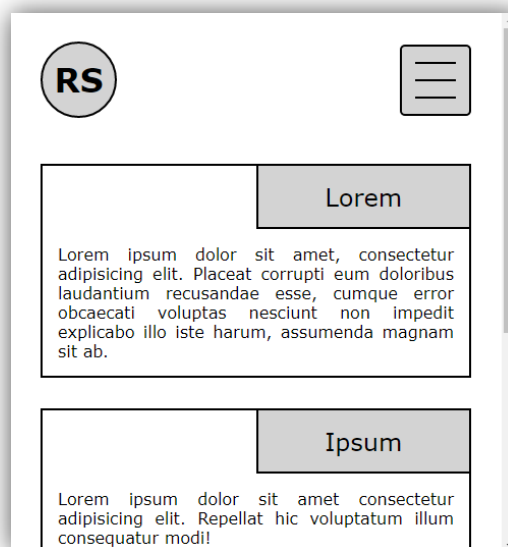
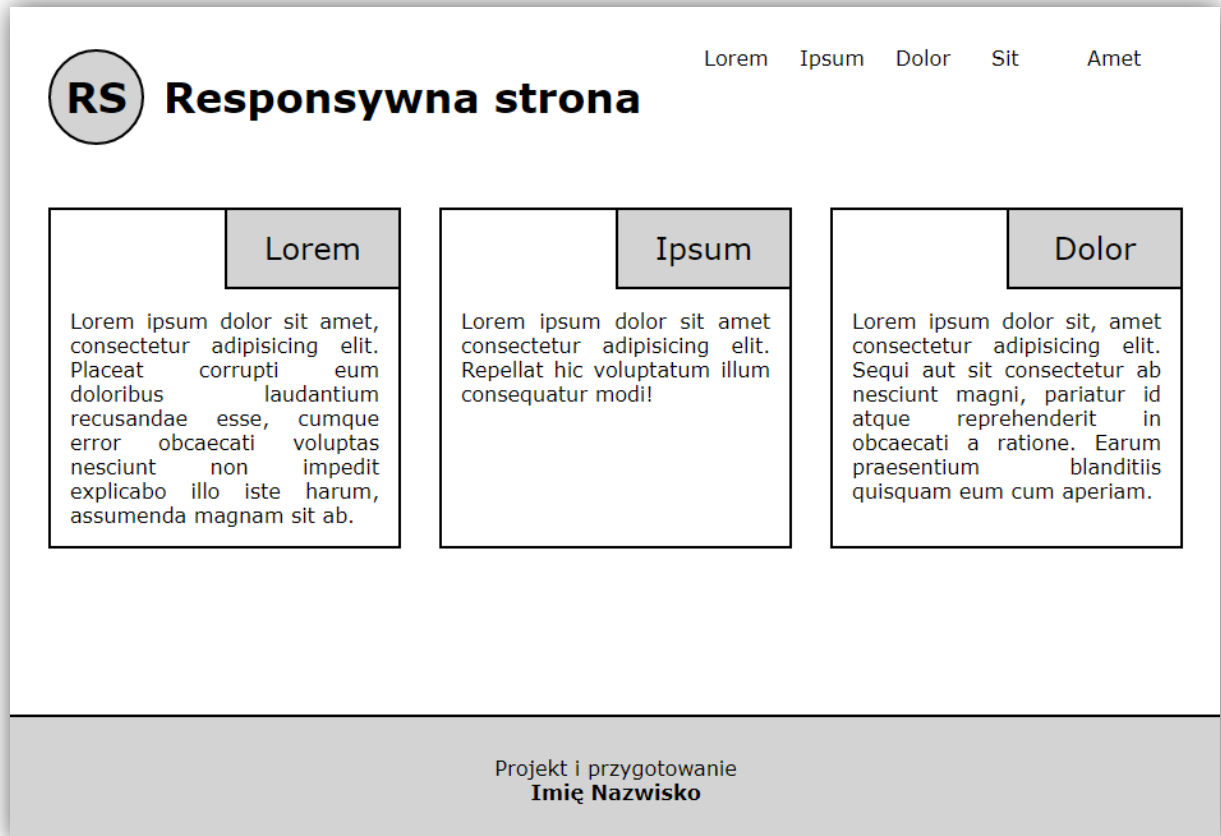


## Responsive design

Celem zadania jest zapoznanie studenta z koncepcją responsywnych stron internetowych. W ramach zadania zostanie stworzona strona automatycznie dostosowująca się do dwóch rozmiarów urządzeń: desktopowych oraz mobilnych. Gotowa strona w dwóch wariantach będzie wyglądała tak jak na poniższych rysunkach.



1. Zaczynij od przygotowania samej struktury strony, czyli dokumentu html. Postaraj się (sensownie) wykorzystać jak najwięcej znaczników semantycznych. Na pierwszy rzut oka widać, że strona jest podzielona na trzy główne sekcje: nagłówek, zawartość oraz stopkę, które można zawrzeć odpowiednio w znacznikach **header**, **main** oraz **footer**. W nagłówku jest pole z napisem RS, które posłuży jako logo, dalej znajduje się tytuł strony oraz sekcja nawigacyjna (**nav**) składająca się z linków. Sekcja główna składa się z trzech artykułów (**article**), z których każdy ma nagłówek oraz akapit tekstu. Stopka może składać się np. z dwóch akapitów. Po uruchomieniu strona bez żadnych stylów powinna wyglądać mniej więcej tak jak na poniższym zrzucie ekranu.

RS

## Responsywna strona

[Lorem Ipsum Dolor Sit Amet](#)

Lorem

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Placeat corrupti eum doloribus laudantium recusandae esse, cumque error obcaecati voluptas nesciunt non impedit explicabo illo iste harum, assumenda magnam sit ab.

Ipsum

Lorem ipsum dolor sit amet consectetur adipisicing elit. Repellat hic voluptatum illum consequatur modi! Illo repellat obcaecati, labore sit rem officia ratione similique, quam id a possimus maiores optio ipsa.

Dolor

Lorem ipsum dolor sit, amet consectetur adipisicing elit. Sequi aut sit consectetur ab nesciunt magni, pariatur id atque reprehenderit in obcaecati a ratione. Earum praesentium blanditiis quisquam eum cum aperiam.

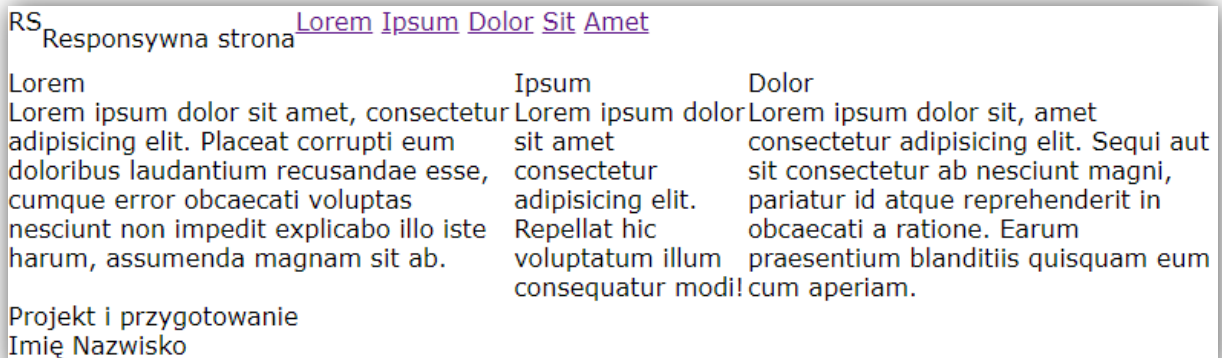
Projekt i przygotowanie

Imię Nazwisko

2. Zmianę wyglądu zaczniemy od zresetowania domyślnych marginesów (**margin**) i wypełnienia (**padding**) elementów **html**, **body** oraz **p**, ponieważ domyślnie mogą one być różne od 0, w zależności od przeglądarki. Utwórz osobny plik CSS, dołącz go do Twojej strony i wprowadź odpowiednią regułę.
3. Na początku warto również podjąć decyzję w jaki sposób będziemy definiowali wielkość elementów znajdujących się na stronie. Za to zadanie odpowiada w CSS właściwość **box-sizing**. Ustaw ją na jedną z wybranych przez Ciebie wartości dla wszystkich elementów w dokumencie. W wyborze może pomóc Tobie artykuł na stronie <https://css-tricks.com/box-sizing/>.
4. Ustaw wszystkim elementom czcionkę **Verdana** w rozmiarze **16px**. Wyłącz również domyślne pogrubienie (**font-weight**) i podkreślenie (**text-decoration**) elementów, a także ustaw kolor wszystkich elementów na czarny.
5. Aby odpowiednio rozmieścić elementy na stronie posłużymy się głównie mechanizmem układu CSS zwanym flexbox. Wejdź na stronę <https://css-tricks.com/snippets/css/a-guide-to-flexbox/> i zapoznaj się pokrótce z zamieszczonym tam przewodnikiem. Dla lepszego zrozumienia i zapamiętania zaprezentowanych tam treści możesz również przejść grę znajdującą się na stronie <https://flexboxfroggy.com/#pl>, która w przystępny sposób zapozna Cię z głównymi mechanizmami flexbox'a. Zacznijmy od zapewnienia poprawnego układu w trzech głównych

Podstawy aplikacji internetowych – laboratorium  
Responsive design

sekcjach naszej strony (nagłówek, treść, stopka). Korzystając z właściwości **display** oraz **flex-direction** ustaw parametry każdej z tych sekcji. Po tym zabiegu strona powinna wyglądać mniej więcej tak jak na poniższym zrzucie ekranu.



6. Zaczniemy teraz sukcesywnie dostosowywać do naszych potrzeb każdą sekcję po kolei, od góry do dołu, zaczynając od nagłówka. Zacznij od dodania wypełnienia do nagłówka w wielkości **30px** oraz wyrównania elementów w pionie (korzystając z jednej właściwości flexbox'a). Po tym zabiegu nagłówek powinien wyglądać tak.

RS Responsywna strona [Lorem Ipsum Dolor Sit Amet](#)

7. Dla tytułu strony (napis „Responsywna strona”) w nagłówku ustaw lewy margines wielkości **15px** oraz pogrubioną czcionkę w rozmiarze dwa razy większym niż na całej stronie.

RS **Responsywna strona** [Lorem Ipsum Dolor Sit Amet](#)

8. Spraw, aby sekcja z odnośnikami (**nav**) była (jeśli to możliwe) 2 razy większa niż sekcje z logo oraz z tytułem strony (**flex-grow**) a także przesun wszystkie odnośniki w tej sekcji do górnej krawędzi sekcji (**align-self**). W sekcji **nav** również włącz mechanizm flexbox'a (**display**) a następnie przesun wszystkie odnośniki na jej prawą stronę (**justify-content**).

RS **Responsywna strona**

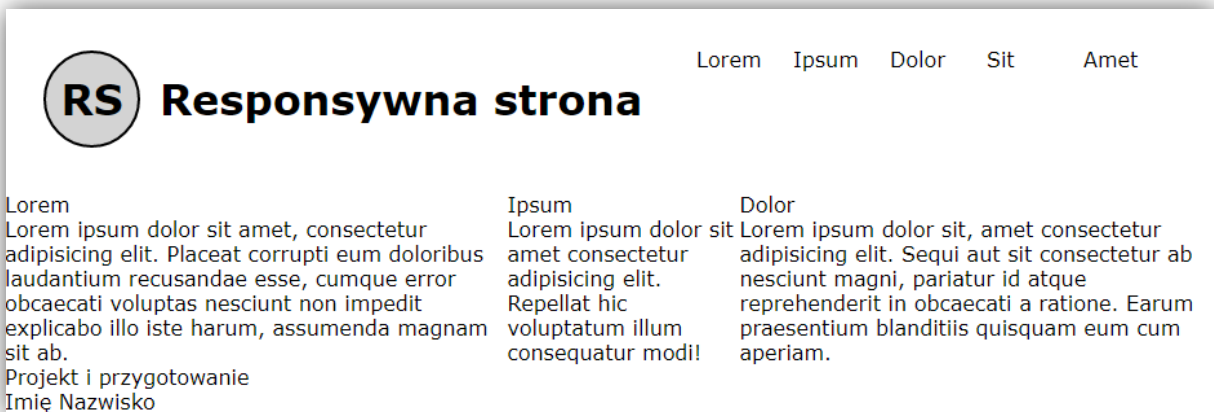
[Lorem](#)[Ipsum](#)[Dolor](#)[Sit](#)[Amet](#)

9. Ustaw szerokość każdego odnośnika na 75px.

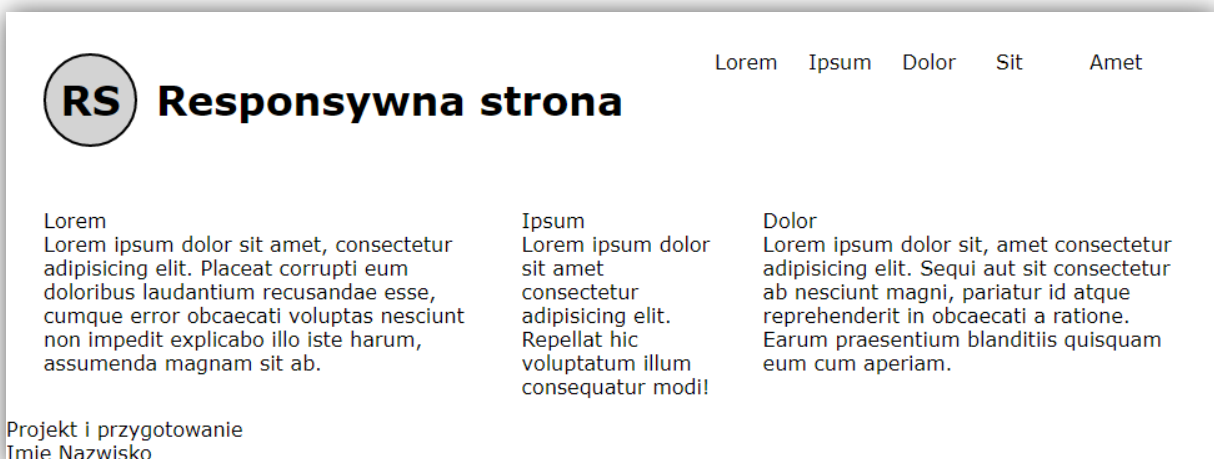


10. W miejscu logo (zamiast napisu „RS”) wstaw obrazek lub samodzielnie stwórz w CSS logo o wymiarach 75px na 75px.

11. Po powyższych zabiegach cała strona powinna wyglądać mniej więcej następująco.

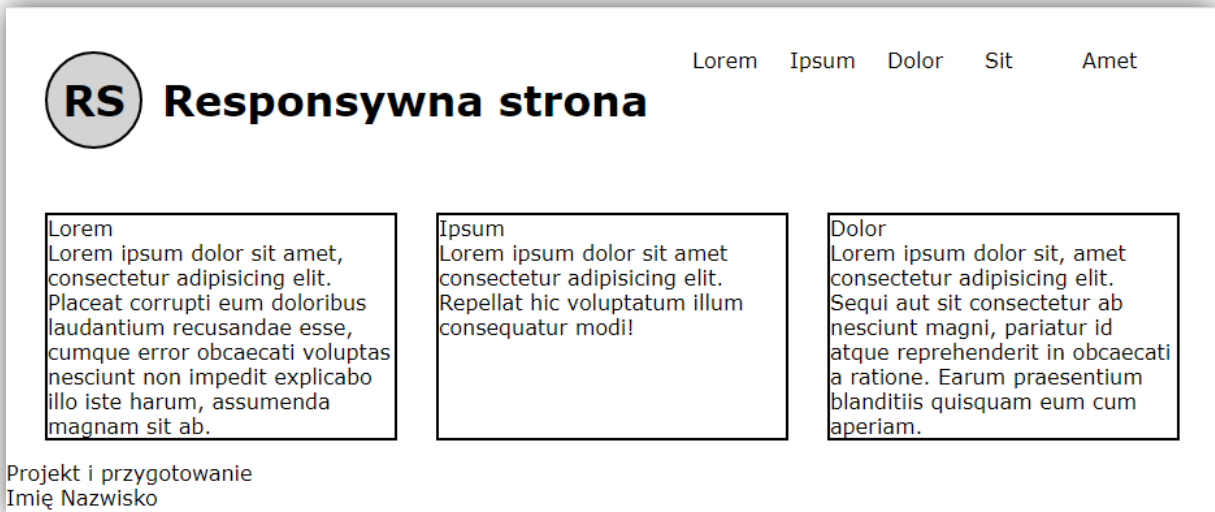


12. Teraz zabierzemy się za sekcję główną. Chcemy, żeby odstępy od artykułów do krawędzi, jak również odstępy między samymi artykułami, wynosiły tyle samo ile w nagłówku, czyli 30px. Efekt ten osiągniemy składając wypełnienie sekcji głównej oraz marginesy artykułów. Ustaw obie te wartości na 15px, co łącznie da nam 30px.

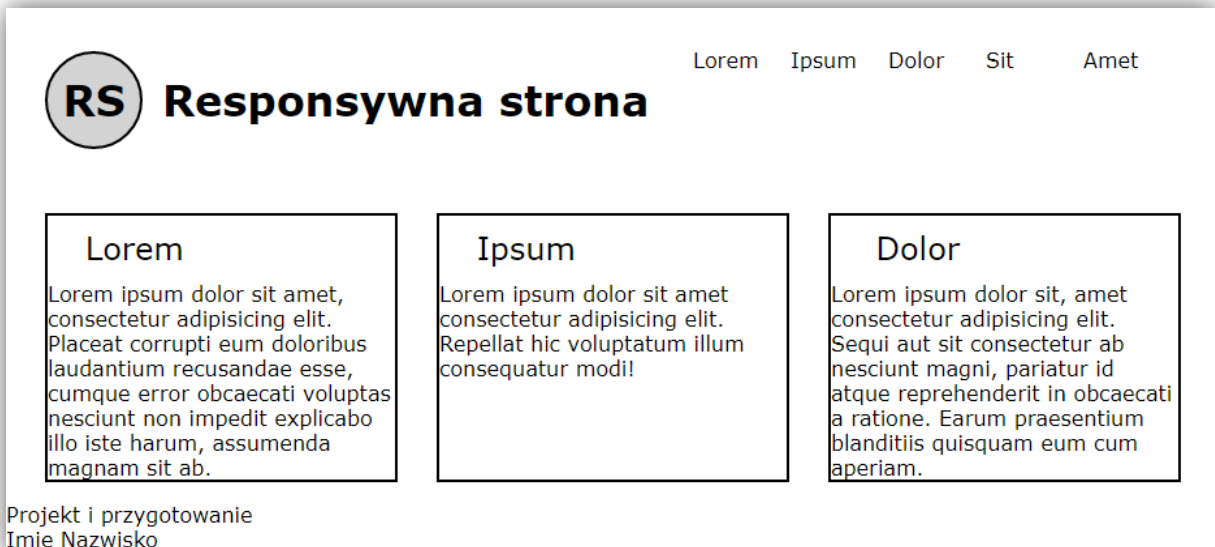


Podstawy aplikacji internetowych – laboratorium  
Responsive design

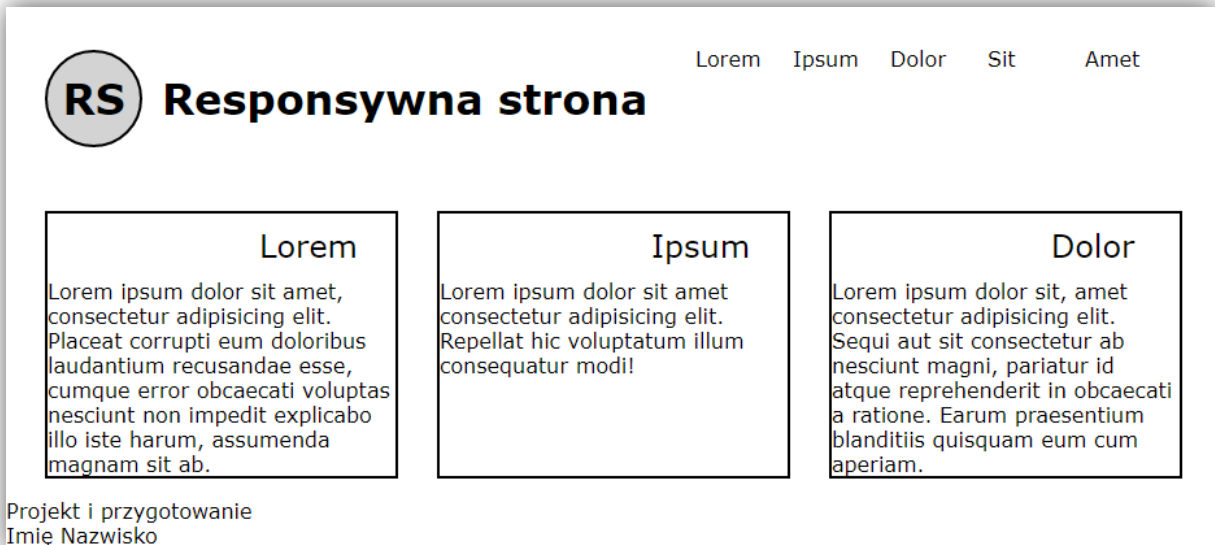
13. Aby każdy artykuł zajmował tyle samo miejsca, niezależnie od ilości zawartego w nim tekstu, ustaw szerokość każdego artykułu na **100%**, a następnie dodaj do artykułów czarną ramkę o grubości **2px**.



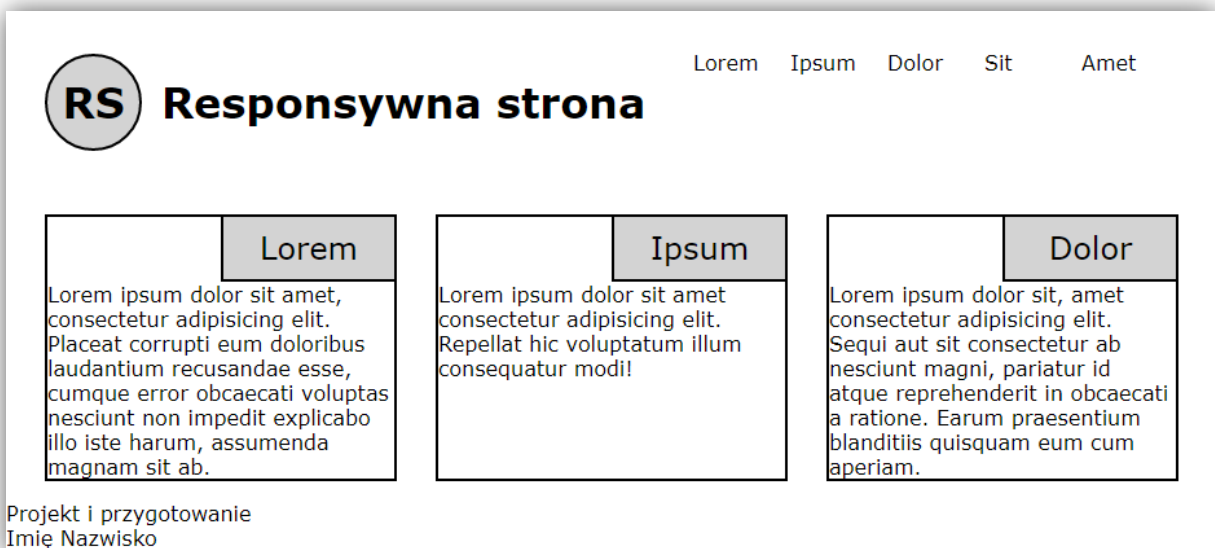
14. Zajmiemy się teraz nagłówkami w artykułach. Zaczniij od ustawienia każdemu nagłówkowi wysokości **50px**, szerokości **50%** oraz rozmiaru czcionki **1.5em**, a następnie wycentruj tekst w każdym nagłówku (**display**, **justify-content**, **align-items**).



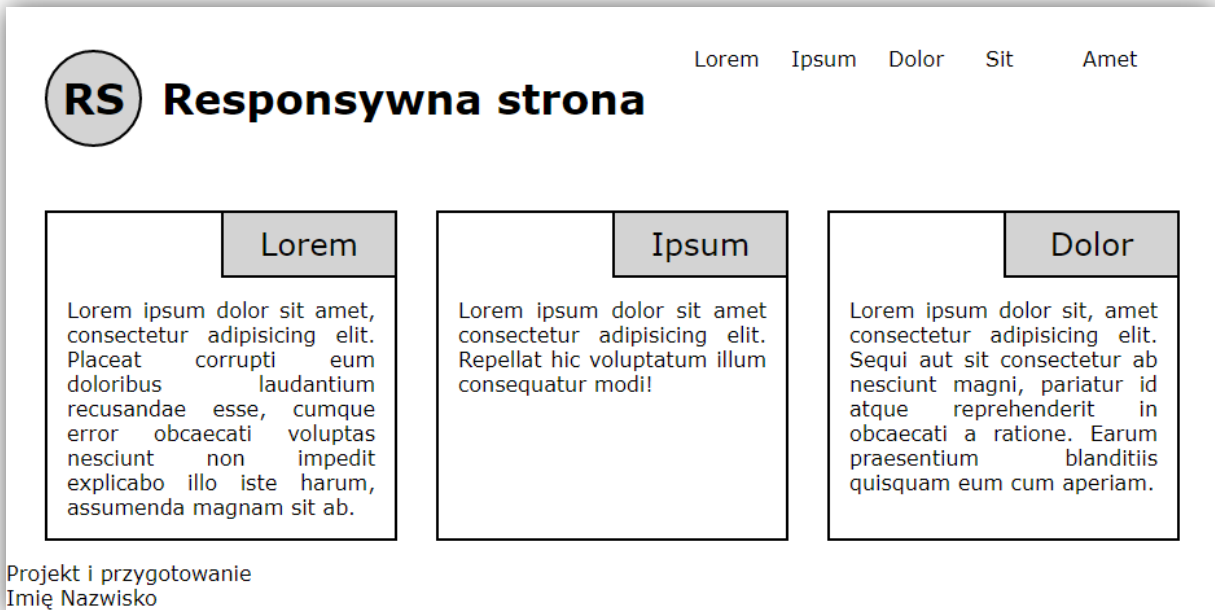
15. Teraz wyrównamy nagłówki do prawej strony. Można to osiągnąć na wiele sposobów, jednak w myśl zasady, że jak się ma dobry młotek to wszystko wygląda jak gwóźdź, my posłużymy się oczywiście flexbox'em. Zacznij od włączenia układu flexbox i ustawienia kierunku kolumnowego w artykułach, a następnie w nagłówkach ustaw odpowiednie wyrównanie (**align-self**).



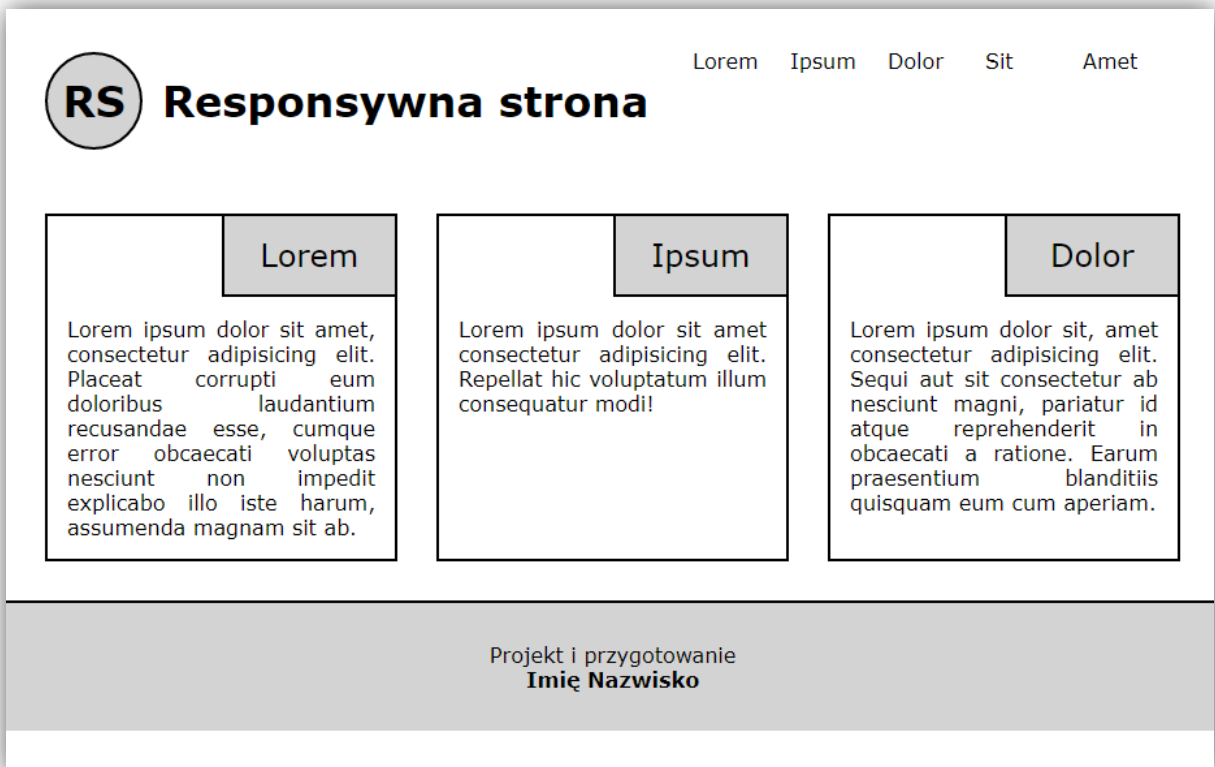
16. Teraz pozostaje tylko ustawić w nagłówkach jasnoszare tło oraz lewą i dolną czarną ramkę o grubości 2px.



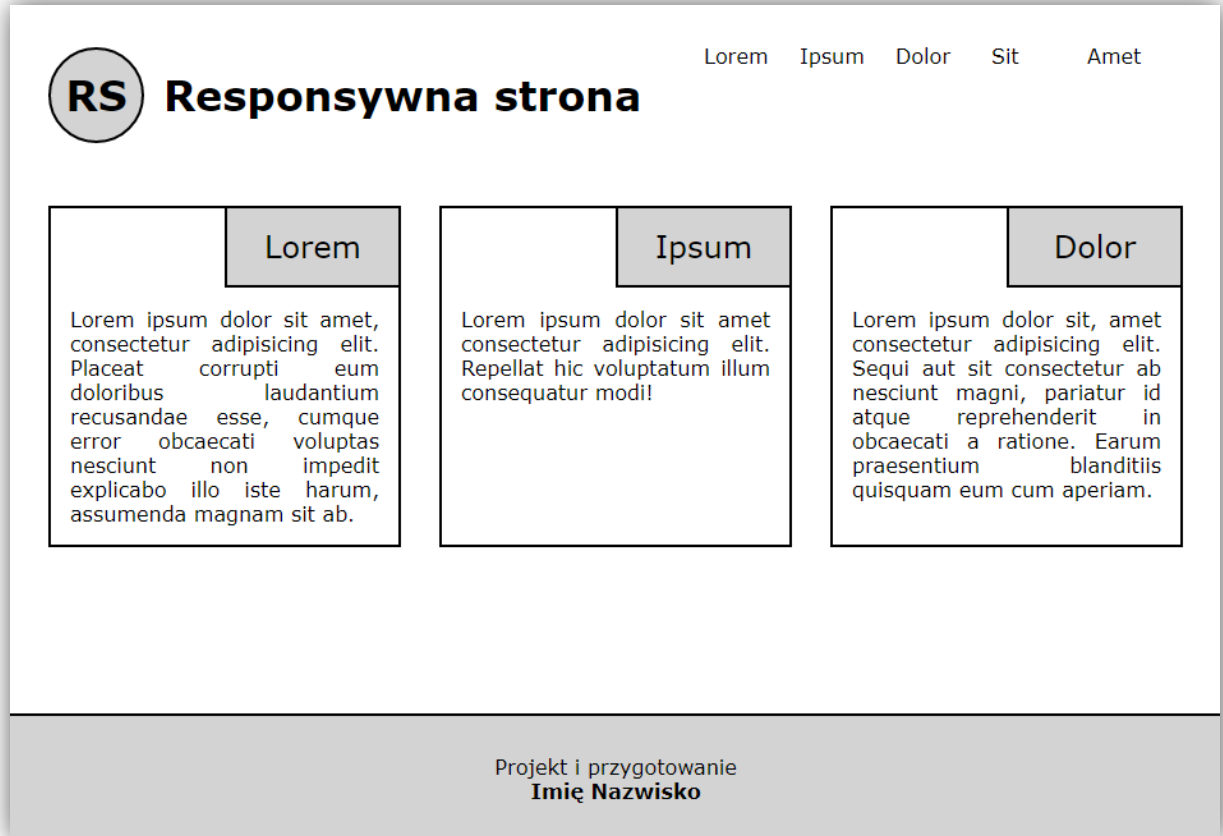
17. Ostatnim zabiegiem pozostaje wyjustowanie akapitów w artykułach (**text-align**) oraz ustawienie im wypełnienia o wielkości **15px**.



18. Teraz pozostaje nam jedynie zająć się stopką. Tutaj sprawa jest już relatywnie prosta. Ustaw stopce stałą wysokość **100px**, jasnoszare tło, czarną górną ramkę o grubości **2px** i górny margines o wysokości **15px**, a następnie wycentruj zawartość w pionie i w poziomie (**align-items**, **justify-content**). Dodatkowo, ustaw drugiemu akapitowi w stopce pogrubioną czcionkę. Efekt powinien być następujący.



19. Jest już prawie dobrze, ale pozostaje jeszcze rozwiązać kwestię pozycji stopki, która powinna być zawsze przyklejona do dołu strony (tzw. *sticky footer*). Jak ze wszystkim innym w CSS i tutaj jest wiele opcji rozwiązania tego problemu (<https://css-tricks.com/couple-takes-sticky-footer/>). Skorzystaj z jednej z nich (na tym etapie chyba najłatwiej będzie... flexbox'em). Zauważ, że niestety nie obejdziesz się bez dodatkowego elementu (**div** z klasą **content**), w którym zawrzemy wszystko poza stopką. Efekt końcowy powinien być następujący.



20. Na tym etapie mamy gotową stronę w wersji na duże ekrany. Teraz pozostaje nam obsłużyć małe ekrany. Aby móc definiować style w zależności od parametrów okna przeglądarki posłużymy się zapytaniami media w CSS. Zaczniemy od sekcji głównej, bo tutaj sprawa jest relatywnie prosta. Na końcu swojego arkusza stylów dodaj poniższą regułę i zobacz co dzieje się ze stroną kiedy przechodzisz przez próg 900px szerokości okna przeglądarki.

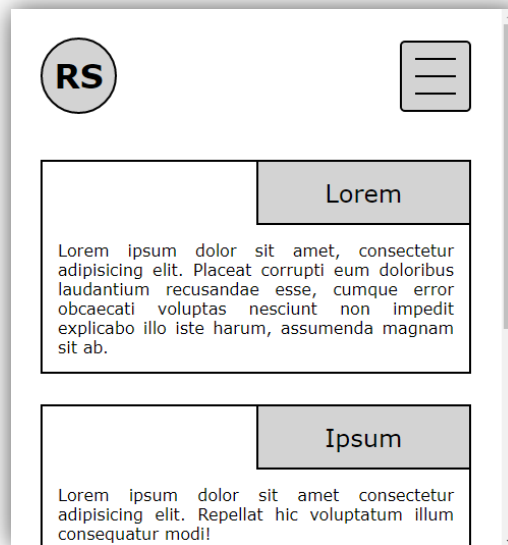
```
@media all and (max-width: 900px) {  
  main {  
    flex-wrap: wrap;  
  }  
}
```

21. Teraz dodaj do zapytania media reguły, które sprawią, że tytuł strony oraz sekcja nawigacyjna znikną na mniejszych urządzeniach (**display: none**), a w ich miejsce pojawi się tzw. „hamburger”, czyli ikona zwiniętego menu. Dodaj w sekcji nagłówkowej strony jako ostatni element „hamburgera” (możesz dodać obrazek lub spróbować stworzyć go samodzielnie z elementów HTML i reguł CSS) i ustaw go domyślnie jako niewyświetlany, a na mniejszych urządzeniach jako wyświetlany (**display: block**). Dodatkowo zmodyfikuj na mniejszych urządzeniach sposób rozmieszczenia elementów w sekcji nagłówkowej tak, aby logo pozostało



Podstawy aplikacji internetowych – laboratorium  
Responsive design

po lewej stronie strony, a hamburger był po prawej (**justify-content**). Efekt końcowy na małych urządzeniach powinien być następujący.



22. Gratulacje! To koniec. Oczywiście nasz hamburger nie działa, ale do tego potrzebowalibyśmy już języka JavaScript, a tego nauczymy się na następnych zajęciach...