: „Jupi”,
  „nazwa”: „Kubuś”,
  „nazwa”: „Hoop”,
  ....
]
....
```

Plik produktyB.json zawiera:

```
Plik produktyB.json zawiera:
„Samochody” : [
  „nazwa”: „Fiat”,
  „nazwa”: „Mercedes”,
  ....
]
„Rowery” : [
  „nazwa”: „Kross”,
  „nazwa”: „Merida”,
  ....
]
„Soki” : [
  „nazwa”: „Jupi”,
  „nazwa”: „Kubuś”,
  ....
]
....
```

Napisz aplikację, która odczyta dane z plików i na ich podstawie wyświetli dane w postaci rozwijalnej listy. Elementy na poszczególnych gałęziach nie mogą się powtarzać.

Pierwszy poziom listy to lista kategorii (tylko kategorie, które posiadają przynajmniej jeden produkt mają być wyświetlane). Drugi poziom to lista produktów (wartości produktów muszą być singletonami).

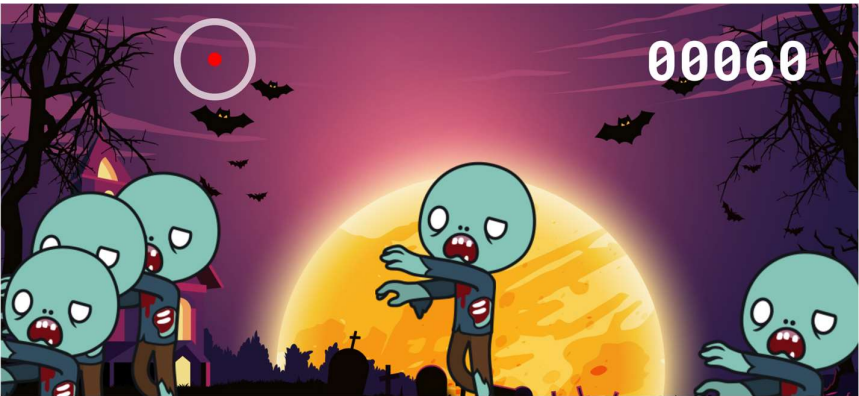
Przykładowo wyglądałby to tak jak na zdjęciu poniżej (podane wartości są tylko przykładowe i w żaden sposób nie dotyczą treści zadania)

Sekcja menu	Sekcja główna
<div><div><div>▼</div><div><div><div>147814298</div><div>299155077</div><div>868566152</div><div>1187187215</div><div>743084005</div></div></div></div><div><div>▼</div><div><div><div>1</div><div>1765323542</div></div></div></div><div><div>▼</div><div><div><div>248207181</div></div></div></div><div><div>></div><div><div><div>45</div><div>602344033</div></div></div></div><div><div>▼</div><div><div><div>699114243</div><div>-35562339</div></div></div></div></div>	<div>147814298, 743084005, 1, 1765323542</div>

Wybrane elementy listy niech wyświetlają się w sekcji głównej na stronie. W przypadku odznaczenia pozycji z listy rozwijalnej dana pozycja znika oczywiście natychmiast z sekcji głównej. Analogicznie jest przy zaznaczaniu – dołącza do pozycji w sekcji głównej

Zadanie 11. Farma Zoombie (6 punktów)

Twoim zadaniem jest implementacja gry widocznej na filmie (dołączonym w materiałach) i screenshocie poniżej.



W zakładce zadanie11 znajdziesz grafiki przydatne do realizacji gry.

Poniżej znajdziesz założenia gry i punktację

Wersja podstawowa, zawiera **(4 punkty)**

- podstawowy mechanizm strzelania do zombie – strzelamy za pomocą myszki (lewym klawiszem)
- zombie pojawiają się z prawej strony ekranu i poruszają się w stronę lewą
- zombie może pojawić się na różnej wysokości ekranu
- pojawienie się zombie jest generowane losowo
- szybkość poruszania jest również generowana losowo w pewnych przedziałach np. 1 – 5, gdzie 1 to standardowa szybkość a 5 to tryb turbo.
- wielkość zombie jest również losowa
- każdy zestrzelony zombie to 12 punktów wyświetlanych na ekranie w trybie on-line
- gra kończy się gdy trójka zombie dotrze do końca ekranu
- punktacja statyczna - za każdego zombie naliczane punkty (jak wyżej), za niecelny strzał ujemne punkty (-6pkt)

Ranking (highscores) **(2 punktów)**

- na wstępie gra prosi o podanie nicka (do skutku) - potem ten nick jest wyświetlany nad planszą do gry
- po zakończeniu gry pobieramy zapisane na serwerze wyniki. Można do tego celu użyć:
// highscore json blob: <https://jsonblob.com/> WSTAW SWOJ HASH JSON")
- dodajemy swój wynik, sortujemy, odcinamy pierwsze 7 wyników i zapisujemy znowu na serwerze
- ranking zawiera pozycję, nicka, ilość punktów i datę wpisu
- przy prezentacji highscores kursor zmienia się na zwykły
- pojawia się guzik umożliwiający powrót do nowej rozgrywki