

Programowanie aplikacji webowych

laboratorium 3

Cel zajęć:

Celem laboratorium jest przećwiczenie zagadnień związanych z tworzeniem stron RWD oraz poznanie właściwości dynamicznych wprowadzanych przez CSS3.

Zadania na rozgrzewkę (bez oceniania).

Zadanie 0. Tworzenie elementów responsywnych w sposób natywny.
W katalogu zad0 znajdziesz dwie przykładowe strony które należy wykorzystać do wykonania poniższego zadania.

Tworzenie strony responsywnej

Przykład użycia media media query, która zwraca ustawienia dla kontekstu strony gdy szerokość urządzenia jest mniejsza lub równa 100px:

```
@media (max-width: 100px) { /* CSS Rules */ }
```

A to przykład gdy wysokość urządzenia jest większa lub równa 350px:

```
@media (min-height: 350px) { /* CSS Rules */ }
```

Zmodyfikuj plik 1_MediaQuery.html taka by po osiągnięciu przez urządzenie na którym wyświetlamy stronę szerokość mniejszej lub równej 800px tekst w akapitach miał ustawione font-size na 10px oraz kolor czcionki zmienił się na czerwony.

Tworzenie zdjęcia responsywnego

Aby stworzyć zdjęcie responsywne należy ustawić mu następujące ustawienia

```
img {  
  max-width: 100%;  
  display: block;  
  height: auto;  
}
```

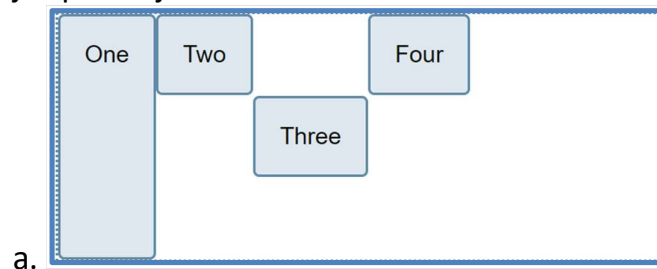
Zmodyfikuj plik 2_ResponsywnyImage.html taka by znajdujące się tam zdjęcie stało sie responsywne. Sprawdź jakie efekty uzyskasz.

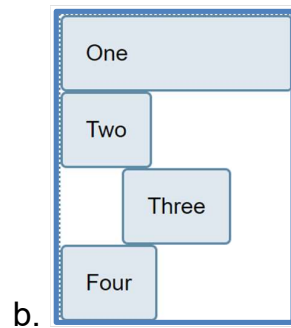
Zad.1. Stwórz szablon strony taki jak na rysunku poniższym (możesz wykorzystać szablon jaki tworzyłeś/aś w lab1.) Następnie wykorzystując stylowanie warunkowe pozwól aby przy zmianie rozdzielczości



układ 3 kolumnowy zmienił się na układ 2 kolumny (kolumna prawa ma być przesunięta poniżej) a następnie w układ 1 kolumnowy. Przy układzie jedno kolumnowym powinno zniknąć logo z sekcji nagłówkowej.

Zad.2. Jedna z najważniejszych właściwości w CSS jest FlexBox. Używając tej właściwości wystyluj stronę index.html z katalogu zdanie 2 tak aby wyglądała jak poniżej:



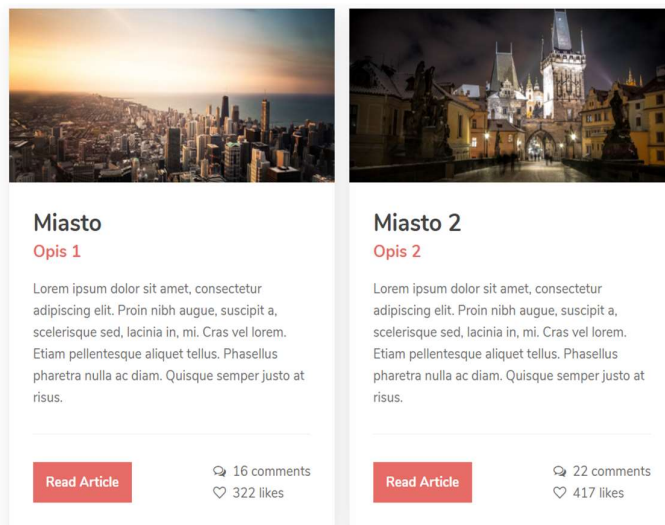


Zadania punktowane. (max 20 pkt)

Wymaganie ogólne dotyczące wszystkich zadań: uwzględnienie responsywności tzn. rozwiązania muszą zawsze wyglądać i zachować się „dobrze”. Zadania bez wykorzystania JS. Zastosowanie poprawnych znaczników semantycznych.

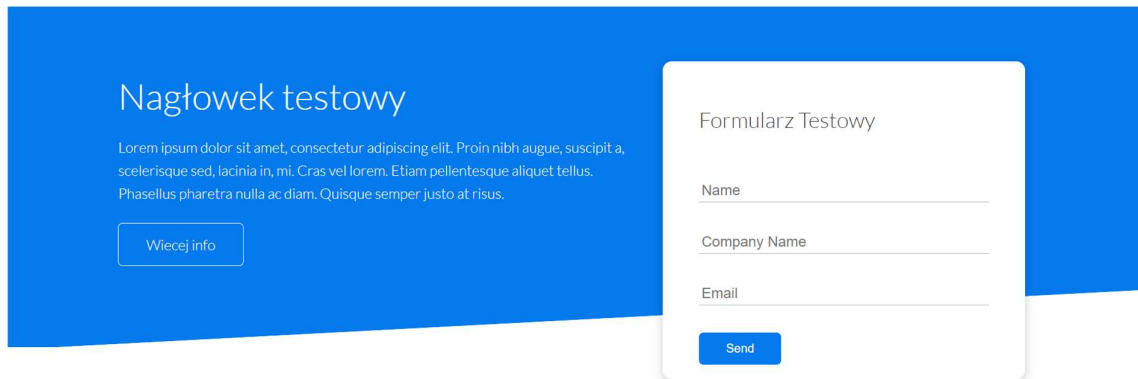
Sekcja podstawowa (max 10 pkt)

Zad 3. (2 pkt) Zaprojektuj karty miast, gdzie pojedyncze miasto opisane jest tak jak na rysunku poniżej. Stwórz galerię 10 kart i zaprojektuj je tak aby poprawnie zachowywały się w dowolnej rozdzielczości. Niech każda karta miasta będzie posiadała cieniowanie i zaokrąglone narożniki. (1pkt)



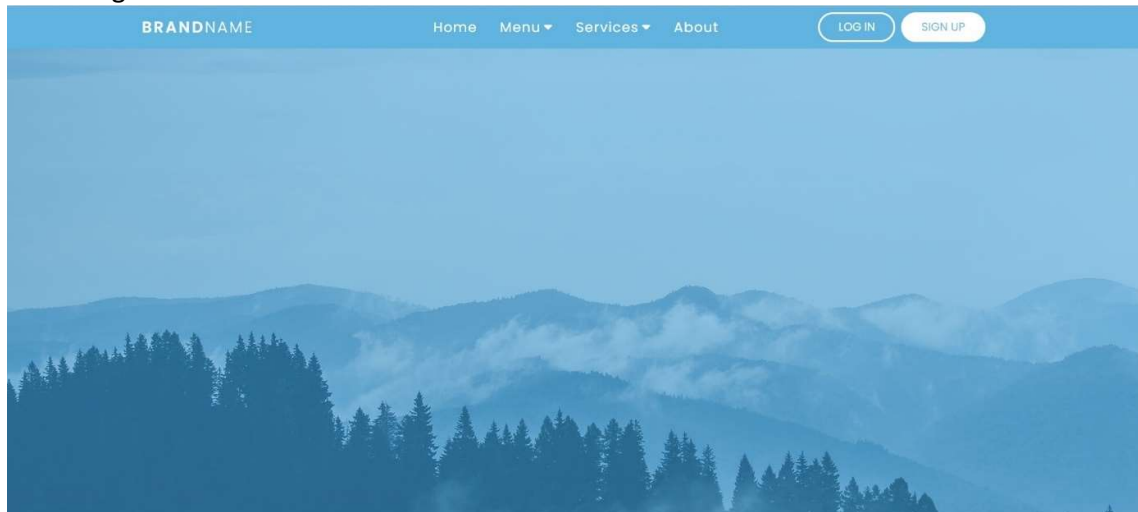
Przy zmianie rozdzielczości zmieniaj źródła zdjęć na wersje odpowiednie do aktualnej rozdzielczości (niech ta będzie inne zdjęcia tego miasta – oczywiście dopasowane pod kątem rozmiaru do rozdzielczości). Zastosuj przynajmniej 3 punkty przejścia. (1pkt)
Wskazówka. Zastosuj atrybut srcset do ładowania odpowiedniej wersji zdjęcia.
Poniżej odpowiedniej rozdzielczości pamiętaj, aby przejść na układ 1 kolumnowy

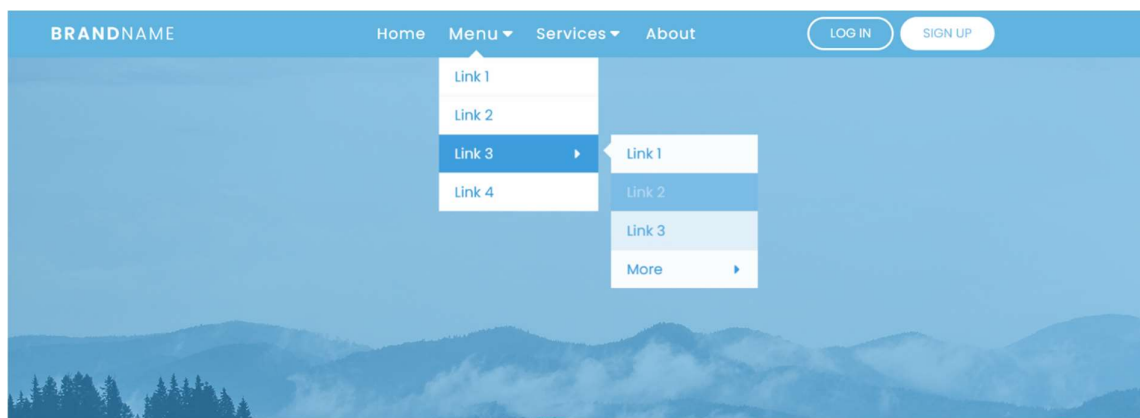
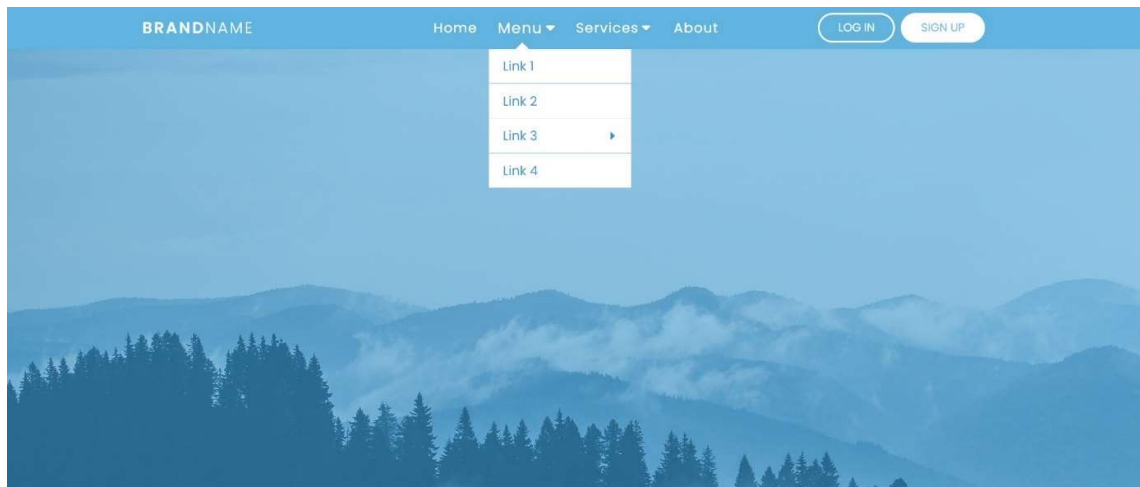
Zad 4. (2 pkt) Zaimplementuj nagłówek tak aby wyglądał jak na rysunku poniżej wróć uwagę na dolną krawędź niebieskiego tła – jest niesymetryczna. Elementy tekstowe oraz formularz powinny się nasunąć z boku na nagłówek. (1pkt)



Wizualizacje efektu o który mi chodzi znajdziesz w filmiku GR_4.
W przypadku zmiany rozdzielczości na poniżej 600px przejdź na układ 1 kolumnowy. Tekst niech wjeżdża wtedy od góry a formularz od dołu. (1pkt)

Zad 5. (2 pkt) Stwórz nagłówek wyglądający jak poniżej i zachowujący się jak poniżej. Niech menu rozwija się w sytuacji gdy najedziesz na niego myszką. Oryginalny rysunek tła znajdziesz w katalogu zadanie5.

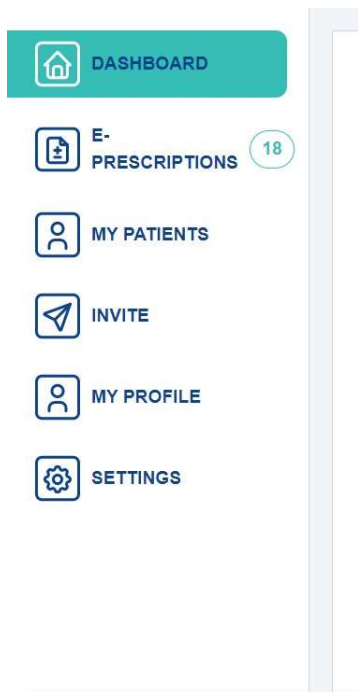




Oryginalny rysunek:

Punktacja: 2 pkt – menu rozwijalne wielopoziomowe + układ zgodny z projektem
1 pkt - „zdjecie za mgłą”

Zad 6. (1 pkt) Stwórz menu wertykalne jak na zdjęciach poniżej:



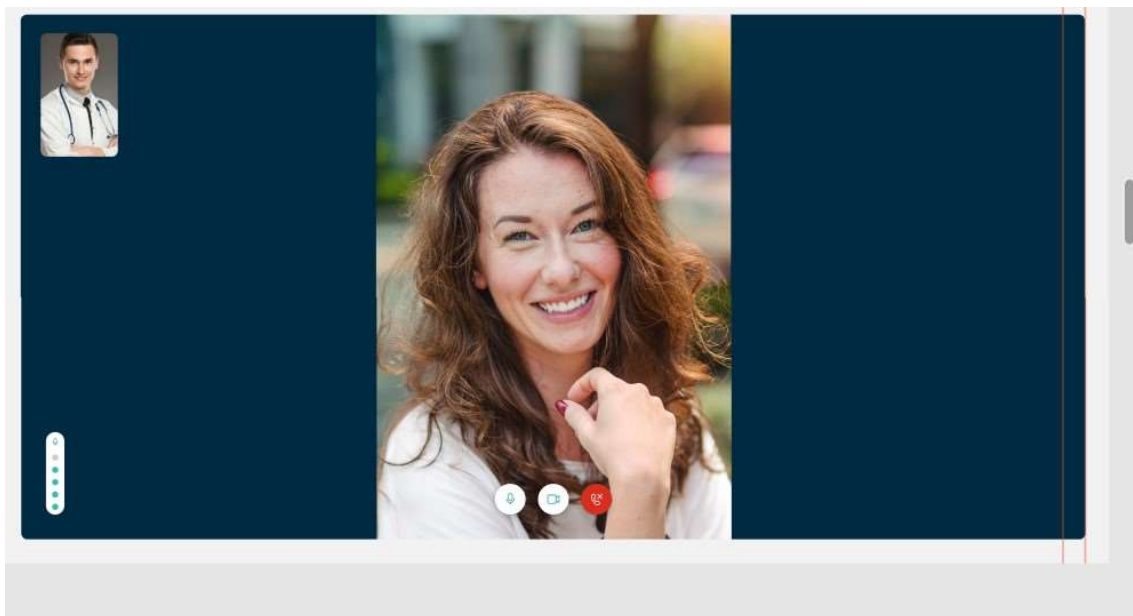
Zawartość Panelu z prawej strony niema znaczenia.

Niech przy zmianie rozdzielczości menu zmienia się jak poniżej. Przejście powinno odbyć się z wykorzystaniem odpowiednich tranzycji.



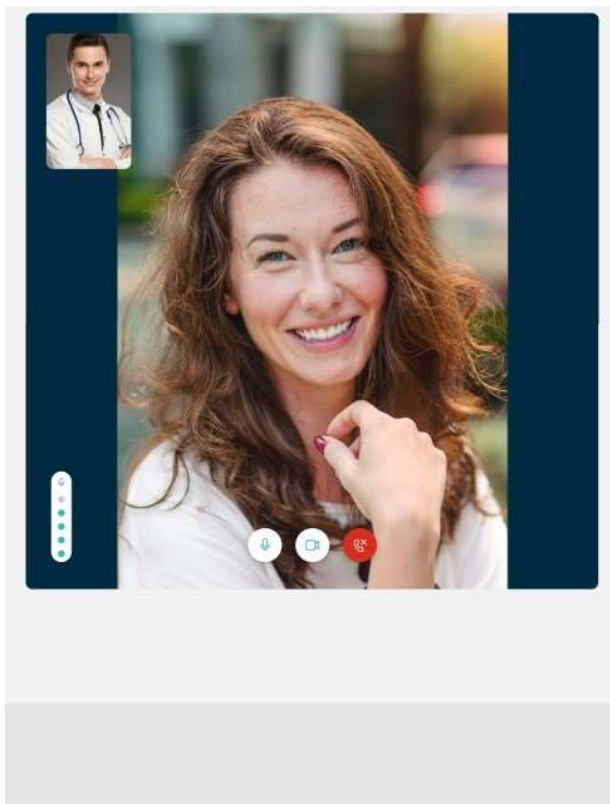
Zad 7. (1 pkt) Stwórz element który wygląda jak na zdjęciu poniżej.

Przy pełnej szerokości ekranu:

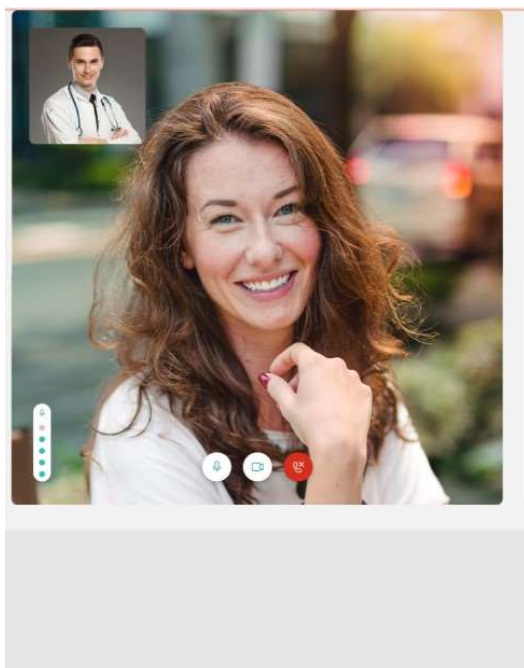


Wysokość strony jest większa niż sam zaprojektowany element, tak aby można było skrolować w dół.

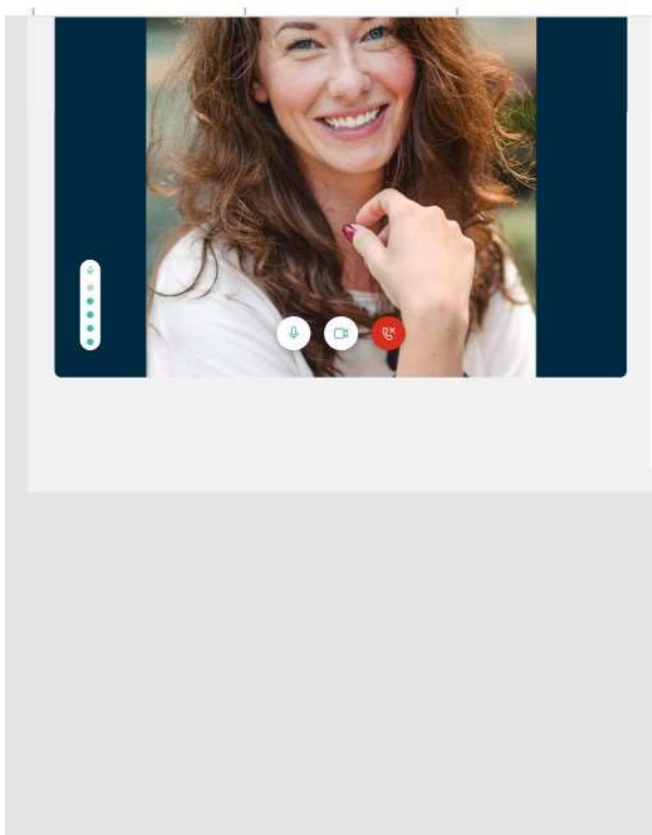
Przy zmianie szerokości viewport:



Przy jeszcze mniejszej szerokości ekran wygląda jak poniżej.



Podczas skrolowania w dół elementy sterujące -> zawsze trzymają się jednego miejsca na ekranie videoCall. Sam ekran VideoCall znika z ekranu wraz z zawartymi w nim elementami.



Zad 8. (2 pkt) Stwórz stronę na której umieścisz trzy zdjęcia oraz przykładowe teksty dokładnie tak jak na rysunku wzorcowym poniżej:



Najeżdżając myszką na dowolne zdjęcie – ma ono znaleźć się na pierwszym tle z dodatkowa obwoluta – tak jak na rysunkach poniżej:



Przy zmianie wielkości viewport strona powinna się tak skomponować aby dalej być czytelna i przejrzysta. Może to być np. tak.

EKSCYTUJĄCE MIEJSCA

TEKST 1

Lorem ipsum dolor sit amet consectetur adipisicing elit. Aperiam, ipsum sapiente aspernatur libero repellat quis consequatur ducimus quam nisi exercitationem omnis earum qui.

TEKST 2

Lorem ipsum dolor sit amet consectetur adipisicing elit. Asperiores nulla deserunt voluptatum nam.

[Wiecej →](#)



Sekcja rozszerzająca (10 pkt)

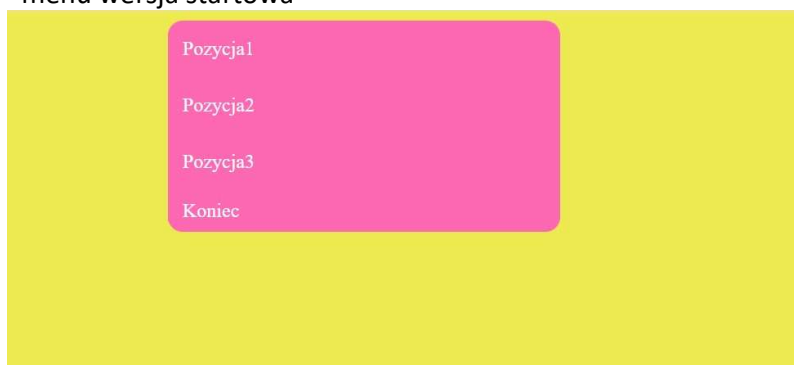
Zad 9. (3 pkt)

Napisz slajder zdjęć przesuwających się w poziomie na podstawie kliknięcia w wybrane zdjęcie.

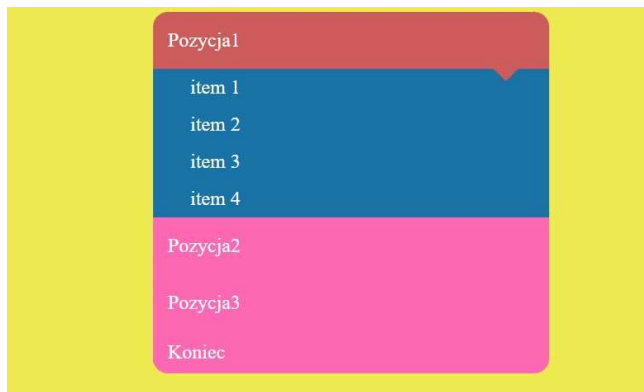
Filmik z demonstracji działania w katalogu zadanie 9.

Zad 10. (3 pkt)

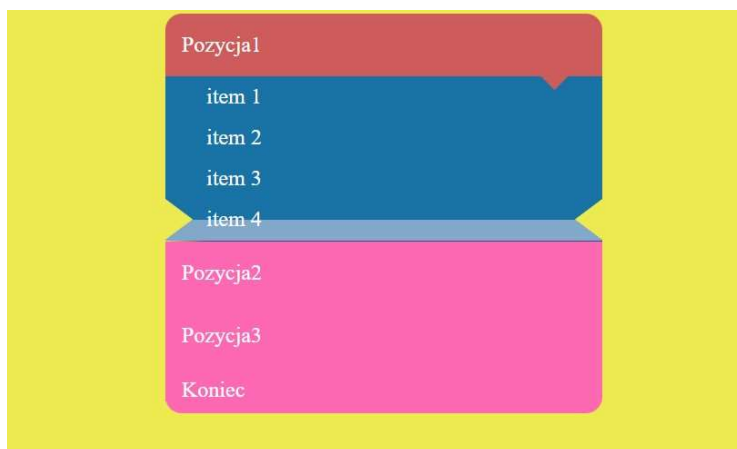
Stwórz menu pionowe, które wygląda i zachowuje się jak pokazano poniżej:
menu wersja startowa



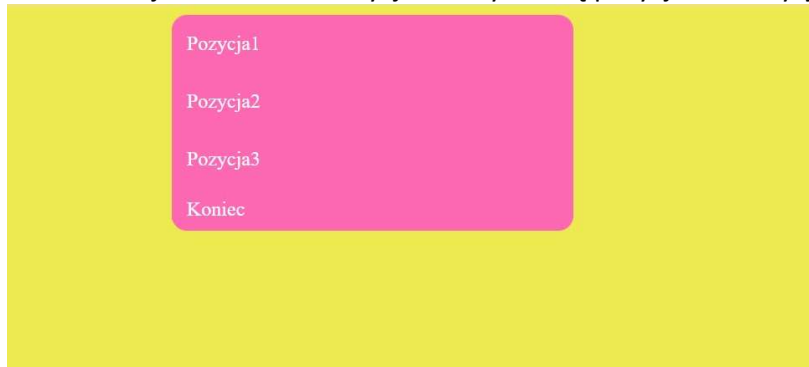
W menu kliknięto Pozycja1 -> następuje rozsuniecie sekcji Pozycja1 (rozsuniecie trwa kilka chwil)



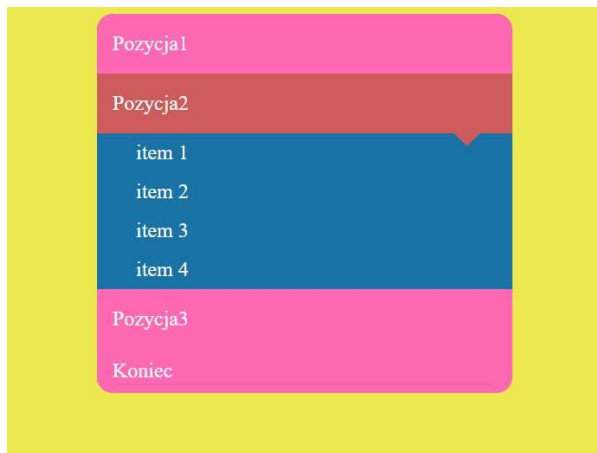
Użytkownik ustawia myszkę na pozycje Pozycja 2 – i nic się nie dzieje. Ciągłe widzimy ekran jak wyżej. Użytkownik wraca na zaznacza myszka pozycje item 4. Wynik ekran jak poniżej:



Naciśnięcie jeszcze raz na Pozycje1 zamyka listę pozycje dla Pozycja1 -> powrót do

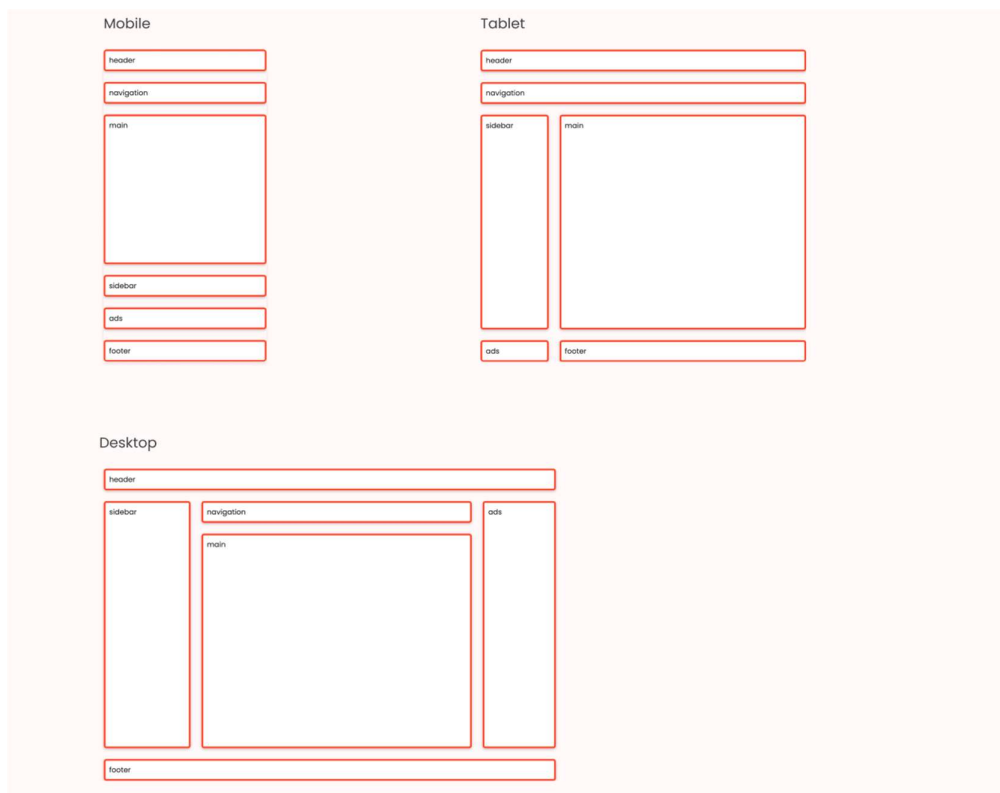


Ewentualnie naciśnięcie podczas otwarcia Pozycje1 na inna pozycje główna np. Pozycja2 -> zamyka Pozycja1 i rozsuwa Pozycja2 -> patrz poniżej.

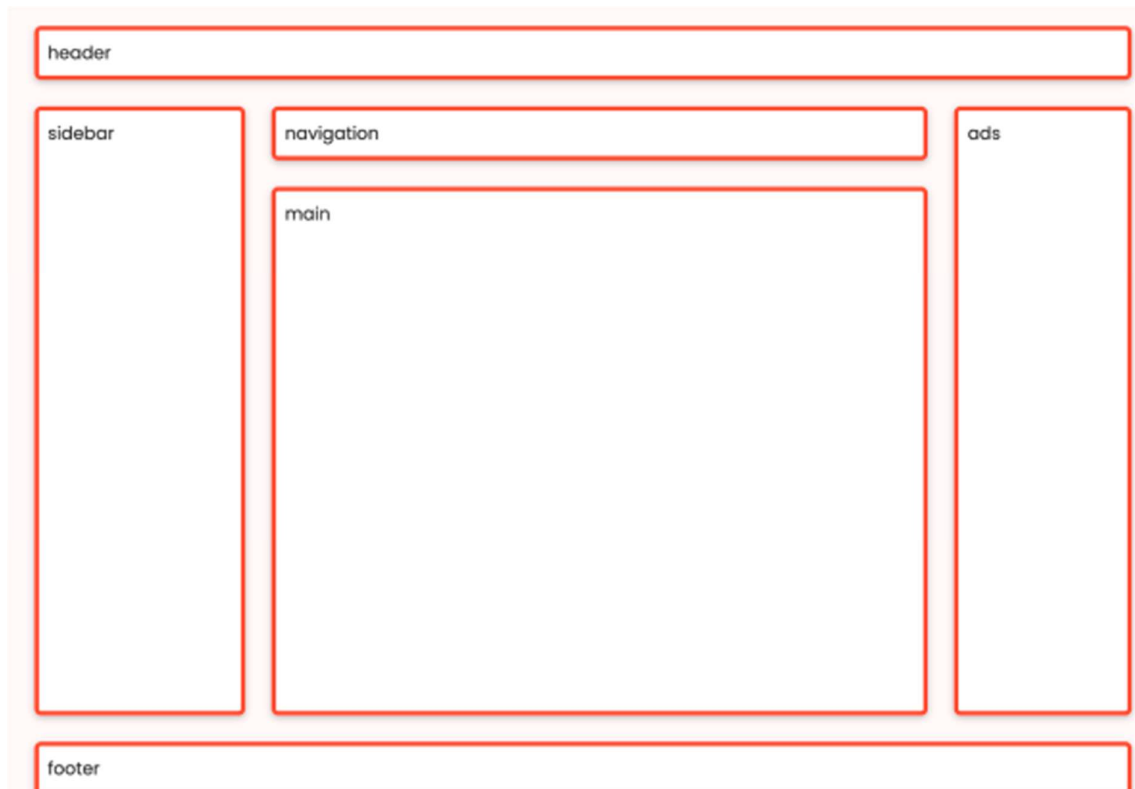


Rozwiązanie tylko za pomocą HTML i CSS. Nie używamy JS. Proszę zainteresować się pseudo-klasą `::after` oraz `::before`. Będą niezbędne do realizacji tego zadania.

Zad 11. (4 pkt) Zaprojektuj szablon strony, który w zależności od wyświetlanego urządzenia będzie wyglądał następująco:

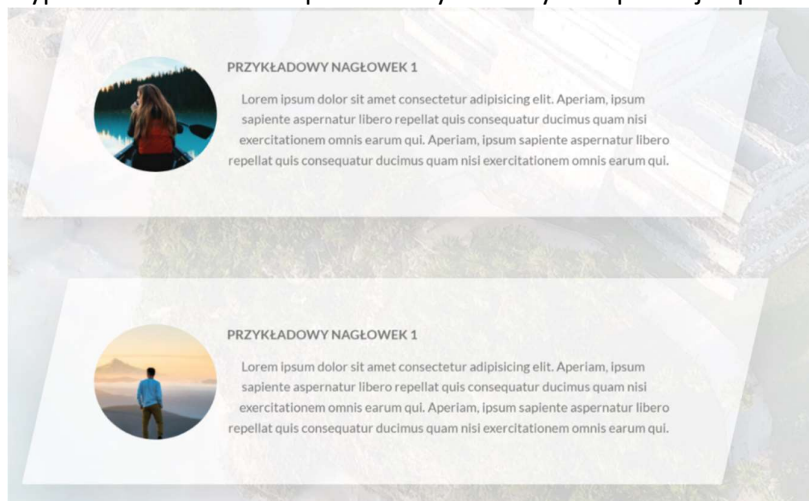


Layout dla wersji desktopowej (w powiększeniu) wygląda następująco:

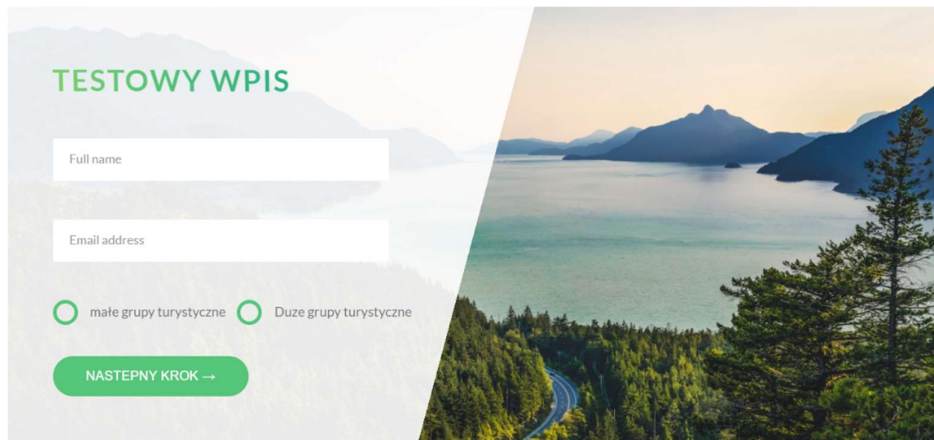


Niech sekcja sidebar zawiera Twoja wizytówkę: (w postaci elementów leżących jeden pod drugim)
zdziecie typu avatar,
imie i nazwisko,
full stack developer
nr telefonu
linki do kont w portalach społecznościowych (ikony)

Niech wizytówka zawsze wyświetla się na ekranie – nawet gdy będziemy przewijać całą stronkę. W tym celu sekcja main musi mieć przynajmniej 2 wysokości viewport. Należy ją wypełnić elementami w postaci listy testowych wpisów jak poniżej (np. 5 takich wpisów)



oraz formularza rezerwacyjnego jak poniżej.



TESTOWY WPIS

Full name

Email address

☐ małe grupy turystyczne ☐ Duże grupy turystyczne

NASTEPNY KROK →

Pozostałe sekcje wypełnij: sekcja reklam (3 różnymi zdjęciami z podpiętymi aktywnymi linkami), menu: 4 dowolne pozycje horyzontalne.

Zasady oceniania:

1 pkt – stworzenie layout responsywnego

1 pkt - stworzenie sidebar i jego „unieruchomienie”

1 pkt – stworzenie listy wpisów wraz z zdjęciem i oblewaniem (tak jak w przykładzie)

1 pkt – stworzenie formularza + nawigacja doklejająca się do górnego ekranu