

# HyperText Markup Language

## HTML & CSS

# Урок 12

## Bootstrap

# Оглавление

1. Что такое Bootstrap? .....	4
2. История создания Bootstrap .....	5
3. Цели и задачи Bootstrap.....	6
4. Подключение Bootstrap.....	7
5. Цветовая схема Bootstrap .....	11
6. Система сеток .....	13
7. Утилиты Bootstrap .....	23
Рамка .....	23
Тени .....	25
Поля и отступы .....	26
Изменение размеров .....	28
Отображение/Скрытие .....	29
8. Использование различных паттернов Bootstrap .....	33
Хлебные крошки .....	33
Карточки .....	34
Аккордеон.....	38

Карусель .....	41
Модальное окно .....	47
Пагинация .....	51
9. Навигация и Bootstrap .....	54
10. Скроллинг .....	70
11. Использование изображений и видео .....	73
12. Пример создания сайта с помощью фреймворка Bootstrap .....	77
Домашнее задание.....	98

Материалы урока прикреплены к данному PDF-файлу. Для до-  
ступа к ним урок необходимо открыть в программе [Adobe Acrobat  
Reader](#).

# 1. Что такое Bootstrap?

**Bootstrap** — это фреймворк HTML, CSS и JS, который предназначен для простой и быстрой верстки адаптивных сайтов. Фреймворком называют набор инструментов, используя который нужно соблюдать определенные правила и порядок.

Фреймворк Bootstrap включает в себя шаблоны оформления различных элементов страницы: форм, кнопок, навигации, всплывающих окон, типографики и так далее. Так же фреймворк предлагает гибкую сетку, используя которую можно достаточно легко получить адаптивный сайт для различных устройств.

## 2. История создания Bootstrap

Изначально Bootstrap разрабатывался как набор инструментов для компании Twitter и назывался Twitter Blueprint. В 2011 году фреймворк был открыт для всемирной сети под именем Bootstrap. С тех пор фреймворк развивался, с каждой новой версией добавлялись новые инструменты и технологии, и, на текущий момент, используется версия Bootstrap 4.\*.

Bootstrap широко распространен в среде разработчиков благодаря своей простоте в использовании и открытому исходному коду.

В июне 2020 года вышла alpha-версия Bootstrap 5. Основные изменения — прекращение использования jQuery и отсутствие поддержки Internet Explorer.

### 3. Цели и задачи Bootstrap

Цели и задачи, которые ставит перед собой фреймворк Bootstrap:

- высокая скорость разработки веб-страниц (достигается использованием готовых решений);
- кросбраузерность и кросплатформенность — одинаковое отображение и работа сайта во всех браузерах и операционных системах;
- адаптивность для различных устройств;
- не обязательное глубокое знание HTML, CSS и JS.

Как видно из перечисленного, Bootstrap имеет множество преимуществ, но в некоторых случаях его использование нецелесообразно. Например, когда нужно выполнить верстку уникального дизайна, так как фреймворк предлагает шаблонные решения.

# 4. Подключение Bootstrap

Подключение Bootstrap можно осуществить несколькими способами: скачав исходники файлов фреймворка с официального сайта [getbootstrap.com](https://getbootstrap.com) либо включить в проект файлы через cdn. **Content Delivery Network** — это сеть компьютеров, обеспечивающая быструю доставку контента. Сервера cdn расположены таким образом, чтобы время ответа посетителям сайта было минимальным. Это достигается тем, что cdn размещают копии контента на разных серверах по всему миру и направляют пользователя на ближайший из них.

Если скачать исходники файлов, последние будут находиться в вашем проекте и нет необходимости иметь постоянный доступ в интернет, с использованием cdn — не нужно ничего скачивать, но необходим постоянный доступ к сети.

## Compiled CSS and JS

Download ready-to-use compiled code for **Bootstrap v4.5.0** to easily drop into your project, which includes:

- Compiled and minified CSS bundles (see [CSS files comparison](#))
- Compiled and minified JavaScript plugins

This doesn't include documentation, source files, or any optional JavaScript dependencies (jQuery and Popper.js).

[Download](#)

Рисунок 1

Рассмотрим практическое применение этих способов. Проходим по ссылке [getbootstrap.com/docs/4.5/getting-started/download/](http://getbootstrap.com/docs/4.5/getting-started/download/) и скачиваем скомпилированные CSS (*bootstrap.min.css*) и JS (*bootstrap.min.js*) файлы, которые добавляем в наш проект (рис. 1).

Подключение файлов в *index.html*:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport"
        content="width=device-width
                initial-scale=1.0">
    <title>Bootstrap</title>
    <!-- подключение стилевого файла фреймворка
        bootstrap -->
    <link rel="stylesheet"
        href="css/bootstrap.min.css">
</head>

<body>
    <!-- основной код страницы -->

    <!-- подключение скриптового файла библиотеки
        jquery, необходимого для работы фреймворка -->
    <script src="https://code.jquery.com/
                jquery-3.5.1.slim.min.js">
    </script>

    <!-- подключение скриптового файла библиотеки
        popper, необходимого для работы
        фреймворка -->
    <script src="https://cdn.jsdelivr.net/npm/
                popper.js@1.16.0/dist/umd/popper.min.js">
    </script>
```

```
<!-- подключение скриптового файла фреймворка  
bootstrap -->  
<script src="js/bootstrap.min.js"></script>  
  
</body>  
</html>
```

Как видно из кода, скриптовые файлы библиотек подключаются удаленно. Если вы хотите получить полностью локальный проект, то необходимо скачать файлы по указанным ссылкам и добавить их в папки своего проекта.

Если вы ничего не хотите скачивать, то можно подключить файлы фреймворка через **cdn** следующим образом:

```
<!DOCTYPE html>  
<html lang="en">  
  
<head>  
    <meta charset="UTF-8">  
    <meta name="viewport"  
        content="width=device-width  
                initial-scale=1.0">  
    <title>Bootstrap</title>  
    <!-- подключение стилевого файла фреймворка  
        bootstrap -->  
    <link rel="stylesheet"  
        href="https://stackpath.bootstrapcdn.com/  
        bootstrap/4.5.0/css/bootstrap.min.css">  
</head>  
  
<body>  
    <!-- основной код страницы -->
```

```
<!--подключение скриптового файла библиотеки
    jquery, необходимого для работы фреймворка
-->
<script src="https://code.jquery.com/
    jquery-3.5.1.slim.min.js">
</script>
<!--подключение скриптового файла библиотеки
    popper, необходимого для работы фреймворка
-->
<script src="https://cdn.jsdelivr.net/npm/
    popper.js@1.16.0/dist/umd/popper.min.js">
</script>
<!-- подключение скриптового файла фреймворка
    bootstrap -->
<script src="https://stackpath.bootstrapcdn.com/
    bootstrap/4.5.0/js/bootstrap.min.js">
</script>
</body>
</html>
```

## 5. Цветовая схема Bootstrap

Bootstrap предлагает использовать десять различных классов для изменения цвета текста либо фона элемента. Имена классов связаны не с цветом, а с контекстом элемента. Выделяют следующие классы:

- ▷ **primary** — основной (первичный) цвет;
- ▷ **secondary** — вспомогательный (вторичный) цвет;
- ▷ **success** — цвет удачного выполнения действия;
- ▷ **danger** — цвет неудачного выполнения действия;
- ▷ **warning** — цвет предупреждения (акцента);
- ▷ **info** — цвет информационных блоков;
- ▷ **light** — светлый (`rgb(248, 249, 250)`);
- ▷ **dark** — темный (`rgb(52, 58, 64)`);
- ▷ **white** — белый;
- ▷ **transparent** — прозрачный.

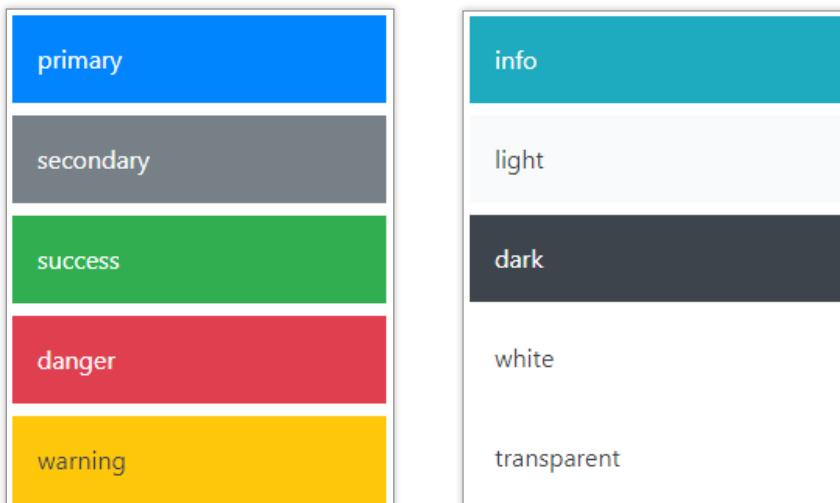


Рисунок 2

На рисунке 2 представлены префиксы классов и цвет, который они задают.

Для того, чтобы применить цвет для текста, следует перед префиксом написать «**text-**», а для задания фона-вого цвета класс «**bg-**».

```
<div class="bg-dark text-white">  
    bg-dark text-white  
</div>  
<div class="bg-light text-dark">  
    bg-light text-dark  
</div>  
<div class="bg-info text-light">  
    bg-info text-light  
</div>
```

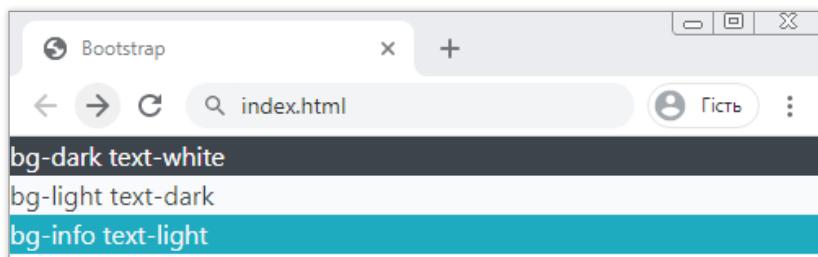


Рисунок 3

# 6. Система сеток

Фреймворк Bootstrap предлагает гибкую систему построения сеточной структуры, используя которую можно сравнительно легко и быстро получить адаптивную страницу. Сетка Bootstrap 4.\* использует технологию flexbox.

Bootstrap использует пять типоразмеров ширины устройства для формирования сетки. Каждому устройству соответствует свой префикс класса.

Таблица 1

Устройство	Ширина устройства	Префикс класса
Мобильные телефоны	<576px	Без префикса
Мобильные телефоны	>=576px	-sm
Планшеты	>=768px	-md
Мониторы	>=992px	-lg
Широкоформатные мониторы	>=1200px	-xl

Основным элементом структуры сетки является контейнер, который предназначен для размещения и/или центрирования элементов внутри него.

Bootstrap предлагает использовать три различных класса для контейнера:

- «**container**», который устанавливает фиксированную ширину для каждого устройства;
- «**container-fluid**», который устанавливает ширину равной 100% ширины родителя;
- «**container-{префикс класса, определяющего устройство}**», который устанавливает ширину равной 100% для устройств меньших, чем указано в префикссе

и фиксированную ширину для указанного устройства в префиксе.

Таблица 2

Устройство	Ширина устройства	Фиксированная ширина контейнера
Мобильные телефоны	<576px	100%
Мобильные телефоны	>=576px	540px
Планшеты	>=768px	720px
Мониторы	>=992px	960px
Широкоформатные мониторы	>=1200px	1140px

Внутри контейнера находятся строки (элементы с классом «`row`»), каждая из которых содержит столбцы (элементы с классом «`col`»).

Классы столбцов имеют число, которое указывает количество столбцов в строке. Максимальное количество столбцов в строке — 12. Ширина столбца никогда не фиксируется, так как задана в процентах от ширины родительского элемента-строки.

Рассмотрим на примерах использование сетки Bootstrap. Создадим строку с 5 столбцами одинаковой ширины. Добавим различные цвета фона столбцам.

```
<div class="container">
    <div class="row">
        <div class="col bg-dark text-white">
            col bg-dark text-white
        </div>
        <div class="col bg-primary text-white">
            col bg-primary text-white
        </div>
```

```

<div class="col bg-success text-white">
    col bg-success text-white
</div>
<div class="col bg-warning text-white">
    col bg-warning text-white
</div>
<div class="col bg-info text-white">
    col bg-info text-white
</div>
</div>
</div>

```



Рисунок 4

В браузере видно, что все ячейки одинаковой ширины, так как у всех одинаковый класс «`col`» без указания числа занимаемых столбцов. Максимальное количество столбцов в строке — 12, если столбцов будет больше, то они автоматически перейдут на новую строку. С помощью элемента с классом «`w-100`» можно разделить столбцы на несколько строк так как вам хочется. Пропишем такой элемент после второго столбца.

```

<div class="container">
    <div class="row">
        <div class="col bg-dark text-white">
            col bg-dark text-white
        </div>

```

```

<div class="col bg-primary text-white">
    col bg-primary text-white
</div>
<div class="w-100"></div>
<div class="col bg-success text-white">
    col bg-success text-white
</div>
<div class="col bg-warning text-white">
    col bg-warning text-white
</div>
<div class="col bg-info text-white">
    col bg-info text-white
</div>
</div>
</div>

```

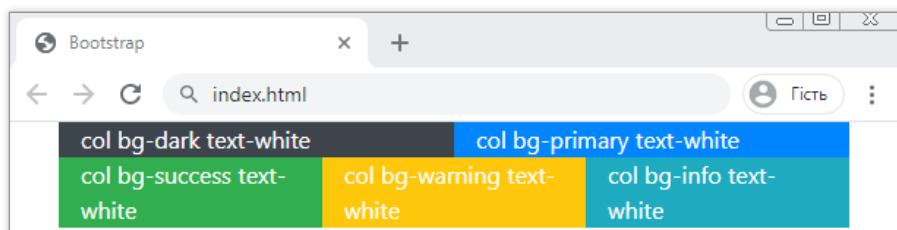


Рисунок 5

Добавляя число в конец класса столбца, можно определить размеры и расположение столбцов. Если сумма чисел, заданных столбцам будет превышать 12, то столбцы автоматически переходят на новую строку.

```

<div class="container">
    <div class="row">
        <div class="col-4 bg-dark text-white">
            col-4 bg-dark text-white
        </div>

```

```

<div class="col-8 bg-primary text-white">
    col-8 bg-primary text-white
</div>
<div class="col-3 bg-success text-white">
    col-3 bg-success text-white
</div>
<div class="col-6 bg-warning text-white">
    col-6 bg-warning text-white
</div>
<div class="col-3 bg-info text-white">
    col-3 bg-info text-white
</div>
</div>
</div>

```

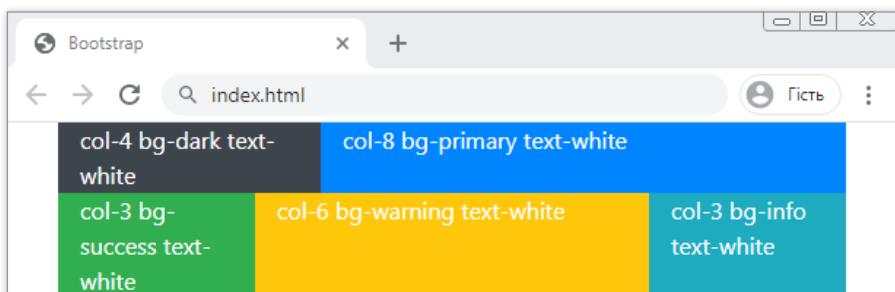


Рисунок 6

Описанный выше пример будет выглядеть одинаково на всех устройствах независимо от ширины устройства. При использовании префиксов устройств, можно переопределять ширину столбцов в зависимости от устройства. Рассмотрим на примере, создадим строку с четырьмя столбцами. На мобильных устройствах столбцы будут занимать всю ширину контейнера, на планшетах — половину, на мониторах и широкоформатных мониторах — одну третью часть ширины контейнера.

```
<div class="container">
  <div class="row">
    <div class="col-12 col-md-6 col-lg-3 bg-dark
      text-white">
      col-12 col-md-6 col-lg-3 bg-dark text-white
    </div>
    <div class="col-12 col-md-6 col-lg-3
      bg-primary text-white">
      col-12 col-md-6 col-lg-3 bg-primary text-white
    </div>
    <div class="col-12 col-md-6 col-lg-3
      bg-success text-white">
      col-12 col-md-6 col-lg-3 bg-success text-white
    </div>
    <div class="col-12 col-md-6 col-lg-3
      bg-warning text-white">
      col-12 col-md-6 col-lg-3 bg-warning text-white
    </div>
  </div>
</div>
```

Вид страницы в браузере представлен на рисунках 7-9.

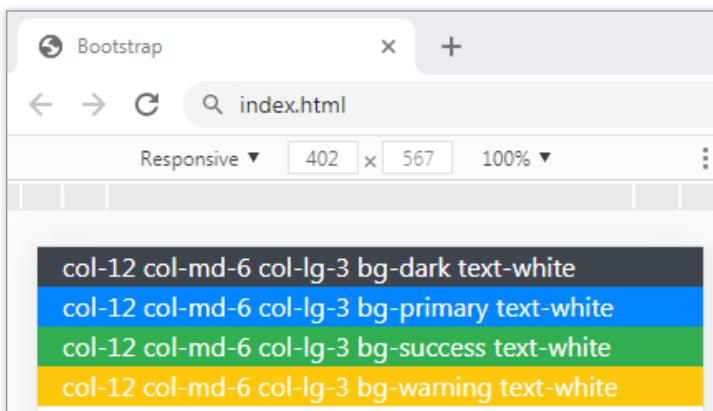


Рисунок 7. На мобильных

## 6. Система сеток

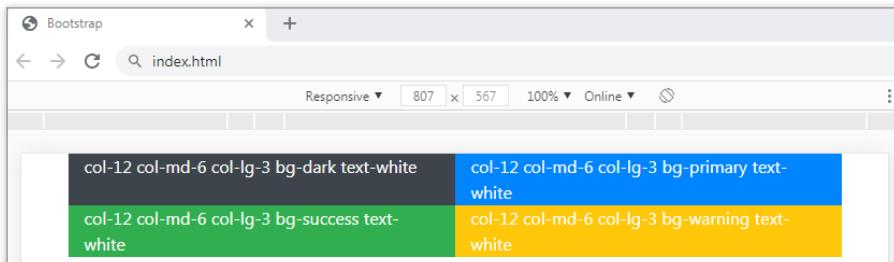


Рисунок 8. На планшетах

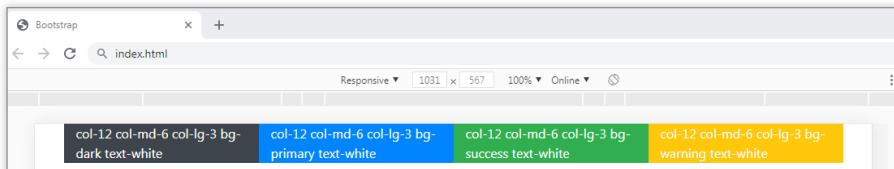


Рисунок 9. На мониторах и широкоформатных мониторах

Как видно из примера, столбцы на разных устройствах занимают разную ширину. Мы достигли этого благодаря прописыванию классов в каждый столбец. В случае, если ширина столбцов в строке должна быть одинаковой, а их количество должно меняться в зависимости от устройства, Bootstrap предлагает использовать класс «`row-cols-*`» для строк. Где звездочка обозначает максимальное число столбцов в строке.

```
<div class="container">
    <div class="row row row-cols-2 row-cols-sm-3
               row-cols-lg-5">
        <div class="col bg-primary text-white">
            col bg-primary text-white
        </div>
        <div class="col bg-success text-white">
            col bg-success text-white
        </div>
```

```

<div class="col bg-warning text-white">
    col bg-warning text-white
</div>
<div class="col bg-dark text-white">
    col bg-dark text-white
</div>
<div class="col bg-info text-white">
    col bg-info text-white
</div>
</div>
</div>

```

В браузере элементы будут располагаться как на рисунках ниже:

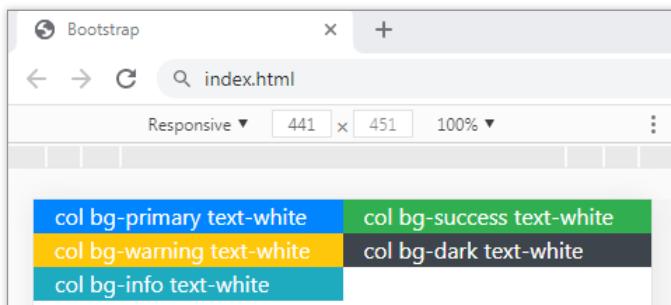


Рисунок 10. На мобильных

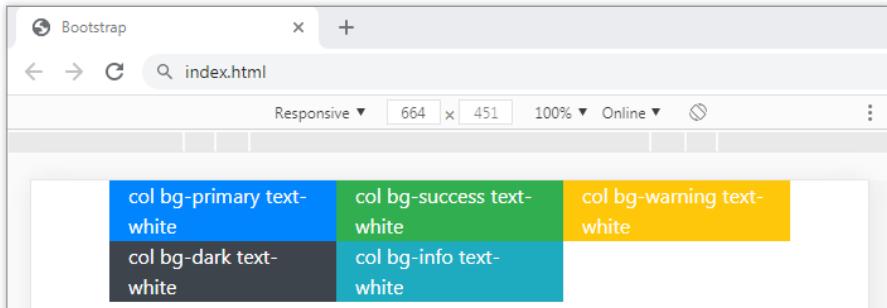


Рисунок 11. На планшетах



Рисунок 12. На мониторах

Как видно из примера, использовать классы для строк более удобно, чем прописывать нужный класс каждому элементу отдельно.

Также Bootstrap позволяет реализовывать отступы между элементами в строке и переопределять порядок отображения.

Для отступов используется класс «`offset-*`», где звездочка обозначает количество столбцов, которые надо отступить перед элементом.

Для переопределения порядка отображения элементов в строке используется класс «`order-*`». Где звездочка обозначает порядковый номер элемента в строке. Также можно отдельно указать первый элемент «`order-first`» и последний элемент «`order-last`». Рассмотрим использование данных классов на примере.

```
<div class="container">
  <div class="row row-cols-1 row-cols-md-3 ">
    <div class="col offset-md-1 bg-primary
              text-white order-4">
      4 col offset-md-1 bg-primary text-white order-4
    </div>
    <div class="col offset-md-2 bg-success
              text-white order-last">
      5 col offset-md-2 bg-success text-white
      order-last
    </div>
```

```
<div class="col bg-warning text-white">
    2 col bg-warning text-white</div>
<div class="col bg-dark text-white order-first">
    1 col bg-dark text-white order-first</div>
<div class="col bg-info text-white">
    3 col bg-info text-white</div>
</div>
</div>
```

В браузере код выглядит как на рисунках ниже:

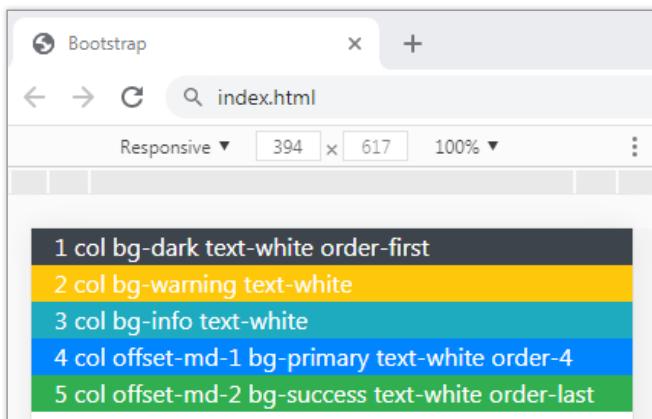


Рисунок 13. На мобильных

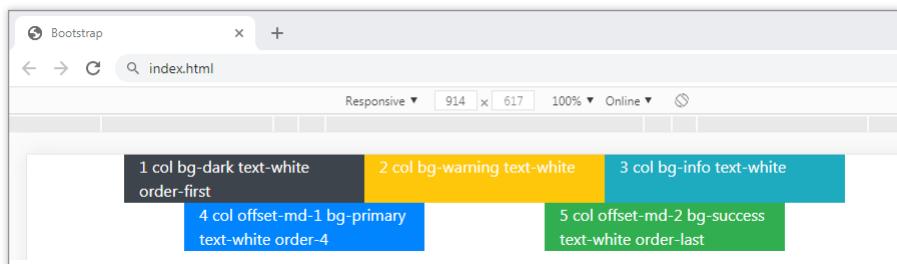


Рисунок 14. На планшетах

# 7. Утилиты Bootstrap

Утилитами называют вспомогательные функции, которые выполняют конкретные задачи и облегчают верстку. Фреймворк Bootstrap предлагает 20 различных утилит. Рассмотрим применение самых используемых из них.

## Рамка

Чтобы задать границы элементу достаточно прописать один из классов:

- ▷ `border`,
- ▷ `border-top`,
- ▷ `border-right`,
- ▷ `border-bottom`,
- ▷ `border-left`.

Для удаления границ у элемента, добавьте один из классов:

- ▷ `border-0`,
- ▷ `border-top-0`,
- ▷ `border-right-0`,
- ▷ `border-bottom-0`,
- ▷ `border-left-0`.

Для того, чтобы добавить цвет рамке, достаточно добавить класс `«border border-*»`, где звездочка название нужного класса цвета.

Чтобы закруглить углы элементу, пропишите один из классов:

- ▷ `rounded`,
- ▷ `rounded-top`,

- ▷ rounded-bottom,
- ▷ rounded-left,
- ▷ rounded-right.

Для большего или меньшего радиуса, необходимо добавить в конец к указанным классам префикс **-lg** или **-sm**.

Для получения окружности нужно использовать класс **rounded-circle** для квадратных элементов. Для получения полукруглых границ у прямоугольных элементов используйте класс **rounded-pill**.

Чтобы убрать закругленные углы у элемента добавьте класс **rounded-0**.

Рассмотрим на примере использование описанных классов:

```
div class="container">
  <div class="row row-cols-4">
    <div class="col border border-primary rounded">
      border border-primary rounded
    </div>
    <div class="col border-bottom border-success">
      border-bottom border-success
    </div>
    <div class="col border border-bottom-0
      border-warning">
      border border-bottom-0 border-warning
    </div>
    <div class="col border border-dark
      rounded-pill">
      border border-dark rounded-pill
    </div>
  </div>
</div>
```

В браузере приведенный пример выглядит так:

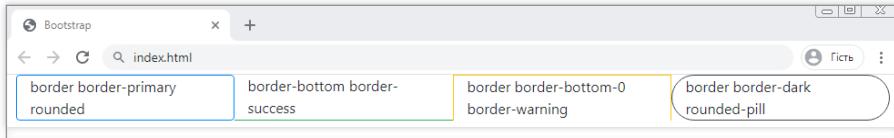


Рисунок 15

## Тени

Добавить или удалить тень можно с помощью классов:

- ▷ `shadow-none` — отсутствует тень,
- ▷ `shadow-sm` — небольшая тень,
- ▷ `shadow` — обычная тень,
- ▷ `shadow-lg` — большая тень.

```
<div class="container">
    <div class="row row-cols-4">
        <div class="col shadow-none">shadow-none</div>
        <div class="col shadow-sm">shadow-sm</div>
        <div class="col shadow">shadow</div>
        <div class="col shadow-lg">shadow-lg</div>
    </div>
</div>
```

В окне браузера приведенный код выглядит так:

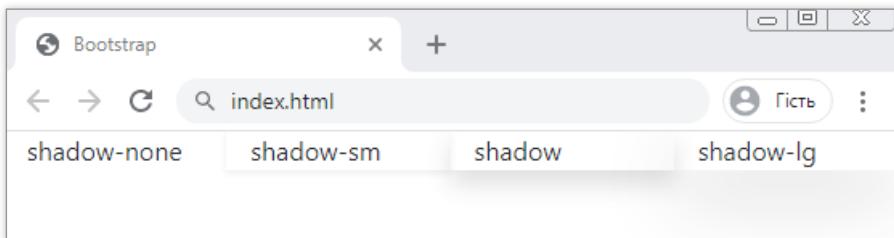


Рисунок 16

## Поля и отступы

Bootstrap предлагает использовать классы для задания адаптивных внешних и внутренних отступов.

Классы строятся по следующему формату: [свойство][сторона]-[префикс класса, определяющего устройство]-[размер].

Свойство может быть **m** (margin) и **p** (padding).

Сторона или стороны определяются следующими префиксами:

- ▷ **t** — верхняя,
- ▷ **b** — нижняя,
- ▷ **l** — левая,
- ▷ **r** — правая,
- ▷ **x** — левая и правая,
- ▷ **y** — верхняя и нижняя,
- ▷ **пусто** — все 4 стороны элемента.

Размер определяется одним из следующих префиксов:

- ▷ **0** — удаление полей либо отступов, устанавливается значение 0,
- ▷ **1** — устанавливается значение **0.25**,
- ▷ **2** — устанавливается значение **0.5**,
- ▷ **3** — устанавливается значение **1**,
- ▷ **4** — устанавливается значение **1.5**,
- ▷ **5** — устанавливается значение **3**,
- ▷ **auto** — устанавливают значение авто для внешних отступов.

Рассмотрим на примере. Создадим 4 столбца с различными левыми внутренними отступами для всех видов устройств и с верхним или нижним внутренним

отступом для устройств, начиная с планшета. Так же добавим внешний верхний отступ для устройств, начиная с планшета.

```
<div class="container-fluid">
    <div class="row row-cols-4">
        <div class="col bg-info pl-5 pt-md-2 mt-md-3">
            pl-5 pt-md-2 mt-md-3
        </div>
        <div class="col bg-warning pl-4 pb-md-2 mt-md-3">
            pl-4 pb-md-2 mt-md-3
        </div>
        <div class="col bg-light pl-3 pt-md-2 mt-md-3">
            pl-3 pt-md-2 mt-md-3
        </div>
        <div class="col bg-success pl-2 pb-md-2 mt-md-3">
            pl-2 pb-md-2 mt-md-3
        </div>
    </div>
</div>
```

В браузере наш пример будет выглядеть так:

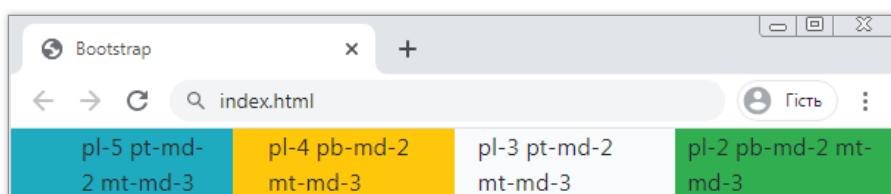


Рисунок 17. На мобильных



Рисунок 18. На планшетах

Bootstrap позволяет задать отрицательный внешний отступ. Для этого надо перед размером добавить «`n`». Добавим левый отрицательный отступ во второй столбец: «`ml-n5`». Теперь пример выглядит так:



Рисунок 19

## Изменение размеров

Фреймворк Bootstrap предлагает инструментарий для изменения ширины и высоты элементов.

Для изменения ширины и высоты относительно родительского элемента используйте один из классов, которые представлены в примерах ниже. Для ширины:

```
<div class="container">
  <div class="pl-3 bg-info w-25">w-25</div>
  <div class="pl-3 bg-warning w-50">w-50</div>
  <div class="pl-3 border bg-light w-75">w-75</div>
  <div class="pl-3 bg-success w-100">w-100</div>
  <div class="pl-3 bg-danger w-auto">w-auto</div>
</div>
```

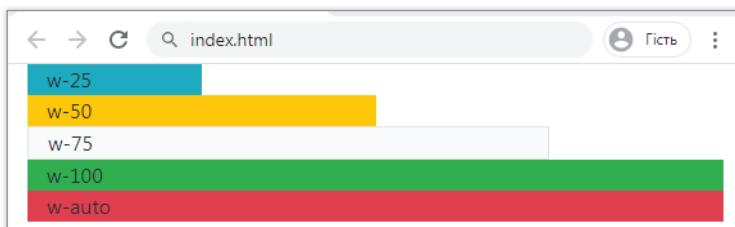


Рисунок 20

Для высоты:

```
<div class="container">
    <div class="row bg-info" style="height: 100px;">
        <div class="col h-25 bg-warning">h-25</div>
        <div class="col h-50 bg-primary">h-50</div>
        <div class="col h-75 bg-success">h-75</div>
        <div class="col h-100 bg-light">h-100</div>
        <div class="col h-auto bg-danger">h-auto</div>
    </div>
</div>
```



Рисунок 21

Также можно устанавливать ширину и высоту относительно области просмотра, используя классы для ширины: «`min-vw-100`» и «`vw-100`», и высоты: «`min-vh-100`» и «`vh-100`».

## Отображение/Скрытие

Для того, чтобы показать либо скрыть элементы на различных устройствах, нужно использовать класс, который строится по следующему формату: **d-[префикс класса, определяющего устройство]-[значение]**.

В качестве значения может быть одно из значений свойства `display`:

- ▷ `none`,
- ▷ `inline`,
- ▷ `inline-block`,
- ▷ `block`,
- ▷ `table`,
- ▷ `table-cell`,
- ▷ `table-row`,
- ▷ `flex`,
- ▷ `inline-flex`.

Рассмотрим на примере. Создадим строку из 5 столбцов. Первый столбец будет скрыт только на самом маленьком устройстве, а на остальных устройствах будет отображаться. Второй столбец будет скрываться только при мобильном отображении. Третий столбец будет отображаться на всех устройствах, кроме планшетов. Четвертый столбец будет скрыт на всех устройствах, кроме мобильного. Пятый столбец будет скрыт на всех устройствах, кроме планшетов.

```
<div class="container">
  <div class="row">
    <div class="col border d-none d-sm-block">
      1 d-none d-sm-block
    </div>
    <div class="col border d-sm-none d-md-block">
      2 d-sm-none d-md-block
    </div>
    <div class="col border d-md-none d-lg-block">
      3 d-md-none d-lg-block
    </div>
```

```
<div class="col border d-none d-sm-block  
      d-md-none">  
    4 d-none d-sm-block d-md-none  
</div>  
<div class="col border d-none d-md-block  
      d-lg-none">  
    5 d-none d-md-block d-lg-none  
</div>  
</div>  
</div>
```

В браузере пример будет выглядеть следующим образом:

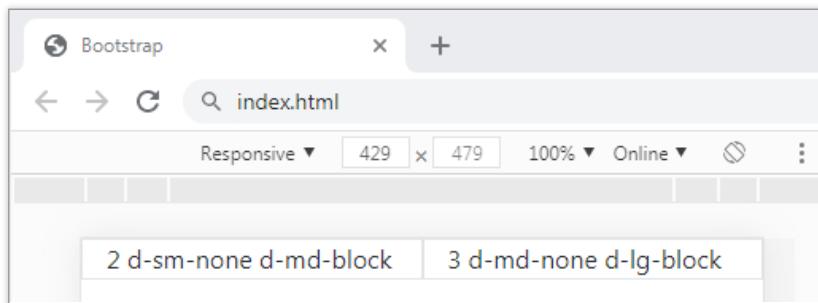


Рисунок 22. На самых маленьких устройствах

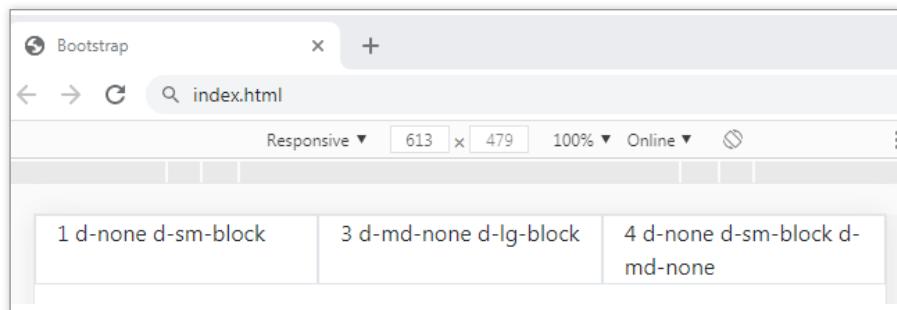


Рисунок 23. На мобильных

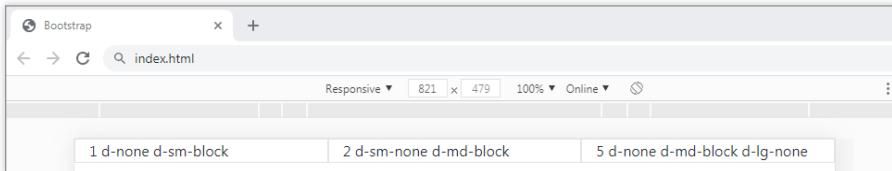


Рисунок 24. На планшетах



Рисунок 25. На мониторах

Как видно из примера, на разных устройствах отображаются различные столбцы, таким образом, используя классы для скрытия и отображения можно легко сверстать адаптивную страницу.

# 8. Использование различных паттернов Bootstrap

Фреймворк Bootstrap предлагает множество различных компонентов, которые представляют из себя шаблонные решения различных популярных задач веб-разработки. Рассмотрим использование некоторых компонентов на практике.

## Хлебные крошки

Хлебными крошками называется цепочка из ссылок, которая указывает на место текущей страницы в навигационной иерархии. Хлебные крошки помогают пользователю ориентироваться на сайте и позволяют быстро перейти на любой уровень иерархической цепочки. Таким образом хлебные крошки улучшают юзабилити и SEO сайта.

Для того, чтобы создать хлебные крошки, необходимо тегу `<nav>` задать атрибут `aria-label="breadcrumb"`, а контейнеру списка прописать класс со значением «`breadcrumb`». Каждый пункт меню должен содержать класс «`breadcrumb-item`», а активному пункту меню добавить класс «`active`» и атрибут `aria-current="page"`.

```
<nav aria-label="breadcrumb">  
    <ol class="breadcrumb">  
        <li class="breadcrumb-item">  
            <a href="#">Home</a>  
        </li>
```

```

<li class="breadcrumb-item">
    <a href="#">Category 1</a>
</li>
<li class="breadcrumb-item active"
    aria-current="page">
    Product 5
</li>
</ol>
</nav>

```

В браузере приведенный код выглядит так:

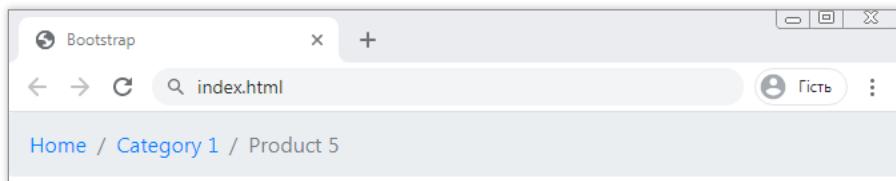


Рисунок 26

## Карточки

Карточки представляют адаптивный контейнер для отображения контента. В качестве контента могут быть изображения, текст, списки, ссылки, цитаты и так далее.

Основным элементом карточки является блочный контейнер с классом «`card`». Контейнер может состоять из трех основных частей: шапки (`class="card-header"`), основной части (`class="card-body"`) и подвала (`class="card-footer"`).

Для заголовка карточки используется класс «`card-title`», для подзаголовка класс «`card-subtitle`». Для текстового содержимого прописывают класс «`card-text`», а для изображений — класс «`card-img-top`».

Рассмотрим использование карточек на примерах. Создадим контейнер со строкой на четыре столбца, в каждом из которых будет карточка с различным контентом.

```
<div class="container-fluid">
  <div class="row row-cols-2 row-cols-lg-4">
    <div class="col">
      <div class="card">
        
        <div class="card-body">
          <h5 class="card-title">
            Заголовок карточки
          </h5>
          <h6 class="card-subtitle mb-2 text-muted">
            Подзаголовок карточки
          </h6>
          <p class="card-text">
            Карточка с изображением сверху.
          </p>
          <a href="#" class="btn btn-info">
            Ссылка
          </a>
        </div>
      </div>
    </div>
    <div class="col">
      <div class="card">
        <div class="card-header">
          <h5 class="card-title">
            Заголовок списка
          </h5>
        </div>
        <div class="card-body">
          <ul class="list-group list-group-flush ">
```

```
<li class="list-group-item ">
    Пункт 1
</li>
<li class="list-group-item ">
    Пункт 2
</li>
</ul>
</div>

</div>
</div>
<div class="col">
    <div class="card text-white">
        
        <div class="card-img-overlay">
            <h5 class="card-title">
                Заголовок карточки
            </h5>
            <p class="card-text">
                Карточка с фоновым изображением.
            </p>
            <a href="#" class="card-link text-info">
                Ссылка
            </a>
        </div>
        <div class="card-footer text-dark">
            З дня назад
        </div>
    </div>
</div>
<div class="col">
    <div class="card">
        <div class="card-header text-info">
```

## 8. Использование различных паттернов Bootstrap

```
Цитата
</div>
<div class="card-body">
  <blockquote class="blockquote mb-0">
    <p>Карточка с цитатой. Exer cita
       cumque quas quod, blanditiis
       ratione a, quos quam!
    </p>
    <footer class="blockquote-footer
      text-info">
      Автор
      <cite title="Source Title">
        название сайта
      </cite>
    </footer>
  </blockquote>
</div>
</div>
</div>
</div>
</div>
</div>
```

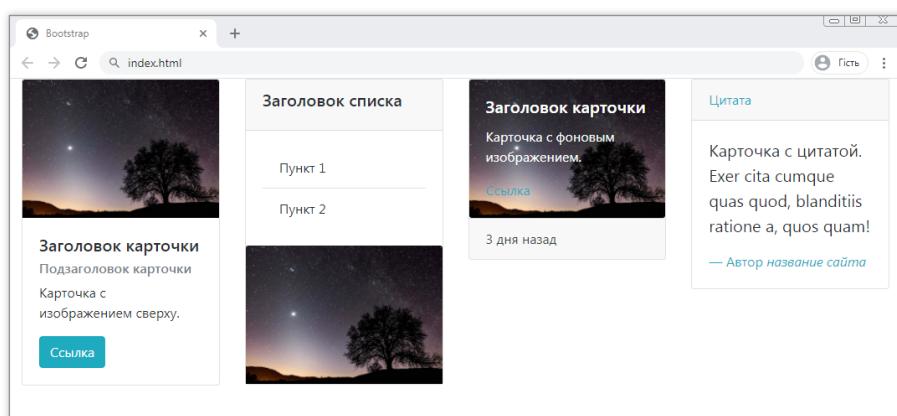


Рисунок 27

## Аккордеон

Аккордеон представляет из себя вертикально расположенный список элементов. При нажатии на каждый из элементов, может «развернуться» или «свернуться» контент, связанный с этим элементом. Аккордеон используют для более компактного представления контента.

Используя класс «`collapse`», с помощью которого осуществляется сворачивание/разворачивание содержимого элемента, а также компонент карточек, можно легко создать аккордеон. Рассмотрим на примере:

```
<!-- блочный элемент-контейнер с классом accordion и уникальным идентификатором, используется для целостной структуры -->
<div class="accordion" id="accordion">
    <!-- элемент аккордеона - карточка -->
    <div class="card">
        <div class="card-header" id="headOne">
            <h2 class="mb-0">
                <!-- элемент управления - кнопка с атрибутом data-target, значением которого является идентификатор скрываемого элемента -->
                <button class="btn btn-link btn-block text-left"
                    type="button"
                    data-target="#itemOne"
                    <!-- добавляется атрибут data-toggle со значением collapse, который осуществляет контроль над сворачиваемым элементом -->
                    data-toggle="collapse"
                    <!-- атрибут aria-expanded имеет значение true, если скрываемый элемент должен
```

```
отображаться при загрузке
страницы -->
aria-expanded="true">
Заголовок 1
</button>
</h2>
</div>
<!-- скрываемый элемент должен содержать класс
collapse, который обеспечивает сворачивание
элемента. Если необходимо отображать элемент
при загрузке страницы, необходимо добавить
класс show -->
<div id="itemOne"
      class="collapse show"
      aria-labelledby="headOne"
      data-parent="#accordion">
<div class="card-body">
    Vitae consequuntur possimus libero
    facere rerum vero consequatur nostrum
    velit, quas in, placeat distinctio,
    cum provident nemo aliquam. Asperiores
    explicabo labore minus.
</div>
</div>
</div>
<div class="card">
    <div class="card-header" id="headTwo">
        <h2 class="mb-0">
            <button class="btn btn-link btn-block
                      text-left collapsed"
                   type="button"
                   data-toggle="collapse"
                   data-target="#itemTwo"
                   aria-expanded="false"
                   aria-controls="collapseTwo">
                Заголовок 2
            </button>
        </h2>
    </div>
```

```
<div id="itemTwo"
      class="collapse "
      aria-labelledby="headTwo"
      data-parent="#accordion">
<div class="card-body">
    Minima repudiandae porro, est neque
    unde nostrum obcaecati, dolor
    cupiditate, exercitationem magnam
    assumenda! Unde ratione quisquam
    alias perferendis! Est tenetur facere
    aliquid?
</div>
</div>
</div>
<div class="card">
    <div class="card-header" id="headThree">
        <h2 class="mb-0">
            <button class="btn btn-link btn-block
                      text-left collapsed"
                   type="button"
                   data-toggle="collapse"
                   data-target="#itemThree"
                   aria-expanded="false"
                   aria-controls="collapseThree">
                Заголовок 3
            </button>
        </h2>
    </div>
    <div id="itemThree"
         class="collapse "
         aria-labelledby="headThree"
         data-parent="#accordion">
        <div class="card-body">
            Pariatur, excepturi. Aut doloremque
            cupiditate illo excepturi laborum
            eveniet iure recusandae non quasi
            harum praesentium vel quas, odit
        </div>
    </div>
</div>
```

```
        aspernatur minima optio suscipit.  
    </div>  
    </div>  
    </div>  
</div>
```

Так аккордеон выглядит в окне браузера:

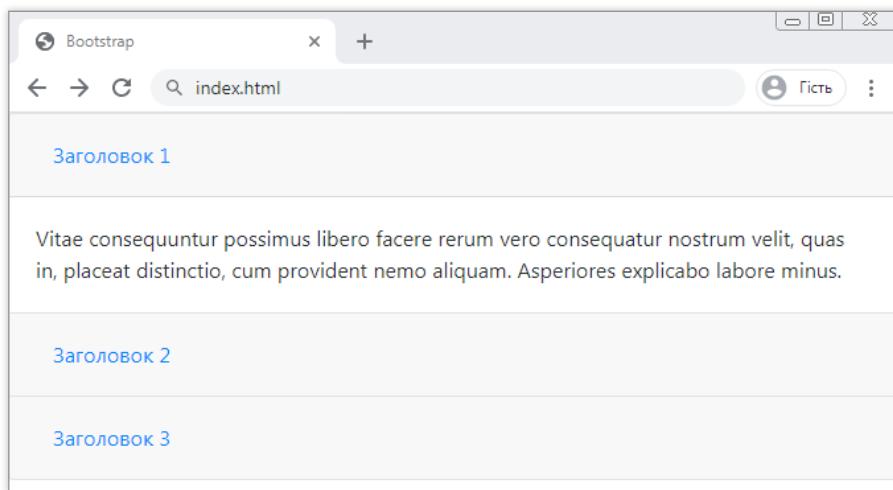


Рисунок 28

## Карусель

Для создания карусели из изображений либо контента, необходимо использовать определенную структуру вложенных блочных элементов с классами «`carousel`». Так же нужно добавить уникальный идентификатор для дополнительных элементов управления, особенно если требуется отображать несколько каруселей на одной странице.

Рассмотрим на примере создание слайд-шоу изображений с надписями. Создадим блочный элемент-

контейнер, который будет содержать изображения и элементы управления.

```
<!-- контейнер с идентификатором slides -->
<div id="slides"
      class="carousel slide"
      data-ride="carousel">
    <!-- элементы управления содержат атрибуты
        data-target с идентификатором карусели
        и data-slide-to с порядковым номером слайда,
        нумерация начинается с 0 -->
    <ol class="carousel-indicators">
        <li data-target="#slides"
            data-slide-to="0"
            class="active">
        </li>
        <li data-target="#slides" data-slide-to="1">
        </li>
        <li data-target="#slides" data-slide-to="2">
        </li>
    </ol>
    <!-- все слайды находятся в контейнере с классом
        carousel-inner-->
    <div class="carousel-inner">
        <!-- каждый слайд - блочный элемент с классом
            carousel-item. Одному из слайдов обязательно
            добавляем класс active, иначе карусель не
            будет отображаться. -->
        <div class="carousel-item active">
            <!-- изображению добавляем класс
                с utilityами для отображения и изменения
                размера -->
            
            <!-- добавляем блок с классом carousel-
                caption для описания изображения,
```

```
который будет видим на планшетах
и больших устройствах-->
<div class="carousel-caption d-none
           d-md-block">
    <h5>Title 1</h5>
    , <p>Lorem ipsum dolor sit amet,
        consectetur adipisicing elit.</p>
</div>
</div>
<div class="carousel-item">
    
    <div class="carousel-caption d-none
           d-md-block">
        <h5>Title 2</h5>
        <p>Praesentium cupiditate, neque ipsam
            magnam quidem ab eius nisi vitae.</p>
    </div>
</div>
<div class="carousel-item">
    
    <div class="carousel-caption d-none
           d-md-block">
        <h5>Title 3</h5>
        <p>Obcaecati ducimus est facere nostrum
            in illo nihil a accusamus iusto.</p>
    </div>
</div>
</div>
<!-- элементы управления - стрелки --&gt;
&lt;a class="carousel-control-prev"
    href="#slides"
    role="button"
    data-slide="prev"&gt;</pre>
```

```
<span class="carousel-control-prev-icon"  
      aria-hidden="true"></span>  
<span class="sr-only">Previous</span>  
</a>  
<a class="carousel-control-next"  
   href="#slides"  
   role="button"  
   data-slide="next">  
  <span class="carousel-control-next-icon"  
        aria-hidden="true"></span>  
  <span class="sr-only">Next</span>  
</a>  
</div>
```

Так карусель будет выглядеть в браузере:

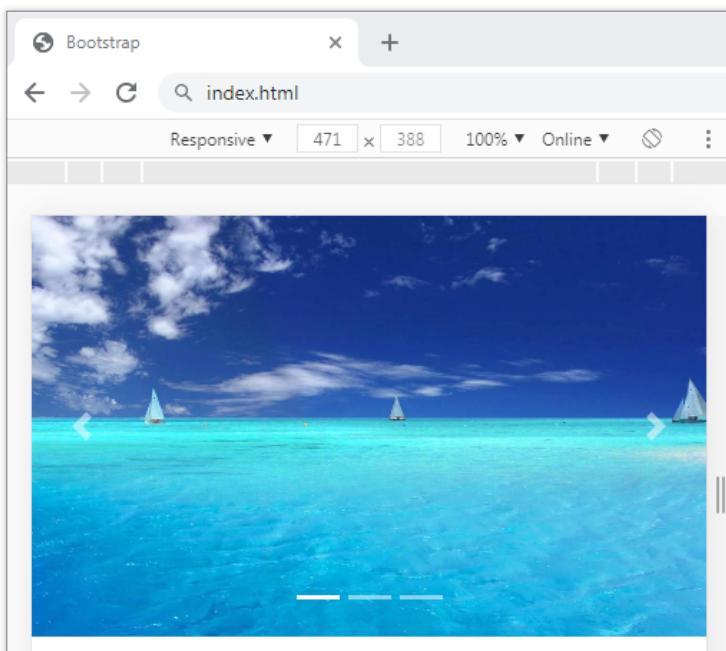


Рисунок 29. На мобильных устройствах первый слайд

## 8. Использование различных паттернов Bootstrap

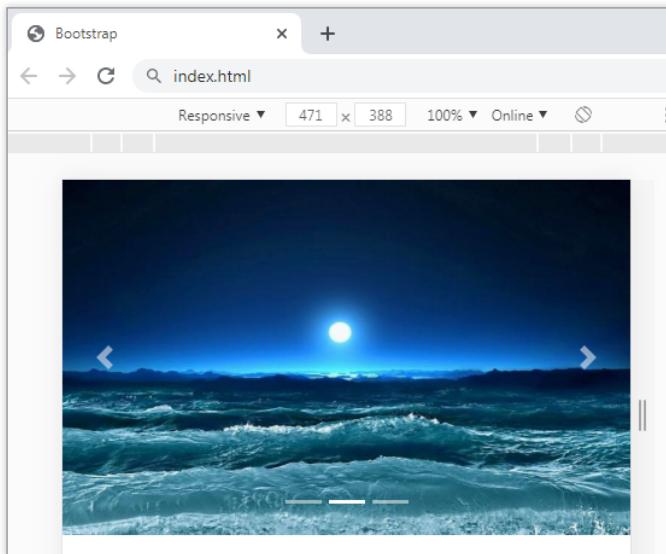


Рисунок 30. На мобильных устройствах второй слайд

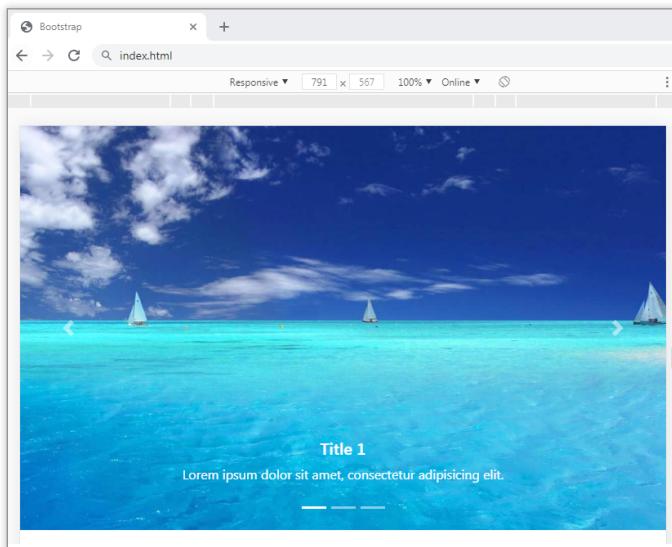


Рисунок 31. На планшетах и больших  
устройствах первый слайд

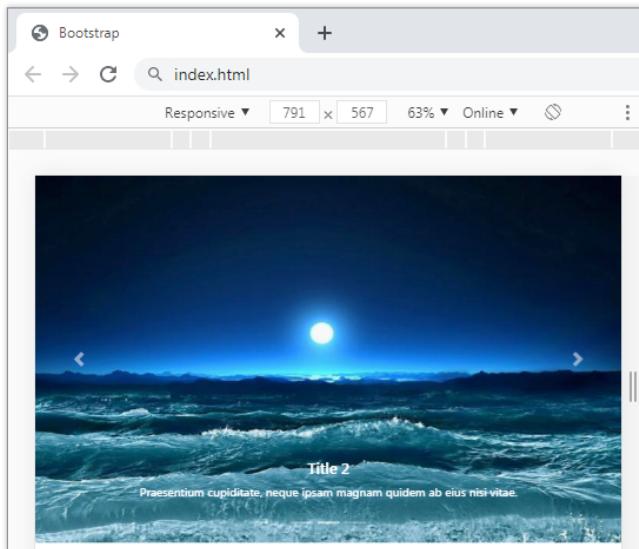


Рисунок 32. На планшетах и больших устройствах второй слайд

Для изменения настроек карусели контейнеру можно добавлять различные атрибуты `data-*`, где в качестве `*` может быть одно из значений, указанных в таблице 3.

Таблица 3

Значение	По умолчанию	Описание
<code>interval</code>	5000	Время задержки между автоматическим изменением слайдов, задается в миллисекундах
<code>keyboard</code>	true	Реакция карусели на события клавиатуры
<code>pause</code>	«hover»	Если установлено значение « <code>hover</code> », то при наведении на карусель курсора мыши, автоматическое перелистывание останавливается. Если установлено значение <code>false</code> , то при наведении на карусель курсора мыши, автоматическое перелистывание не прекращается.

Значение	По умолчанию	Описание
ride	false	Если установлено <b>false</b> , то автоматическое перелистывание начинается после того, как пользователь вручную переключает первый слайд. Если установлено значение « <b>carousel</b> », то автоматическое перелистывание начинается сразу после загрузки.
wrap	true	Если задано <b>true</b> – карусель постоянно перелистывается, если задано <b>false</b> – останавливается, дойдя до конца.
touch	true	Если задано <b>true</b> – карусель можно перелистывать движением пальцами вправо/влево, если задано <b>false</b> – карусель не реагирует на движение пальцев.

Мы рассмотрели максимально сложный пример создания карусели с различными элементами управления и описанием слайдов, на практике же можно исключать элементы управления и/или текстовые описания.

## Модальное окно

Модальными окнами называют всплывающие окна, которые блокируют работу пользователя с основной страницей до тех пор, пока пользователь модальное окно не закроет. В основном такие окна используют для диалога с пользователем.

Рассмотрим на примере создание модального окна с помощью фреймворка Bootstrap. В нашем примере модальное окно будет вызываться после нажатия на кнопку. Для этого создадим кнопку, которая будет иметь два атрибута для связи с модальным окном: **data-toggle="modal"**

и `data-target="#modalWindow"`. Первый атрибут добавляется управляющему элементу, в данном случае кнопке, который позволяет контролировать включение модального окна. Во втором атрибуте в качестве значения прописан уникальный идентификатор блочного элемента модального окна.

При клике на кнопку будет появляться модальное окно, а контент основной страницы будет скрываться под прозрачным серым фоном, занимающим всю область видимой части окна браузера. Само окно будет представлять из себя контейнер с контентом, разделенным на три части: шапку, основную часть и подвал.

```
<!-- кнопка, при нажатии на которую откроется  
модальное окно. имеет связь с окном через  
идентификатор модального окна, который прописывается  
после # в атрибуте data-target -->  
<button type="button"  
        class="btn btn-primary"  
        data-toggle="modal"  
        data-target="#modalWindow">  
    Кликни, чтобы увидеть модальное окно  
</button>  
<!-- блочный элемент-контейнер с классом modal  
и уникальным идентификатором modalWindow, который  
связан с кнопкой. -->  
<div class="modal fade"  
     id="modalWindow"  
     data-backdrop="static"  
     data-keyboard="false"  
     tabindex="-1"  
     role="dialog"  
     aria-hidden="true">  
    <!-- блочный элемент-контейнер для диалогового  
модального окна -->
```

```
<div class="modal-dialog">
    <!-- блочный элемент-контейнер для контента -->
    <div class="modal-content">
        <!-- блочный элемент-контейнер для шапки
        контента -->
        <div class="modal-header">
            <h5 class="modal-title">
                Заголовок модального окна
            </h5>
            <!-- кнопка для закрытия модального
            окна -->
            <button type="button"
                    class="close"
                    data-dismiss="modal"
                    aria-label="Close">
                <span aria-hidden="true">&times;;
                </span>
            </button>
        </div>
        <!-- блочный элемент-контейнер для
        основной части -->
        <div class="modal-body">
            <p>Основное содержимое модального окна</p>
        </div>
        <!-- блочный элемент-контейнер для
        подвала контента -->
        <div class="modal-footer">
            <!-- кнопка для закрытия модального
            окна -->
            <button type="button"
                    class="btn btn-secondary"
                    data-dismiss="modal">
                Закрыть
            </button>
            <!-- дополнительные кнопки с различным
            функционалом -->
        </div>
    </div>
</div>
```

```

<button type="button"
        class="btn btn-primary">
    Кнопка в подвале
</button>
</div>
</div>
</div>

```

Так пример отображается в браузере при клике на кнопку:

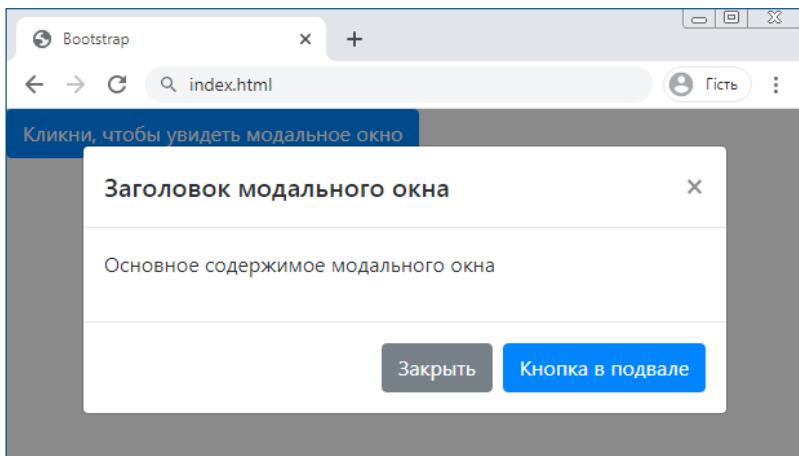


Рисунок 33

В качестве содержимого может быть любой необходимый вам контент: текст, изображения, форма регистрации/авторизации и так далее.

Для изменения настроек модального окна контейнеру можно добавлять различные атрибуты **data-\***, где в качестве **\*** может быть одно из значений, которые указаны в таблице 4.

Таблица 4

Значение	По умолчанию	Описание
backdrop	true	Если установлено значение «static», то модальное окно не скрывается при клике вне его.
keyboard	true	Если установлено значение true, то модальное окно закрывается при нажатии клавиши Escape.
focus	true	Если установлено значение true, то при инициализации фокус устанавливается на модальном окне.
show	true	Если установлено значение true, то при инициализации показывает модальное окно.

## Пагинация

Пагинацией называют разделение большого количества контента, имеющегося на сайте, на отдельные страницы для удобства использования.



Рисунок 34. Пример пагинации на сайте AliExpress

Для создания пагинации следует использовать элемент для навигации `nav`, который будет содержать список, в каждом пункте которого будет ссылка на страницу.

Для выделения пункта с ссылкой на текущую страницу используется класс «`active`». Для обозначения крайних пунктов, по которым невозможно осуществить переход, следует добавить класс «`disabled`».

В качестве контента вместо текста «Предыдущая/Следующая» для указания следующей или предыдущей страницы можно использовать специальные символы: «&laquo;» и «&raquo;».

```
<nav class="mt-2">
    <ul class="pagination justify-content-center">
        <!-- пункт с ссылкой, по которой невозможно
        осуществить переход -->
        <li class="page-item disabled">
            <a class="page-link" href="#" aria-label="Previous">
                <span aria-hidden="true">
                    &laquo;
                </span>
            </a>
        </li>
        <!-- пункт с ссылкой на текущую страницу -->
        <li class="page-item active">
            <a class="page-link" href="#">
                1
            </a>
        </li>
        <li class="page-item">
            <a class="page-link" href="#">
                2
            </a>
        </li>
        <li class="page-item">
            <a class="page-link" href="#">
                3
            </a>
        </li>
        <li class="page-item">
            <a class="page-link" href="#" aria-label="Next">
```

```
<span aria-hidden="true">&raquo;</span>
</a>
</li>
</ul>
</nav>
```

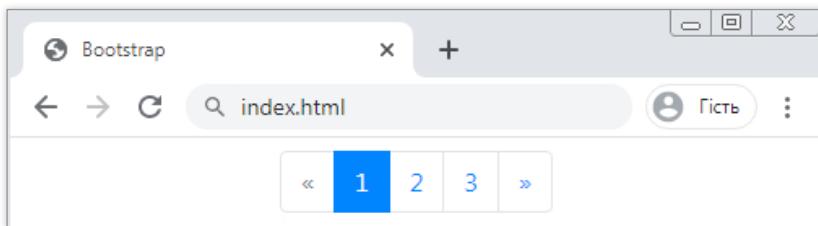


Рисунок 35

В данном разделе были рассмотрены популярные паттерны, которые часто используются при построении страницы. В следующих разделах урока мы рассмотрим более подробно еще два компонента, а именно навигацию и скроллинг.

## 9. Навигация и Bootstrap

**Навигация** — один из основных элементов любого сайта, как многостраничного, так и одностраничного.

Для построения навигации используется основной класс «`nav`», который добавляют контейнеру — элементу, содержащему ссылки. Например, списку:

```
<ul class="nav">
    <li class="nav-item">
        <a class="nav-link active" href="#">
            Active link
        </a>
    </li>
    <li class="nav-item">
        <a class="nav-link" href="#">
            Link
        </a>
    </li>
    <li class="nav-item">
        <a class="nav-link disabled" href="#">
            Disabled link
        </a>
    </li>
</ul>
```

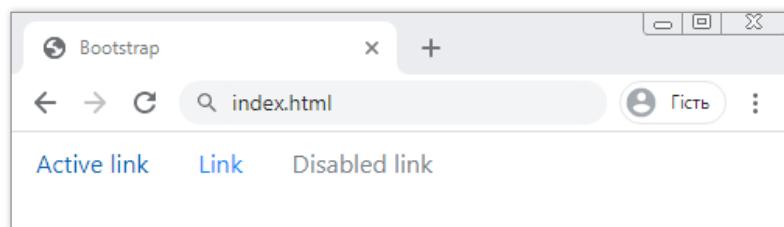


Рисунок 36. Отображение меню в браузере

Также класс «`nav`» можно добавить тегу `nav`:

```
<nav class="nav">
  <a class="nav-link active" href="#">
    Link active
  </a>
  <a class="nav-link" href="#">
    Link
  </a>
  <a class="nav-link disabled" href="#">
    Link disabled
  </a>
</nav>
```

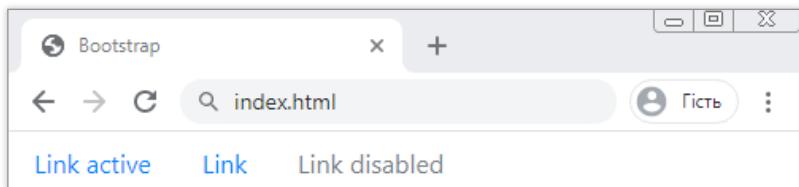


Рисунок 37. Отображение меню в браузере

Как видно из примеров, не обязательно использовать элемент списка для создания меню, достаточно добавить класс «`nav`».

В таблице ниже представлены классы с описанием, которые используются для построения навигации.

Таблица 5

Класс	Описание
<code>nav</code>	Базовый класс добавляется контейнеру для построения горизонтального меню с использованием технологии Flexbox
<code>nav-item</code>	Базовый класс добавляется тегу <code>&lt;li&gt;</code> для построения горизонтального меню с использованием технологии Flexbox

Класс	Описание
nav-link	Базовый класс добавляется ссылке для построения горизонтального меню с использование технологии Flexbox
justify-content-center	Для выравнивания элементов по центру. Добавляется контейнеру
justify-content-end	Для выравнивания элементов по правому краю. Добавляется контейнеру
flex-column	Для построения вертикального меню. Добавляется контейнеру
nav-tabs	Для создания меню в виде вкладок. Добавляется контейнеру
nav-pills	Для создания меню в виде кнопок. Добавляется контейнеру
nav-fill	Добавляется контейнеру, чтобы заполнить все доступное пространство пунктами меню
nav-justified	Добавляется контейнеру, чтобы все пункты меню были одинаковой ширины
Active	Добавляется ссылке на текущую страницу.
Disabled	Добавляется отключенной ссылке
Dropdown	Добавляется тегу <li>, который содержит подменю.
dropdown-toggle	Добавляется ссылке, по клику на которую открывается подменю
dropdown-menu	Добавляется элементу, являющемуся контейнером для подменю
dropdown-item	Добавляется ссылке подменю

Пример построения двух уровневого меню:

```
<ul class="nav nav-pills justify-content-center
          nav-justified">
    <li class="nav-item">
        <a class="nav-link active" href="#">
            Active link</a>
    </li>
```

```
<!-- пункт меню, содержащий подменю -->
<li class="nav-item dropdown">
    <a class="nav-link dropdown-toggle"
        data-toggle="dropdown"
        href="#">
        Dropdown link
    </a>
    <div class="dropdown-menu">
        <a class="dropdown-item" href="#">link</a>
        <a class="dropdown-item" href="#">link</a>
    </div>
</li>

<li class="nav-item">
    <a class="nav-link" href="#">Simple link</a>
</li>

<li class="nav-item">
    <a class="nav-link disabled" href="#">
        Disabled link
    </a>
</li>

</ul>
```

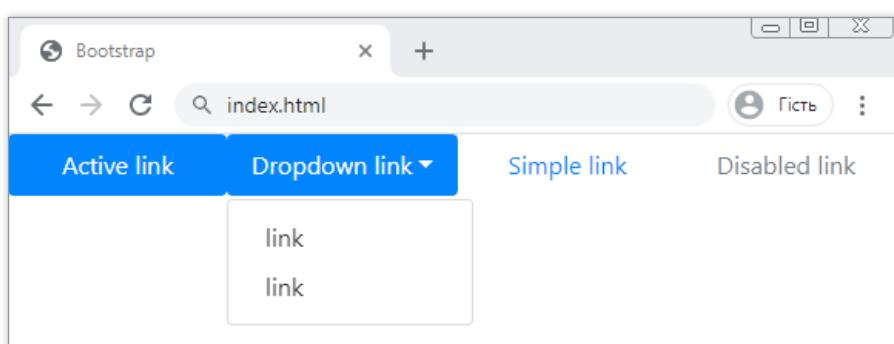


Рисунок 38. Отображение двухуровневого меню в браузере

Главное меню сайта обычно представлено навигационной панелью, которая находится в шапке сайта. Для ее построения Bootstrap предлагает использовать дополнительные классы, описанные в таблице ниже.

Таблица 6

Класс	Описание
navbar	Базовый класс добавляется контейнеру для построения горизонтальной навигационной панели
navbar-expand-*	Базовый класс добавляется контейнеру для указания префикса устройства, на котором будет происходить сворачивание/разворачивание навигационной панели
navbar-brand	Добавляется ссылке для обозначения главной страницы
navbar-nav	Для обозначения элемента-контейнера с меню
navbar-toggler	Добавляется элементу, по клику на который будет сворачиваться/разворачиваться меню
navbar-text	Для добавления вертикально центрированных строк текста
navbar-collapse	Для обозначения части меню, которая должна сворачиваться/разворачиваться на маленьких устройствах
fixed-top	Добавляется контейнеру для фиксации позиции навигационной панели сверху

Рассмотрим на примере построение адаптивной навигации, которая будет прятаться на мобильных устройствах и полностью отображаться на планшетах и больших устройствах. Навигационное меню будет содержать якорные ссылки для перехода к определенному разделу страницы. В качестве контента для разделов будем использовать изученные ранее компоненты.

```
<div class="container-fluid">
  <div class="row">
    <!-- навигационный элемент, который содержит
    непосредственно меню и элемент управления -->
    <nav class="col col-md-8 offset-md-2 navbar
      navbar-expand-md fixed-top
      navbar-light bg-white">
      <a class="navbar-brand" href="#section1">
        Brand
      </a>
      <!-- элемент управления для сворачивания/
      разворачивания мобильного меню -->
      <button class="navbar-toggler"
        type="button"
        data-toggle="collapse"
        data-target="#navbar"
        aria-controls="navbar"
        aria-expanded="false"
        aria-label="navigation">
        <!-- иконка для отображения кнопки элемента
        управления -->
        <span class="navbar-toggler-icon"></span>
      </button>
      <!-- часть меню, которая должна сворачиваться
      в мобильное меню для устройств меньших чем
      планшет -->
      <div class="collapse navbar-collapse"
        id="navbar">
        <!-- меню -->
        <div class="navbar-nav">
          <a class="nav-link active" href="#section1">
            Slider</a>
          <a class="nav-link" href="#section2">
            Accordion
          </a>
          <a class="nav-link" href="#section3">
            Cards
          </a>
        </div>
      </div>
    </nav>
  </div>
</div>
```

```
        </div>
    </div>
</nav>
</div>
</div>
<div class="container-fluid">
    <!-- первая секция - слайдер -->
    <div class="row mt-5" id="section1">
        <div class="col col-md-8 offset-md-2 mt-5">
            <h2>Carousel</h2>
            <div id="slides"
                class="carousel slide"
                data-ride="carousel">
                <ol class="carousel-indicators">
                    <li data-target="#slides"
                        data-slide-to="0"
                        class="active">
                    </li>
                    <li data-target="#slides"
                        data-slide-to="1">
                    </li>
                    <li data-target="#slides"
                        data-slide-to="2">
                    </li>
                </ol>
                <div class="carousel-inner">
                    <div class="carousel-item active">
                        
                        <div class="carousel-caption
                            d-none d-md-block">
                            <h5>Title 1</h5>
                            <p>Lorem ipsum dolor sit amet,
                                consectetur adipisicing elit.</p>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>
</div>
```

```
<div class="carousel-item">
    
    <div class="carousel-caption d-none
        d-md-block">
        <h5>Title 2</h5>
        <p>Praesentium cupiditate,
            neque ipsam magnam quidem ab eius
            nisi vitae.</p>
    </div>
</div>
<div class="carousel-item">
    
    <div class="carousel-caption d-none
        d-md-block">
        <h5>Title 3</h5>
        <p>Obcaecati ducimus est facere
            nostrum in illo nihil a accusamus
            iusto.</p>
    </div>
</div>
</div>
<a class="carousel-control-prev"
    href="#slides"
    role="button"
    data-slide="prev">
    <span class="carousel-control-prev-icon"
        aria-hidden="true"></span>
    <span class="sr-only">Previous</span>
</a>
<a class="carousel-control-next"
    href="#slides"
    role="button"
    data-slide="next">
```

```
<span class="carousel-control-next-icon"
      aria-hidden="true"></span>
<span class="sr-only">Next</span>
</a>
</div>
</div>
</div>

<!-- вторая секция - аккордеон -->
<div class="row mt-5" id="section2">
<div class="col col-md-8 offset-md-2 mt-5">
    <h2>Accordion</h2>
<div class="accordion" id="accordion">
    <div class="card">
        <div class="card-header" id="headOne">
            <h2 class="mb-0">
                <button class="btn btn-link
                           btn-block text-left"
                           type="button"
                           data-toggle="collapse"
                           data-target="#itemOne"
                           aria-expanded="true">
                    Заголовок 1
                </button>
            </h2>
        </div>
        <div id="itemOne"
             class="collapse show"
             aria-labelledby="headOne"
             data-parent="#accordion">
            <div class="card-body">
                <p> lorem(вместо слова lorem
                   вставьте много текста) </p>
                <p> lorem</p>
            </div>
        </div>
    </div>
</div>
```

```
<div class="card">
  <div class="card-header" id="headTwo">
    <h2 class="mb-0">
      <button class="btn btn-link
                  btn-block
                  text-left collapsed"
              type="button"
              data-toggle="collapse"
              data-target="#itemTwo"
              aria-expanded="false"
              aria-controls="collapseTwo">
        Заголовок 2
      </button>
    </h2>
  </div>
  <div id="itemTwo"
        class="collapse"
        aria-labelledby="headTwo"
        data-parent="#accordion">
    <div class="card-body">
      <p> lorem</p>
      <p> lorem</p>
    </div>
  </div>
</div>
<div class="card">
  <div class="card-header"
      id="headThree">
    <h2 class="mb-0">
      <button class="btn btn-link
                  btn-block text-left
                  collapsed"
              type="button"
              data-toggle="collapse"
              data-target="#itemThree"
              aria-expanded="false"
              aria-controls="collapseThree">
```

```
Заголовок 3
</button>
</h2>
</div>
<div id="itemThree"
      class="collapse"
      aria-labelledby="headThree"
      data-parent="#accordion">
<div class="card-body">
    <p> lorem </p>
    <p> lorem </p>
</div>
</div>
<div class="card">
    <div class="card-header" id="headFour">
        <h2 class="mb-0">
            <button class="btn btn-link
                        btn-block text-left
                        collapsed"
                   type="button"
                   data-toggle="collapse"
                   data-target="#itemFour"
                   aria-expanded="false"
                   aria-controls=
                        "collapseFour">
                Заголовок 4
            </button>
        </h2>
    </div>
    <div id="itemFour"
          class="collapse"
          aria-labelledby="headFour"
          data-parent="#accordion">
        <div class="card-body">
            <p> lorem </p>
```

```
        <p> lorem</p>
    </div>
</div>
</div>
</div>
</div>

<!-- третья секция - карточки --&gt;
&lt;div class="row mt-5" id="section3"&gt;
    &lt;div class="col col-md-8 offset-md-2 mt-5"&gt;
        &lt;h2&gt;Cards&lt;/h2&gt;
    &lt;/div&gt;
&lt;/div&gt;
&lt;div class="row" id="cards"&gt;
    &lt;div class="col col-md-4 offset-md-2"&gt;
        &lt;div class="card"&gt;
            &lt;img src="images/img-1.jpg"
                class="card-img-top"
                alt="night"&gt;
            &lt;div class="card-body"&gt;
                &lt;h5 class="card-title"&gt;
                    Заголовок карточки
                &lt;/h5&gt;
                &lt;h6 class="card-subtitle text-muted"&gt;
                    Подзаголовок карточки
                &lt;/h6&gt;
                &lt;p class="card-text"&gt;
                    Карточка с изображением сверху
                &lt;/p&gt;
                &lt;a href="#" class="btn btn-info"&gt;
                    Ссылка
                &lt;/a&gt;
            &lt;/div&gt;
        &lt;/div&gt;
    &lt;/div&gt;
&lt;/div&gt;</pre>
```

```
<div class="col col-md-4">
    <div class="card">
        
        <div class="card-body">
            <h5 class="card-title">
                Заголовок карточки
            </h5>
            <h6 class="card-subtitle text-muted">
                Подзаголовок карточки
            </h6>
            <p class="card-text">
                Карточка с изображением сверху.
            </p>
            <a href="#" class="btn btn-info">
                Ссылка
            </a>
        </div>
    </div>
</div>
<div class="row">
    <div class="col col-md-4 offset-md-2">
        <div class="card">
            
            <div class="card-body">
                <h5 class="card-title">
                    Заголовок карточки
                </h5>
                <h6 class="card-subtitle text-muted">
                    Подзаголовок карточки
                </h6>
```

```
<p class="card-text">  
    Карточка с изображением сверху.  
</p>  
<a href="#"  
    class="btn btn-info">  
    Ссылка  
</a>  
</div>  
</div>  
</div>  
  
<div class="col col-md-4">  
    <div class="card">  
          
        <div class="card-body">  
            <h5 class="card-title">  
                Заголовок карточки  
</h5>  
            <h6 class="card-subtitle text-muted">  
                Подзаголовок карточки  
</h6>  
            <p class="card-text">  
                Карточка с изображением сверху.  
</p>  
            <a href="#"  
                class="btn btn-info">  
                Ссылка  
</a>  
</div>  
</div>  
</div>  
</div>
```

Навигационная панель в браузере выглядит следующим образом:

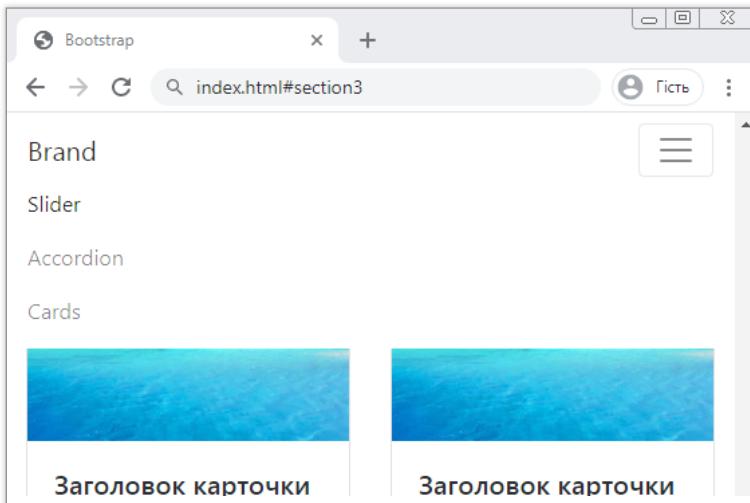


Рисунок 39. На мобильных устройствах в развернутом виде

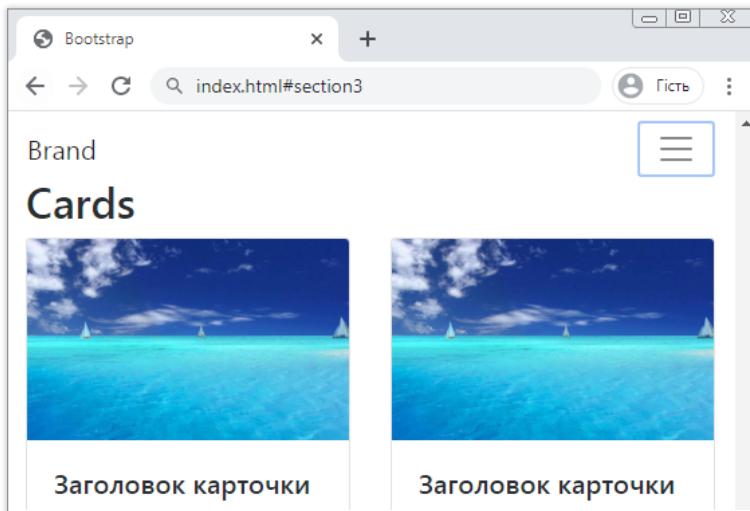


Рисунок 40. На мобильных устройствах в свернутом виде

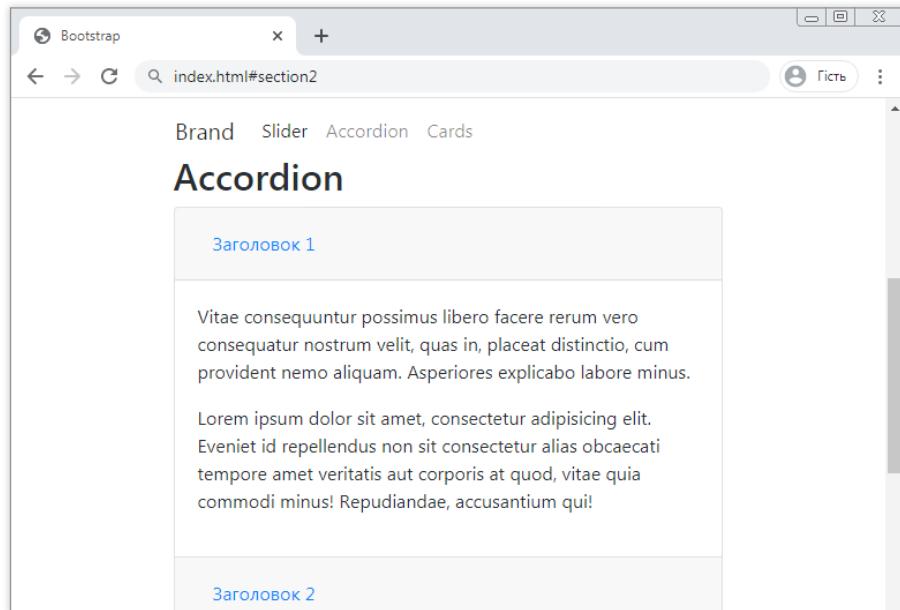


Рисунок 41. На планшетах и больших устройствах

Из приведенного примера видно, что для построения адаптивной навигации, необходимо использовать не только определенные классы, но и придерживаться определенной структуры элементов.

Однако, как мы видим, на последнем рисунке, активной ссылкой всегда является «Slider», так как именно эта ссылка имеет класс «**active**». С помощью компонента скроллинг, мы можем это исправить.

# 10. Скроллинг

Компонент скроллинг позволяет автоматически обновлять элементы навигации в зависимости от прокрутки, добавляя класс «`active`» той ссылке, которая отображается в области просмотра. Для того, чтобы легко добавить поведение прокрутки к навигации на верхней панели, добавьте атрибут `data-spy` со значением «`scroll`» элементу, на котором вы будете отслеживать прокрутку. Также добавьте атрибут `data-target`, значением которого будет идентификатор навигационного элемента.

Также можно добавить следующие атрибуты:

Таблица 7

Атрибут	По умолчанию	Описание
<code>data-offset</code>	10	Пиксели для смещения сверху при вычислении положения прокрутки.
<code>data-method</code>	Auto	Если установлено значение « <code>auto</code> », то выбирается наилучший метод отслеживания прокрутки. Если установлено значение « <code>offset</code> », то выбирается метод смещения для получения координат прокрутки. Если установлено значение « <code>position</code> », то выбирается метод позиционирования для получения координат прокрутки.

Для добавления компонента скроллинг, нам нужно немного модифицировать пример, рассмотренный выше.

```
<div class="container-fluid">
  <div class="row">
    <nav class="col col-md-8 offset-md-2 navbar
      navbar-expand-md fixed-top navbar-light bg-white">
```

```
<a class="navbar-brand" href="#section1">
    Brand
</a>

<button class="navbar-toggler"
        type="button"
        data-toggle="collapse"
        data-target="#navbar"
        aria-controls="navbar"
        aria-expanded="false"
        aria-label="navigation">
    <span class="navbar-toggler-icon"></span>
</button>

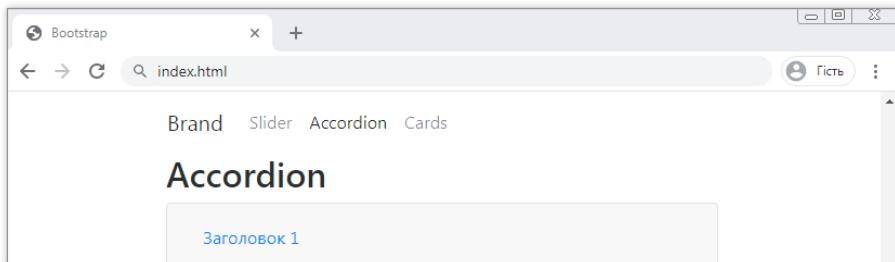
<div class="collapse navbar-collapse"
     id="navbar">
    <!-- добавляем идентификатор для связи
    с областью, в которой будем отслеживать
    положение прокрутки -->
    <div class="navbar-nav" id="navbarnav">
        <a class="nav-link active"
            href="#section1">
            Slider
        </a>
        <a class="nav-link" href="#section2">
            Accordion
        </a>
        <a class="nav-link" href="#section3">
            Cards
        </a>
    </div>
</div>
</nav>
</div>
</div>
<!-- добавляем атрибуты data-* для включения
    компонента скроллинга -->
```

```
<div class="container-fluid"
    data-spy="scroll"
    data-target="#navbarNav"
    data-offset="0">
```

Для работы функционала, необходимо добавить следующий скрипт перед закрывающим тегом **body**.

```
<script>
    $('body').scrollspy({ target: '#navbarNav' })
</script>
```

В этом скрипте инициализируется плагин **scrollspy** на целевом элементе с идентификатором «**navbarNav**».



*Рисунок 42*

На скриншоте выше видно, что теперь активная ссылка соответствует области просмотра, чего мы, собственно, и добивались.

# 11. Использование изображений и видео

Изображения являются неотъемлемой частью любого сайта. Фреймворк Bootstrap предлагает использовать следующие классы для изображений.

Таблица 8

Класс	Описание
img-fluid	Задает изображению свойства « <code>max-width: 100%;</code> » и « <code>height: auto;</code> »
img-thumbnail	Используется для придания изображению закругленной границы в <code>1px</code>
float-left	Применяется для выравнивания изображения по левой стороне
float-right	Применяется для выравнивания изображения по правой стороне
mx-auto d-block	Применяется для выравнивания изображения по центру, для этого нужно преобразовать изображение в блочный элемент, с помощью класс <code>d-block</code>

Рассмотрим использование некоторых классов для изображений на примере:

```
<div class="container-fluid">
    <div class="row mt-1">
        <div class="col">
            
        </div>
```

```
<div class="col">
    
</div>
</div>

<div class="row mt-1">
    <div class="col">
        
    </div>
</div>
</div>
```

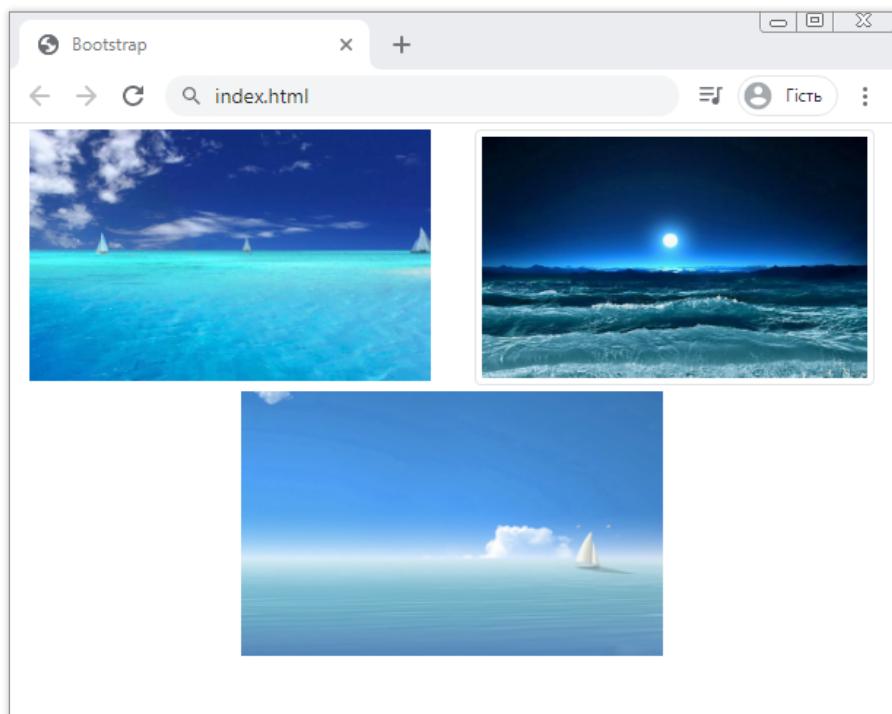


Рисунок 43. Отображение в браузере

Для встраиваемых элементов, таких как `<iframe>`, `<embed>`, `<video>` и `<object>`, необходимо использовать блочный элемент-обертку с классом «`embed-responsive`». Самим встраиваемым элементам задается класс «`embed-responsive-item`».

Также можно настроить соотношение сторон с помощью классов-модификаторов:

- ▷ `embed-responsive-21by9`,
- ▷ `embed-responsive-16by9`,
- ▷ `embed-responsive-4by3`,
- ▷ `embed-responsive-1by1`.

Рассмотрим на примере подключение встраиваемого видео с youtube с использованием вышеописанных классов:

```
<div class="container-fluid">
    <div class="row">
        <div class="col col-md-8 offset-md-2 mt-1">
            <div class="embed-responsive embed-responsive-16by9">
                <iframe class="embed-responsive-item" src="https://www.youtube.com/embed/DZKf9142WCo" allow="accelerometer; autoplay; encrypted-media; gyroscope; picture-in-picture" allowfullscreen>
                </iframe>
            </div>
        </div>
    </div>
</div>
```

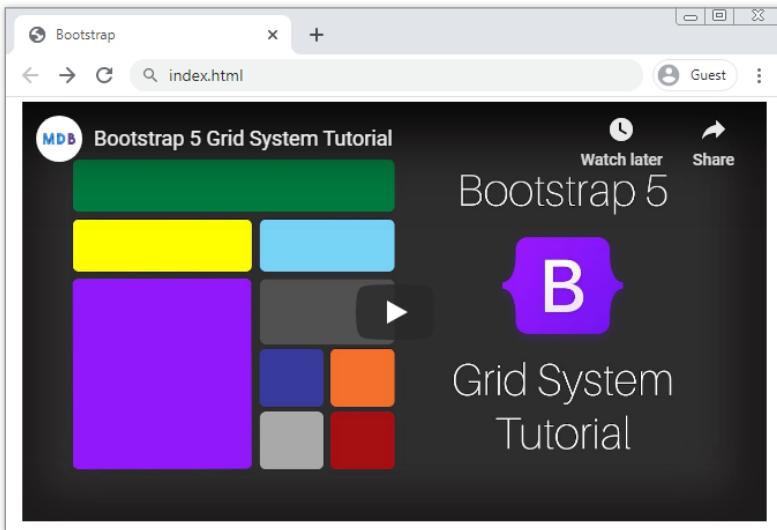


Рисунок 44. Отображение примера в браузере

# 12. Пример создания сайта с помощью фреймворка Bootstrap

Для закрепления полученных знаний создадим небольшой сайт, в котором будем использовать изученные компоненты фреймворка. На сайте будет 4 страницы: домашняя, блог, о нас и контакты. Ниже на изображениях (рис. 45-48) представлены отображения страниц на планшете.

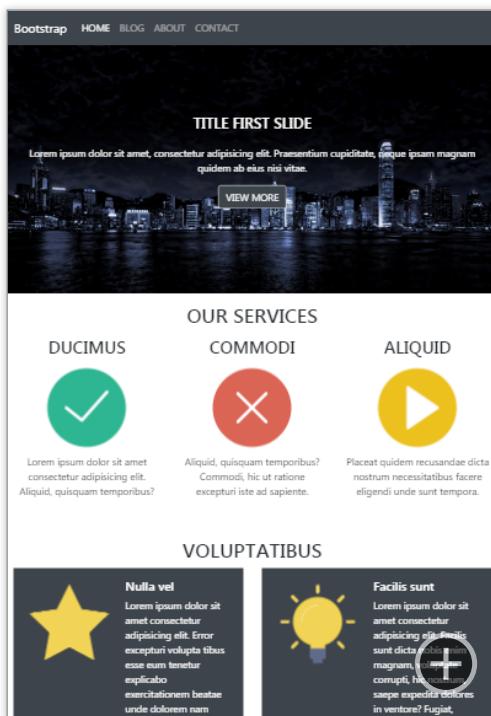


Рисунок 45. Фрагмент «Домашней страницы»

# Bootstrap

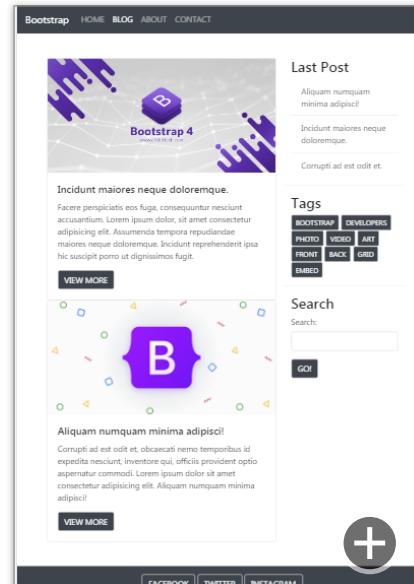


Рисунок 46. Страница «Блог»

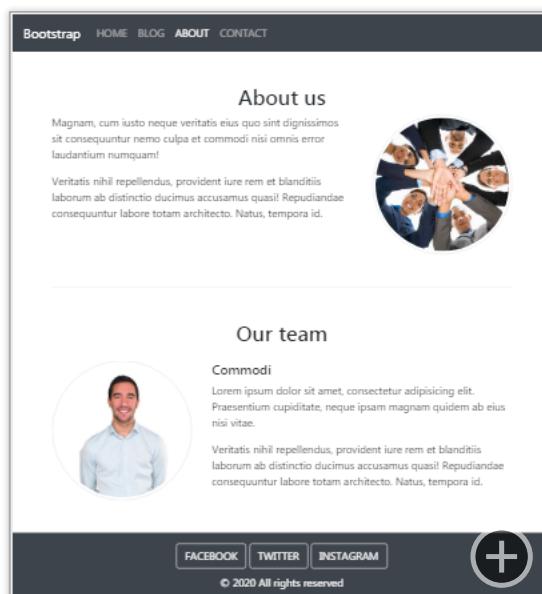


Рисунок 47. Страница «О нас»

## 12. Пример создания сайта с помощью фреймворка Bootstrap

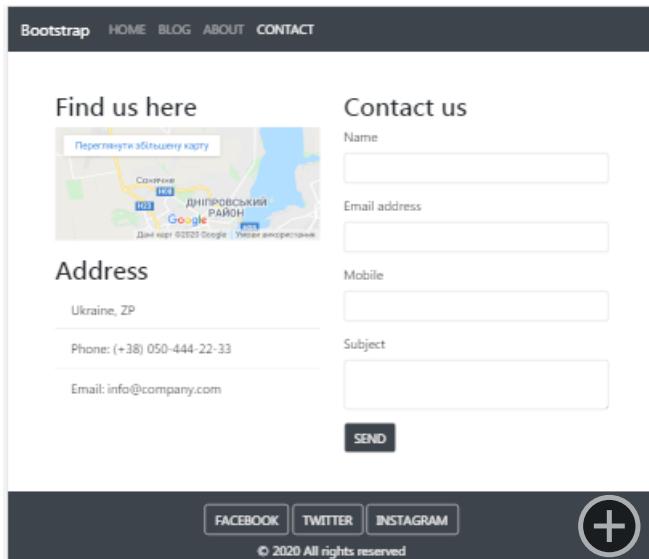


Рисунок 48. Страница «Контакты»

«Домашнюю страницу» реализуем на уроке, остальные три нужно будет создать в качестве домашнего задания.

Домашняя страница будет состоять из четырех строк. Первая область содержит слайдер с записями блога, вторая область будет состоять их трех колонок, третья область — из 4 колонок, в последней области три колонки.

Навигационное меню и подвал будут одинаковыми для всех страниц. Навигационное меню будет адаптивным: сворачиваться на мобильном отображении и разворачиваться на всю ширину браузера на планшетах и больших устройствах. В подвале будут ссылки на социальные сети.

Проектировать каждую область будем отдельно, используя принцип адаптивной верстки «начинай с мобильного».

Начнем с создания проекта и подключения файлов фреймворка. Нам также понадобится свой стилевой файл

для подключения фоновых изображений и добавления собственных стилевых свойств.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport"
        content="width=device-width, initial-scale=1.0">
    <title>Bootstrap</title>
    <link rel="stylesheet"
        href="css/bootstrap.min.css">
    <!-- стилевой файл проекта подключаем после
    стилевого файла фреймворка -->
    <link rel="stylesheet" href="css/style.css">
</head>

<body>
    <script src="https://code.jquery.com/
        jquery-3.5.1.slim.min.js">
    </script>
    <script src="https://cdn.jsdelivr.net/npm/
        popper.js@1.16.0/dist/umd/popper.min.js">
    </script>
    <script src="js/bootstrap.min.js"></script>
</body>
</html>
```

В шапке страницы необходимо создать адаптивное меню, которое будет сворачиваться на устройствах с шириной экрана менее **768px**.

Первая область будет содержать карусель, слайдами которой будет выступать текстовый контент. На устройствах с шириной экрана менее **570px**, в качестве контента будут выступать заголовок и кнопка, на остальных устройствах к контенту добавится абзац с текстом.

На рисунках 49-52 представлено отображение первой области вместе с основным навигационным меню на различных устройствах.

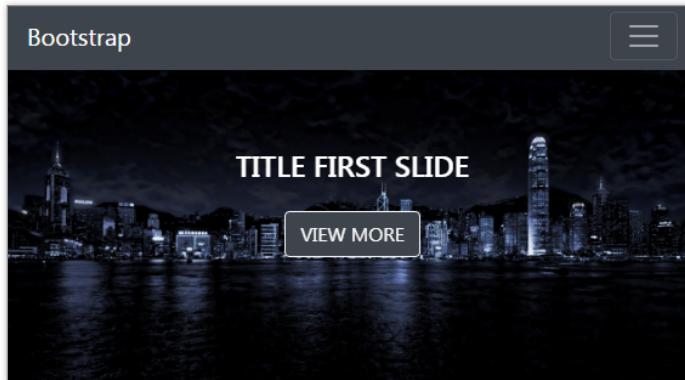


Рисунок 49. Первая область на устройствах с шириной экрана менее 570px со свернутым меню

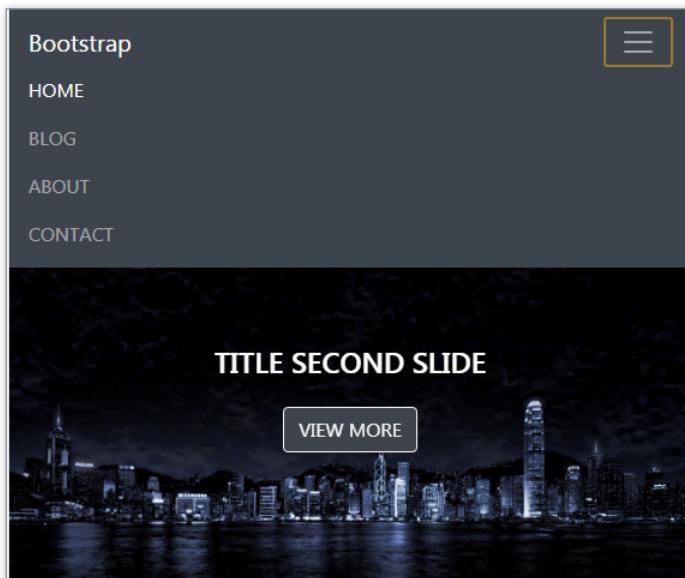


Рисунок 50. Первая область на устройствах с шириной экрана менее 570px с развернутым меню

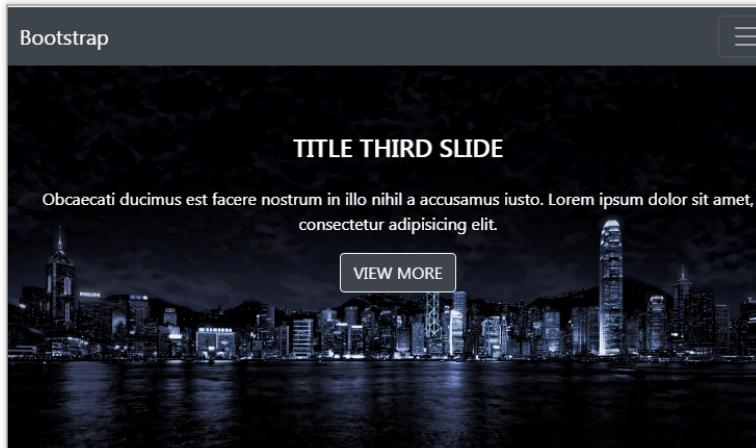


Рисунок 51. Первая область на мобильных устройствах с шириной экрана более 570px со свернутым меню

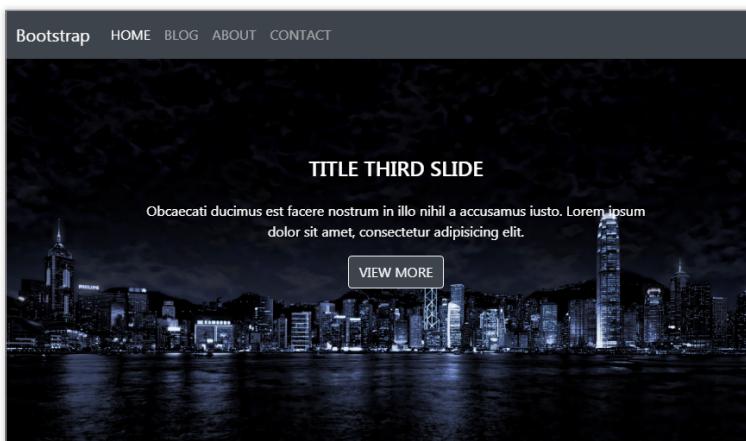


Рисунок 52. Первая область на планшетах и более широких устройствах с видимым меню

Приступим к созданию меню, которое будет одинаковым на всех страницах нашего сайта. Для этого создадим контейнер, который занимает всю ширину окна браузера на всех устройствах. В контейнере бу-

дёт одна строка, содержащая навигационный элемент с адаптивным меню.

```
<div class="container-fluid">
    <div class="row">
        <nav class="col navbar navbar-expand-md
                    navbar-dark bg-dark">
            <a class="navbar-brand"
                href="#">
                Bootstrap
            </a>

            <button class="navbar-toggler"
                    type="button"
                    data-toggle="collapse"
                    data-target="#navbar"
                    aria-controls="navbar"
                    aria-expanded="false"
                    aria-label="navigation">
                <span class="navbar-toggler-icon">
                </span>
            </button>

            <div class="collapse navbar-collapse"
                id="navbar">
                <div class="navbar-nav text-uppercase">

                    <!-- добавляем класс active к пункту
                    меню текущей страницы -->
                    <a class="nav-link active"
                        href="#">
                        Home
                    </a>
                    <a class="nav-link"
                        href="blog.html">
                        Blog
                    </a>
                </div>
            </div>
        </nav>
    </div>
</div>
```

```

        <a class="nav-link"
            href="about.html">
            About
        </a>
        <a class="nav-link"
            href="contact.html">
            Contact
        </a>
    </div>
</div>
</nav>
</div>
</div>

```

Следующий этап работы — создание карусели. Фоновое изображение карусели, также, как и меню, будет занимать всю ширину окна браузера на любом устройстве, следовательно, мы будем снова использовать контейнер с классом «`container-fluid`». Для фонового изображения создадим в собственном стилевом файле класс «`header_bg`», со следующими свойствами:

```

.header_bg{
    height: 50vw;
    max-height: 90vh;
    background-image: url(..../images/city1.jpg);
    background-size: 100%;
    background-repeat: no-repeat;
}

```

Приступаем к созданию карусели. Каждый слайд карусели будет представлять из себя текстовый контент, состоящий из заголовка, абзаца и кнопки. Так как абзац

должен быть виден только на устройствах с шириной экрана менее 570px, добавим соответствующие классы «d-none d-sm-block».

```
<div class="container-fluid">
    <div class="row header_bg">
        <div class="col pt-5">
            <div id="slides"
                class="carousel slide"
                data-ride="carousel">

                <div class="carousel-inner text-center
                    text-white mt-md-5">
                    <div class="carousel-item active">
                        <h4 class="py-3 text-uppercase">
                            Title first slide
                        </h4>
                        <p class="d-none d-sm-block">
                            Lorem ipsum dolor sit amet,
                            consectetur adipisicing elit.
                            Praesentium cupiditate, neque ipsam
                            magnam quidem ab eius nisi vitae.
                        </p>
                        <button class="btn btn-light
                            text-light bg-dark
                            text-uppercase">
                            View More
                        </button>
                    </div>

                    <div class="carousel-item">
                        <h4 class="py-3 text-uppercase">
                            Title second slide
                        </h4>
                        <p class="d-none d-sm-block">
                            Praesentium cupiditate, neque
                            ipsam magnam quidem ab eius nisi
                        </p>
                    </div>
                </div>
            </div>
        </div>
    </div>
```

```
        vitae. Obcaecati ducimus est  
        facere nostrum in illo nihil a  
        accusamus iusto.  
</p>  
<button class="btn btn-light text-light  
        bg-dark text-uppercase">  
    View More  
</button>  
</div>  
  
<div class="carousel-item">  
    <h4 class="py-3 text-uppercase">  
        Title third slide  
</h4>  
    <p class="d-none d-sm-block">  
        Obcaecati ducimus est facere  
        nostrum in illo nihil a accusamus  
        iusto. Lorem ipsum dolor sit  
        amet, consectetur adipisicing  
        elit.  
</p>  
  
<button class="btn btn-light text-light  
        bg-dark text-uppercase">  
    View More  
</button>  
</div>  
</div>  
</div>  
</div>  
</div>
```

Следующая область, будет перестраиваться в зависимости от ширины окна браузера. Соответствующее отображение представлено на рисунках 53, 54:

## 12. Пример создания сайта с помощью фреймворка Bootstrap

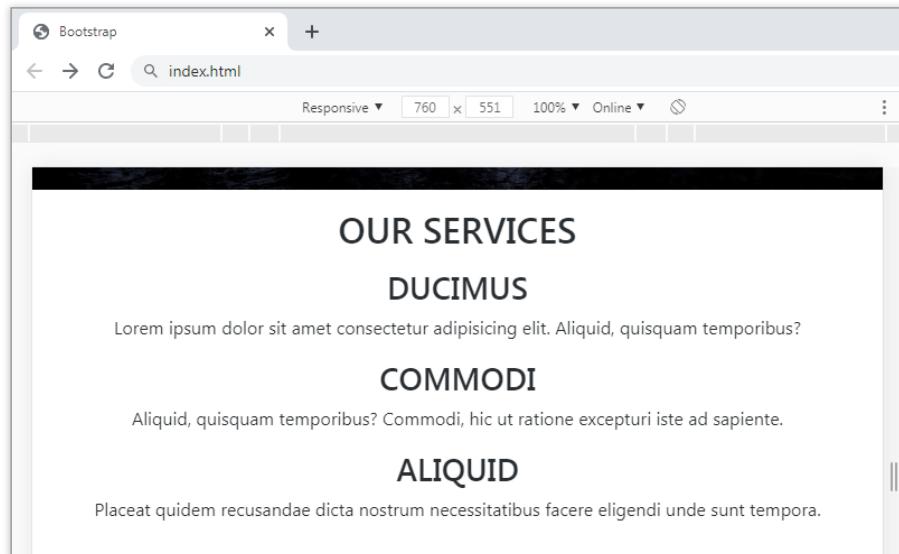


Рисунок 53. Вторая область на мобильных устройствах

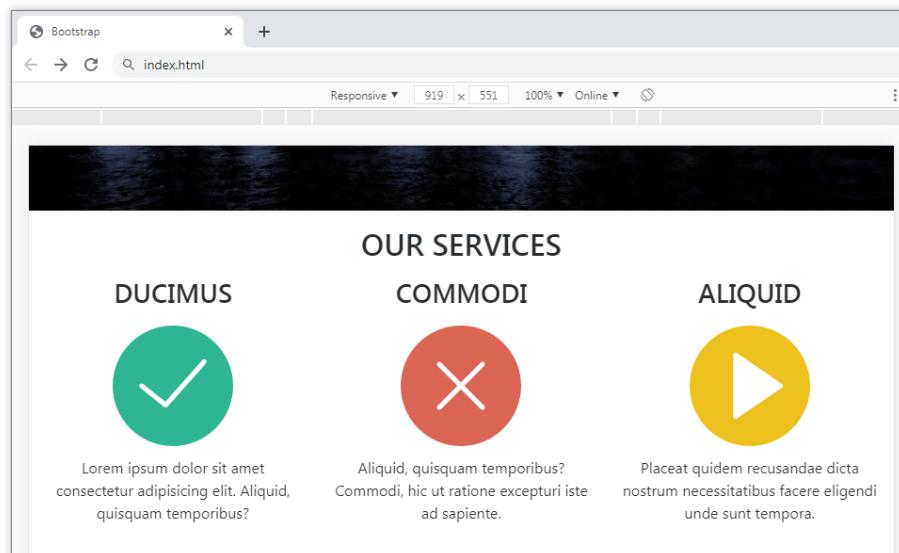


Рисунок 54. Вторая область на планшетах  
и более широких устройствах

Как видно из рисунков, нужно будет не только поменять расположение элементов с вертикального по одному в ряд на горизонтальное по три в ряд, но и добавить отображение иконок на планшетах и более широких устройствах.

```
<div class="container-fluid">

    <!-- первая строка для отображения заголовка
    области — одинаковая для всех устройств -->

    <div class="row mt-3">
        <div class="col">
            <h2 class="text-center text-uppercase">
                Our services
            </h2>
        </div>
    </div>

    <!-- вторая строка для отображения области
    в одну колонку на устройствах меньших чем
    планшет и в три колонки на планшетах и больших
    устройствах -->

    <div class="row row-cols-1 row-cols-md-3 mt-2">
        <div class="col text-center">
            <h3 class=" text-uppercase">Ducimus</h3>
            
            <p>Lorem ipsum dolor sit amet consectetur
            adipisicing elit. Aliquid, quisquam
            temporibus? </p>
        </div>

        <div class="col text-center">
            <h3 class=" text-uppercase">Commodi</h3>
```

```

<p>Aliquid, quisquam temporibus? Commodo,
   hic ut ratione excepturi iste ad
   sapiente.
</p>
</div>

<div class="col text-center">
  <h3 class=" text-uppercase">
    Aliquid
  </h3>
  
  <p>Placeat quidem recusandae dicta nostrum
     necessitatibus facere eligendi unde sunt
     tempora.
  </p>
</div>
</div>

</div>
```

Третья область будет содержать медиа блоки с изображениями, выровненными по левой стороне блока. Медиа-блок представляет из себя контейнер с изображением и текстовой частью.

Для удобства чтения будем изменять количество отображаемых блоков в ряду: на устройствах меньших планшета, по одному блоку в ряд, на остальных устройствах —

по два блока в ряд. На рисунках 55, 56 показаны соответствующие отображения:

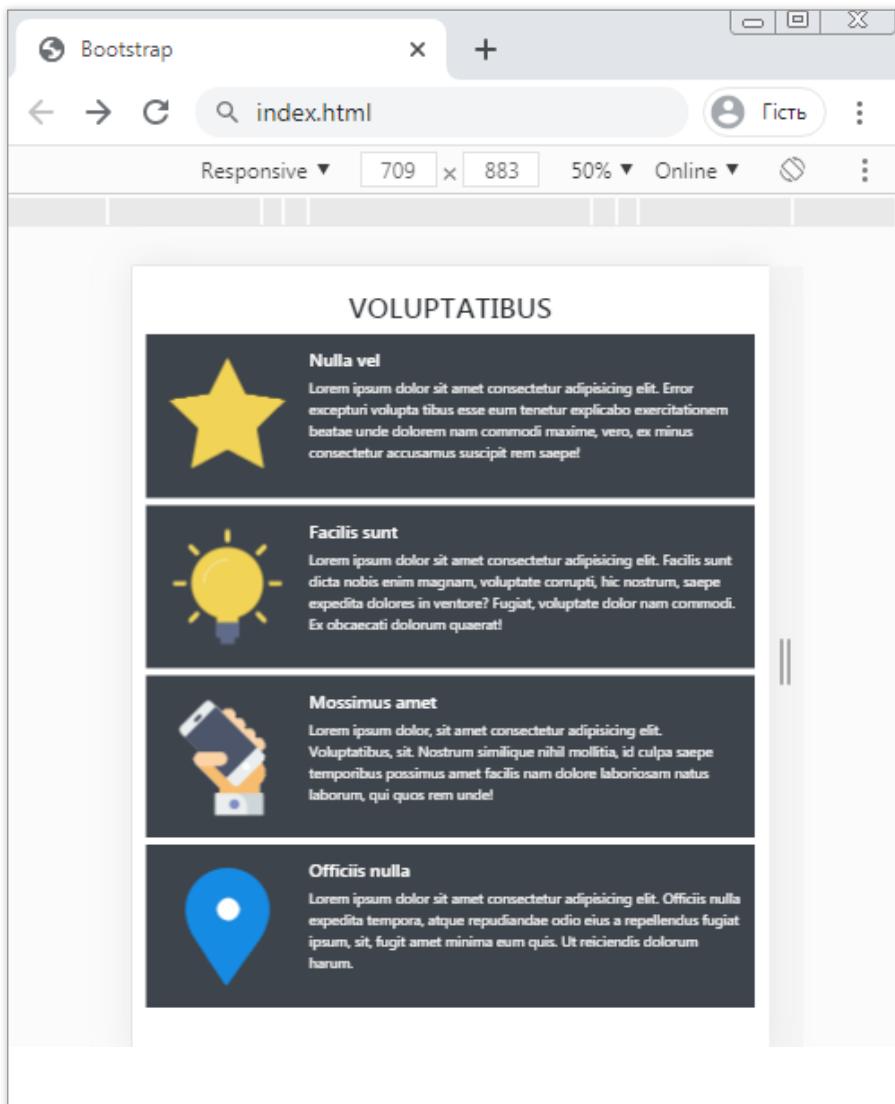


Рисунок 55. Третья область на мобильных устройствах  
(масштаб отображения на скрине – 50%)

## 12. Пример создания сайта с помощью фреймворка Bootstrap

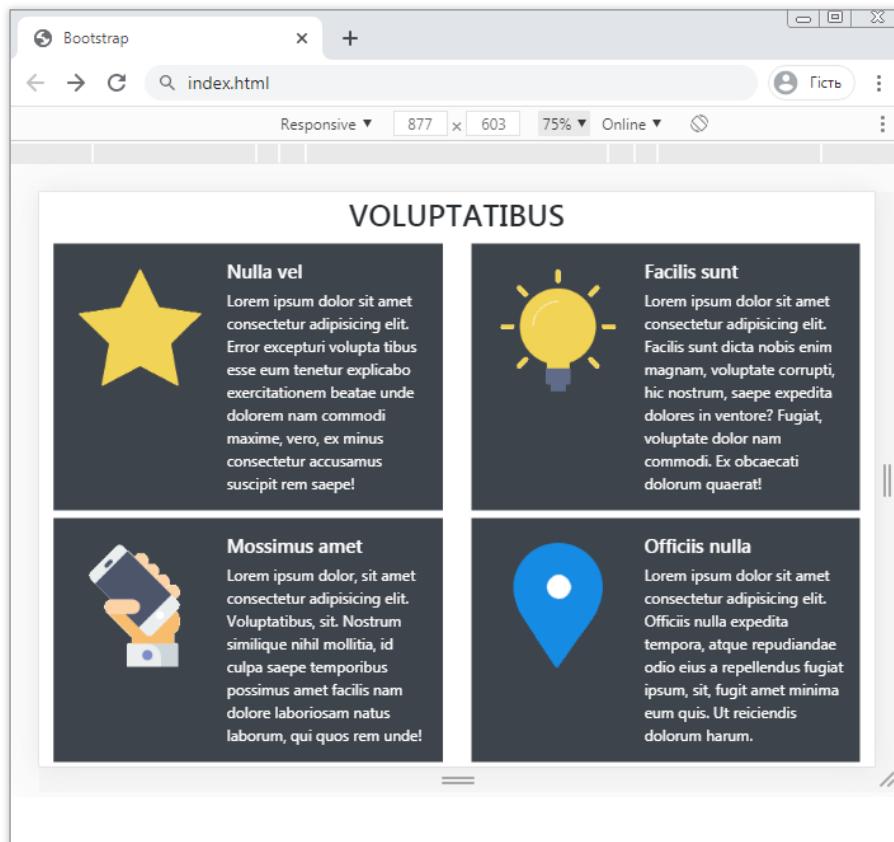


Рисунок 56. Третья область на планшетах и более широких устройствах (масштаб отображения на скрине – 75%)

Контейнер, включающий третью область, будет также отображаться на всю ширину окна браузера на всех устройствах. Для заголовка области создаем отдельную строку, занимающую всю ширину окна с выравниванием теста по середине.

Для блоков используем классы «`row`-`cols-1` `row`-`cols-md-2`» для перестройки элементов в зависимости от устройства.

```
<div class="container-fluid mt-5">
    <div class="row">
        <div class="col">
            <h2 class="text-center text-uppercase">
                Voluptatibus
            </h2>
        </div>
    </div>

    <div class="row row-cols-1 row-cols-md-2">
        <div class="col">
            <div class="media bg-dark text-white p-3 my-1">
                
                <div class="media-body">
                    <h5 class="mt-0">Nulla vel</h5>
                    <p>Lorem ipsum dolor sit amet
                        consectetur adipisicing elit.
                        Error excepturi volupta tibus
                        esse eum tenetur explicabo
                        exercitationem beatae unde dolorem
                        nam commodi maxime, vero, ex minus
                        consectetur accusamus suscipit rem
                        saepe!
                    </p>
                </div>
            </div>
        </div>

        <div class="col">
            <div class="media bg-dark text-white p-3 my-1">
                
                <div class="media-body">
                    <h5 class="mt-0">Facilis sunt</h5>
                </div>
            </div>
        </div>
    </div>
```

```
<p>Lorem ipsum dolor sit amet  
consectetur adipisicing elit.  
Facilis sunt dicta nobis enim  
magnam, voluptate corrupti, hic  
nostrum, saepe expedita dolores in  
ventore? Fugiat, voluptate dolor  
nam commodi. Ex obcaecati dolorum  
quaerat!  
</p>  
</div>  
</div>  
</div>  
</div>  
<div class="col">  
    <div class="media bg-dark text-white p-3 my-1">  
          
        <div class="media-body">  
            <h5 class="mt-0">Mossimus amet</h5>  
            <p>Lorem ipsum dolor, sit amet  
                consectetur adipisicing elit.  
                Voluptatibus, sit. Nostrum similique  
                nihil mollitia, id culpa saepe  
                temporibus possimus amet facilis nam  
                dolore laboriosam natus laborum, qui  
                quos rem unde!  
            </p>  
        </div>  
    </div>  
</div>  
<div class="col">  
    <div class="media bg-dark text-white p-3 my-1">  
          
        <div class="media-body">  
            <h5 class="mt-0">Officiis nulla</h5>
```

```
<p>Lorem ipsum dolor sit amet  
consectetur adipisicing elit.  
Officiis nulla expedita tempora,  
atque repudiandae odio eius a  
repellendus fugiat ipsum, sit, fugit  
amet minima eum quis. Ut reiciendis  
dolorum harum.</p>  
</div>  
</div>  
</div>  
</div>  
</div>
```

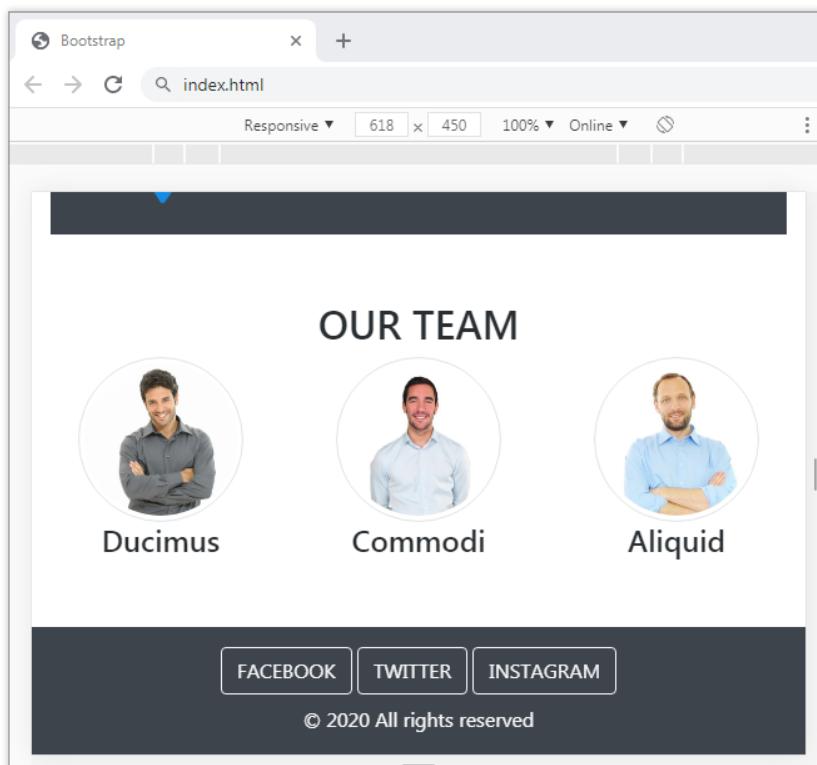


Рисунок 57. Четвертая область и подвал на всех устройствах

Приступим к разработке последней области. Контейнер, как и в других областях, будет занимать всю ширину окна и делиться на две строки: для заголовка области и для основного контента области. В строке основного контента будет три колонки для отображения изображений с подписями.

Для придания изображениям формы круга, используем класс «`rounded-circle`». Для того, чтобы изображения имели границу с отступом, используем класс «`img-thumbnail`». Если у вас большие изображения и вы хотите их уменьшить, можно добавить класс для изменения ширины «`w-75`».

```
<div class="container-fluid mt-5">
    <div class="row">
        <div class="col">
            <h2 class="text-center text-uppercase">
                Our team
            </h2>
        </div>
    </div>
    <div class="row row-cols-3">
        <div class="col text-center">
            
            <h4>Ducimus</h4>
        </div>
        <div class="col text-center">
            
            <h4>Commodi</h4>
        </div>
    </div>

```

```

<div class="col text-center">
    
    <h4>Aliquid</h4>
</div>
</div>
</div>

```

Нам осталось добавить подвал на нашу страницу. Для подвала будем использовать соответствующий тег **<footer>**, который будет являться контейнером для двух строк: первая с меню социальных сетей, и вторая с информацией о правах.

Для стилизации будем использовать классы, которые задают цвет: «bg-dark» для контейнера и «btn-light text-light bg-dark» для ссылок.

```

<footer class="container-fluid mt-5 pt-3 bg-dark">
    <div class="row row-cols-1">
        <div class="col">
            <div class="text-center">
                <a href="#" 
                    class="btn btn-light text-light
                        bg-dark text-uppercase">
                    Facebook
                </a>
                <a href="#" 
                    class="btn btn-light text-light
                        bg-dark text-uppercase">
                    Twitter
                </a>
                <a href="#" 
                    class="btn btn-light text-light
                        bg-dark text-uppercase">

```

```
        Instagram
    </a>
</div>
</div>
<div class="col text-center text-light mt-2">
    <p>&copy; 2020 All rights reserved</p>
</div>
</div>
</footer>
```

Мы создали домашнюю страницу небольшого сайта-блога, в которой использовали изученные компоненты и утилиты. Для закрепления материала вам необходимо выполнить домашнее задание, в котором вам предстоит создать оставшиеся три страницы.

# Домашнее задание

Добавить три страницы «Блог», «О нас» и «Контакты» в проект, который создавали выше на занятии. Протестировать отображение страниц на различных устройствах: мобильном, планшете, мониторе.

Страница «Блог» должна быть разделена на две колонки на планшетах. В левой широкой колонке располагаются статьи блога, для разработки которых можно использовать компонент карточки. Правая колонка представляет из себя сайдбар с тремя разделами: меню последних статей, области с тегами на популярные темы и формой для поиска по статьям блога.

На устройствах меньше планшета, вышеописанные элементы перестраиваются в одну колонку, причем сайдбар становится вниз под статьи.

На устройствах больше планшета статьи блога выстраиваются по две в ряд.

Навигационное меню в шапке сайта и подвал сайта остаются такими же, как и домашней странице. На рисунках 58-61 представлены скрины страницы «Блога» на различных устройствах.

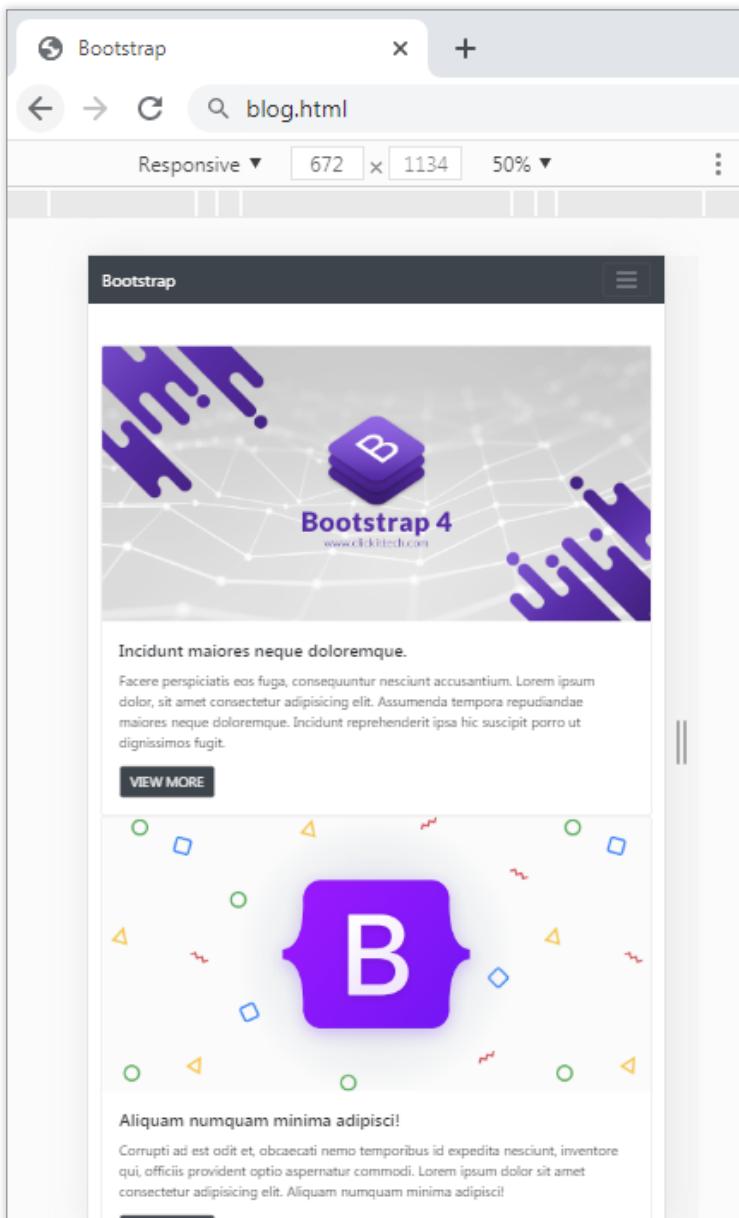


Рисунок 58. Отображение статей на мобильных устройствах и меньше

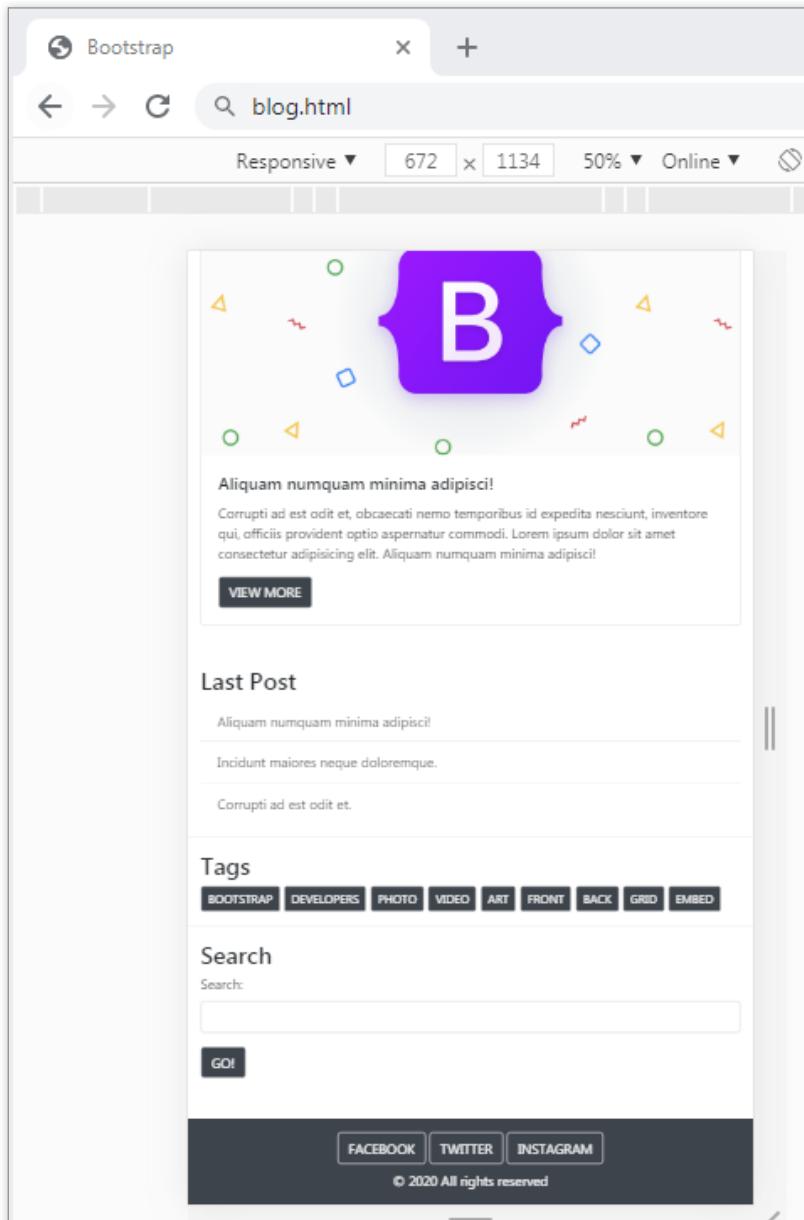


Рисунок 59. Отображение сайдбара на мобильных устройствах и меньше

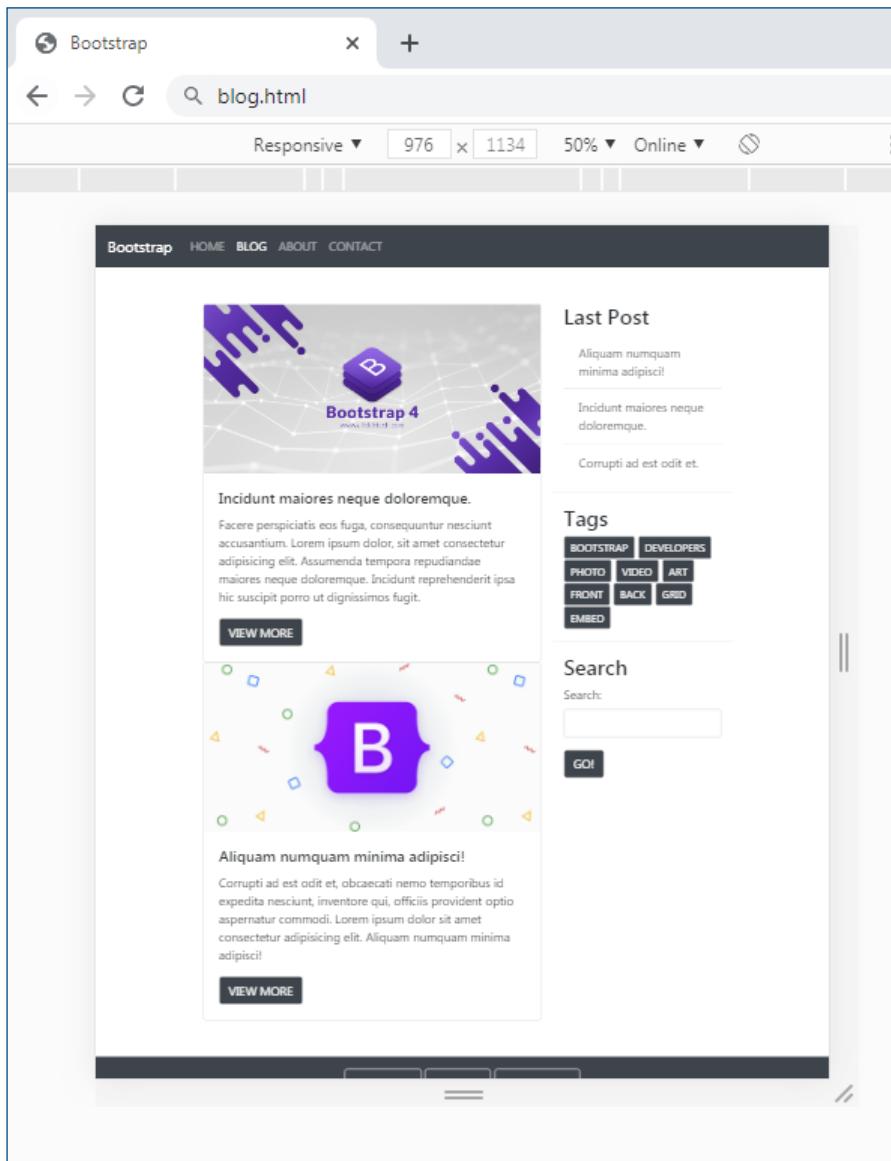


Рисунок 60. Отображение страницы «Блог» на планшетах

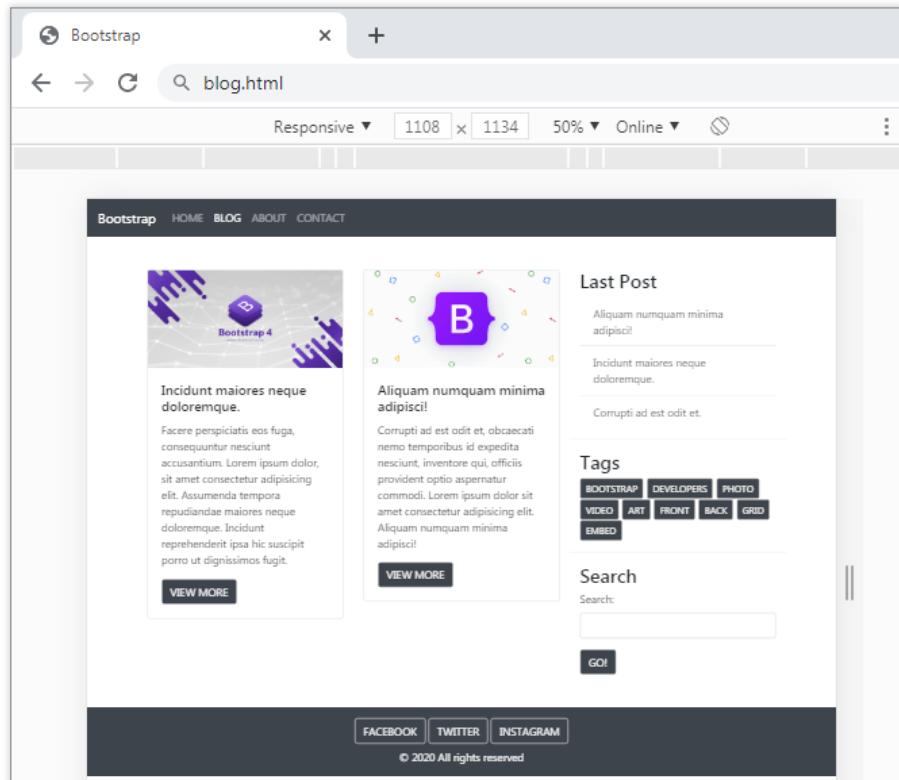


Рисунок 61. Отображение страницы «Блог» на мониторах и больших устройствах

Страница «О нас» должна состоять из двух разделов. Первый с карточкой и изображением, выровненным по правой стороне. Второй с каруселью, где слайды представлены в виде контента: изображения, заголовка и абзацев. Между областями должна быть разделительная горизонтальная линия.

На скринах ниже показано как должна отображаться страница «О нас» на различных устройствах.

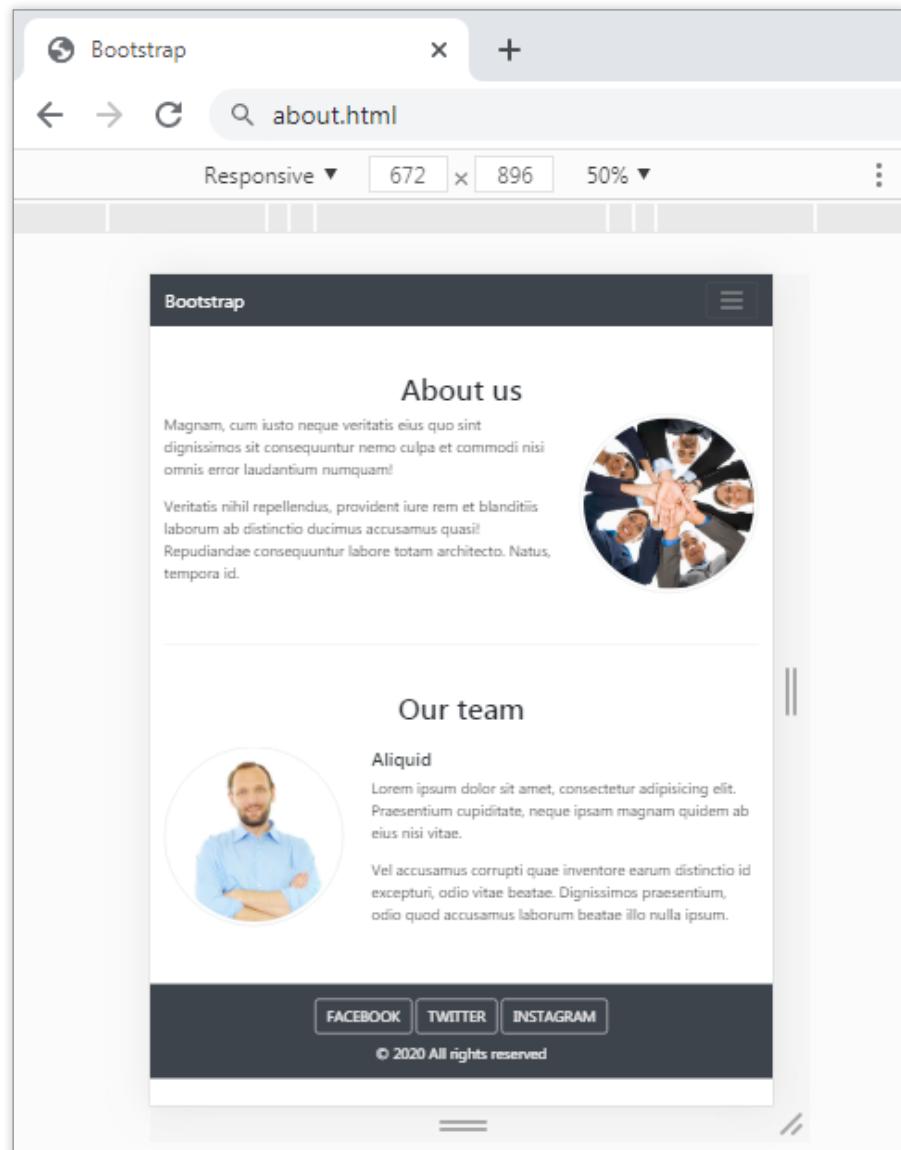


Рисунок 62. Отображение страницы «О нас» на мобильных устройствах и меньше

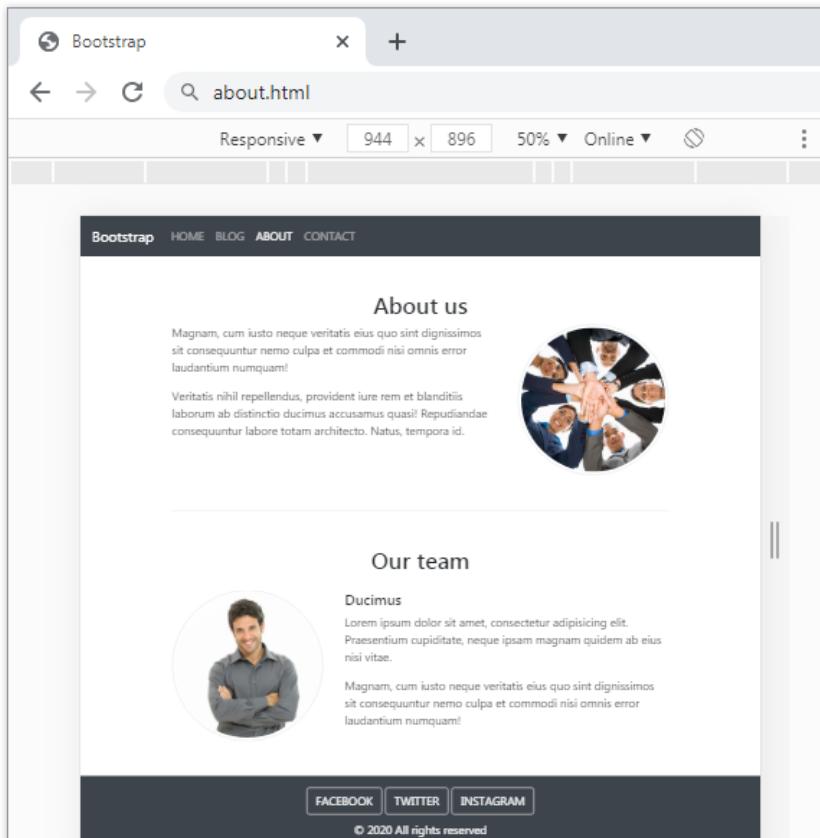


Рисунок 63. Отображение страницы «О нас» на планшетах и больших устройствах

Страница «Контакты» будет перестраиваться в две колонки на планшетах и больших устройствах. Элементами в первой колонке будут встраиваемая карта с соотношением сторон 21x9 и информация об адресе в виде списка. Вторая правая колонка должна содержать форму обратной связи.

При отображении на устройствах меньших чем планшет, правая колонка должна располагаться под левой, таким образом образуя одну колонку.

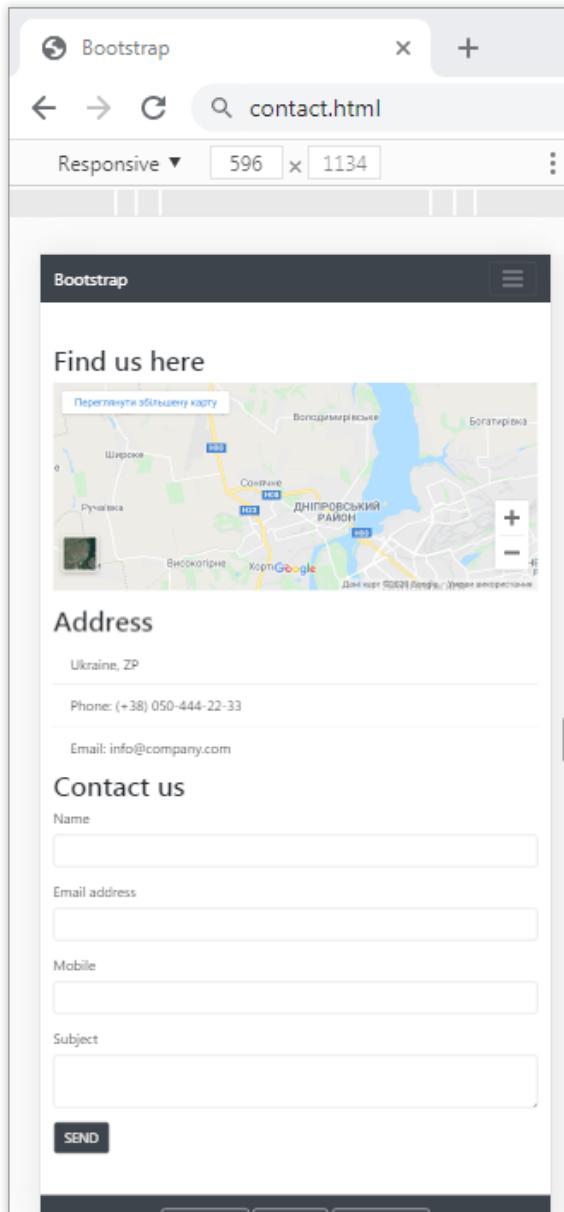


Рисунок 64. Отображение страницы на мобильных устройствах

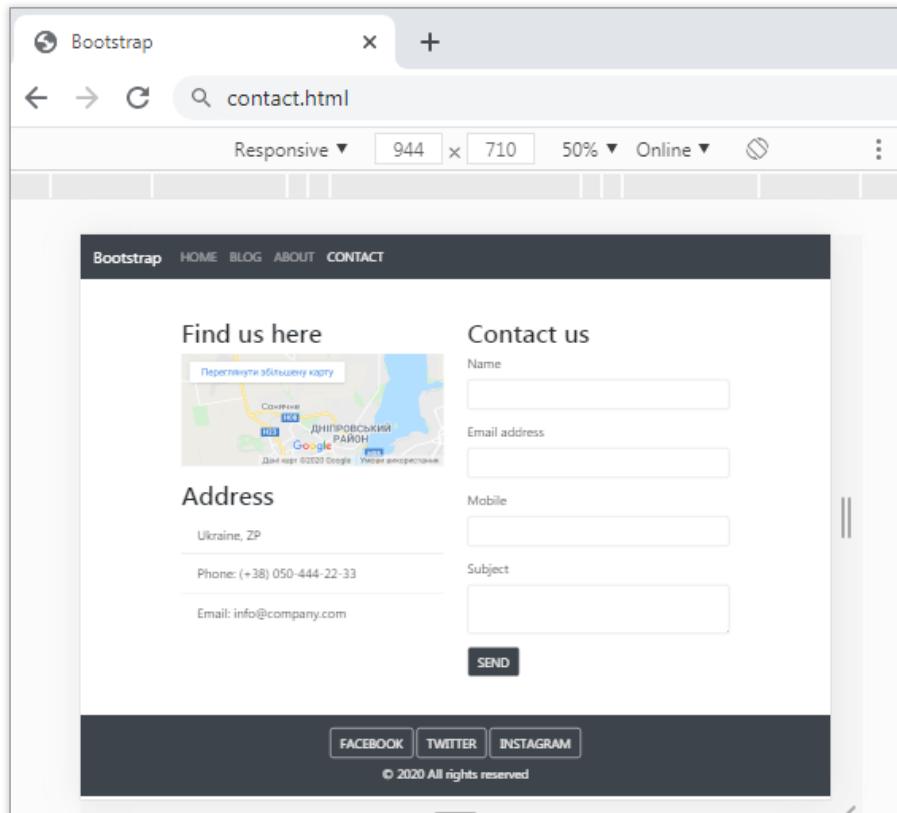


Рисунок 65. Отображение страницы на планшетах и больших устройствах

## Оглавление