

Advanced Web Programming

Lab2 - Bootstrap 5

Section 1. Bootstrap 5

Materials prepared on the basis of

<https://getbootstrap.com/docs/5.0/getting-started/download/>

and

<https://www.w3schools.com/bootstrap5/index.php>

Bootstrap is the most popular and powerful front-end (HTML, CSS, and JavaScript) framework for faster and easier responsive web development. It includes HTML and CSS based design templates for creating common user interface components like forms, buttons, navigations, dropdowns, alerts, modals, tabs, accordions, carousels, tooltips, and so on.

Bootstrap gives you ability to create flexible and responsive web layouts with much less efforts.

1. Getting Started with Bootstrap - Making HTML File as a Bootstrap Template

Now you're going to create a basic Bootstrap template by including the Bootstrap CSS and JS files via CDN. Bootstrap requires a third-party library [Popper.js](#) for some of its components like popovers and tooltips. You can either include it separately or simply include Bootstrap JS bundled with Popper.

I recommend adding Bootstrap in your project via CDN (Content Delivery Network) because CDN offers performance benefit by reducing the loading time, since they are hosting the files on multiple servers spread across the globe so that when a user requests the file it will be served from the server nearest to them. We're also using the CDN links in our examples:

```
<!DOCTYPE html>
```

```

<html lang="en">
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1">
  <title>Basic Bootstrap Template</title>
  <!-- Bootstrap CSS -->
  <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.
min.css" rel="stylesheet">
  <!-- Bootstrap js -->
  <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bu
ndle.min.js"></script>
</head>
<body>
  <h1>Welcome EFREI!</h1>
</body>
</html>

```

And we're all set! After adding the Bootstrap's CSS and JS files to our HTML page, we can begin to develop any responsive site or application with the Bootstrap framework.

Tip: If the visitor to your website has already downloaded the Bootstrap's CSS and JS files from the same CDN while visiting the other sites, it will be loaded from the browser's cache instead of re-downloading, which leads to faster loading time.

2. Define Bootstrap 5 Containers

Now I will describe the construction of the template scaffolding, i.e. the elements that form the skeleton of the website and inside which we will place other elements.

As you may have noticed, most websites on the Internet are not the full width of the browser window, but have empty spaces on the sides. Such a solution increases the readability of the page and is a standard practice when creating pages.

Creating Containers with Bootstrap

Containers are the most basic layout element in Bootstrap and are required when using the grid system. Containers are basically used to wrap content with some padding. They are also used to align the content horizontally center on the page in case of fixed width layout.

Bootstrap provides three different types containers:

- `.container`, which has a max-width at each responsive breakpoint.
- `.container-fluid`, which has 100% width at all breakpoints.



Creating Responsive Fixed-width Containers

```
<div class="container">
  <h1>This is a heading</h1>
  <p>This is a paragraph of
text.</p>
</div>
```

Creating Fluid Containers

```
<div class="container-fluid">
  <h1>This is a heading</h1>
  <p>This is a paragraph of text.</p>
</div>
```

Adding Background and Borders to Containers

By default, container doesn't have any `background-color` or `border`. But if you need you can apply your own styles, or simply use the Bootstrap background-color and border utility classes to add background-color or border on them, as shown in the following example.

```
<!-- Container with dark background and white text color -->
<div class="container bg-dark text-white">
  <h1>This is a heading</h1>
  <p>This is a paragraph of text.</p>
</div>
```

```
<!-- Container with light background -->
<div class="container bg-light">
  <h1>This is a heading</h1>
  <p>This is a paragraph of text.</p>
</div>
```

```
<!-- Container with border -->
<div class="container border">
  <h1>This is a heading</h1>
  <p>This is a paragraph of text.</p>
</div>
```

What is Bootstrap Grid System?

Bootstrap grid system provides an easy and powerful way to create responsive layouts of all shapes and sizes. It is built with flexbox with mobile-first approach. Also, it is fully responsive and uses twelve column system (12 columns available per row) and six default responsive tiers.

You can use the Bootstrap's predefined grid classes for quickly making the layouts for different types of devices like mobile phones, tablets, laptops, desktops, and so on. For example, you can use the `.col-*` classes to create grid columns for extra small devices like mobile phones in portrait mode, and the `.col-sm-*` classes for mobile phones in landscape mode.

Similarly, you can use the `.col-md-*` classes to create grid columns for medium screen devices like tablets, the `.col-lg-*` classes for devices like small laptops, the `.col-xl-*` classes for laptops and desktops, and the `.col-xxl-*` classes for large desktop screens.

The following table summarizes the key features of the Bootstrap's grid system.

| Features | X-Small (xs) | Small (sm) | Medium (md) | Large (lg) | X-Large (xl) | XX-Large (xxl) |
|-----------------------|--------------------|-----------------------|-----------------------|-----------------------|-----------------------|------------------------|
| Bootstrap Grid System | <576px | ≥576px | ≥768px | ≥992px | ≥1200px | ≥1400px |
| Container max-width | None (auto) | 540px | 720px | 960px | 1140px | 1320px |
| Class prefix | <code>.col-</code> | <code>.col-sm-</code> | <code>.col-md-</code> | <code>.col-lg-</code> | <code>.col-xl-</code> | <code>.col-xxl-</code> |
| Number of columns | 12 | | | | | |

The classes above can be combined to create more dynamic and flexible layouts.

Note: In a grid layout, content must be placed inside the columns (.col and .col-*) and only columns may be the immediate children of rows (.row). Also, rows should be placed inside a [container \(either fixed or fluid\)](#) for proper padding and alignment.

Creating Two Column Layouts

The following example will show you how to create two column layouts for medium, large and extra large devices like tables, laptops and desktops etc. However, on mobile phones (screen width less than 768px), the columns will automatically become horizontal (2 rows, 1 column).

```
<div class="container">

  <!--Row with two equal columns-->

  <div class="row">

    <div class="col-md-6">Column left</div>

    <div class="col-md-6">Column right</div>

  </div>

  <!--Row with two columns divided in 1:2 ratio-->

  <div class="row">

    <div class="col-md-4">Column left</div>

    <div class="col-md-8">Column right</div>

  </div>

  <!--Row with two columns divided in 1:3 ratio-->

  <div class="row">

    <div class="col-md-3">Column left</div>

    <div class="col-md-9">Column right</div>

  </div>

</div>
```

Since the Bootstrap grid system is based on 12 columns, therefore to keep the columns in a one line (i.e. side by side), the sum of the grid column numbers within a single row should not be greater than 12.

Creating Three Column Layouts

Similarly, you can create other layouts based on the above principle. For instance, the following example will typically create three column layouts for laptops and desktops screens. It also works in tablets in landscape mode if screen resolution is more than or equal to 992 pixels (e.g. Apple iPad). However, in portrait mode the grid columns will be horizontal as usual.

```
<div class="container">

  <!--Row with three equal columns-->

  <div class="row">

    <div class="col-lg-4">Column left</div>

    <div class="col-lg-4">Column middle</div>

    <div class="col-lg-4">Column right</div>

  </div>

  <!--Row with three columns divided in 1:4:1 ratio-->

  <div class="row">

    <div class="col-lg-2">Column left</div>

    <div class="col-lg-8">Column middle</div>

    <div class="col-lg-2">Column right</div>

  </div>

  <!--Row with three columns divided unevenly-->

  <div class="row">

    <div class="col-lg-3">Column left</div>

    <div class="col-lg-7">Column middle</div>

    <div class="col-lg-2">Column right</div>

  </div>
```

</div>

Note: If more than 12 grid columns are placed within a single row, then each group of extra columns, as a whole, will wrap onto a new line.

Column Wrapping Behavior

Now we are going to create more flexible layouts that changes the column orientation based on the viewport size. The following example will create a three column layout on large devices like laptops and desktops, as well as on tablets (e.g. Apple iPad) in landscape mode, but on medium devices like tablets in portrait mode ($768\text{px} \leq \text{screen width} < 992\text{px}$), it will change into a two column layout where the third column moves at the bottom of the first two columns.

```
<div class="container">
  <div class="row">
    <div class="col-md-4 col-lg-3">Column one</div>
    <div class="col-md-8 col-lg-6">Column two</div>
    <div class="col-md-12 col-lg-3">Column three</div>
  </div>
</div>
```

As you can see in the example above the sum of the medium grid column numbers (i.e. `col-md-*`) is $3 + 9 + 12 = 24 > 12$, therefore the third `<div>` element with the class `.col-md-12` that is adding the extra columns beyond the maximum 12 columns in a `.row`, gets wrapped onto a new line as one contiguous unit on the medium screen size devices.

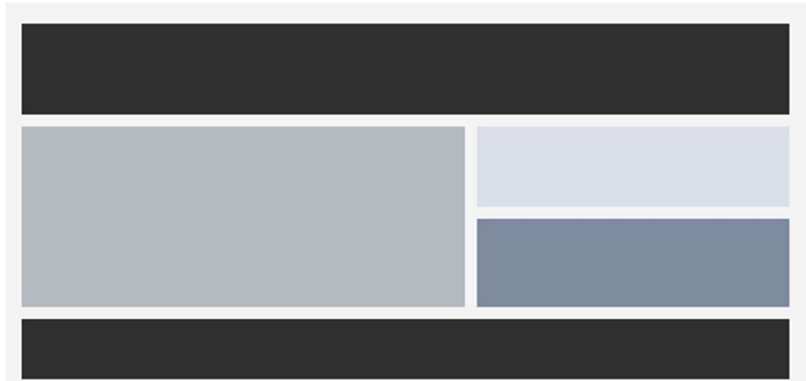
Similarly, you can create even more adaptable layouts for your websites using the Bootstrap's grid column wrapping feature.

Exercise 1.

Two Column Website Layouts Tablets, Desktops and Large Screens

Here're the examples of two column website layouts for tablets (portrait and landscape), desktops and large desktops.

These layouts will become horizontal i.e. changed to one column layout on mobile devices.

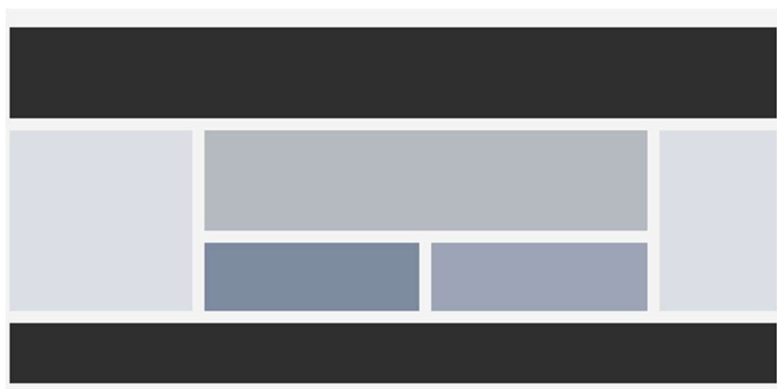


Three Column Website Layouts Tablets (landscape), Desktops and Large Screens

Here're the examples of three column website layouts for desktops and large desktops as well as for tablets (e.g. Apple iPad) in landscape mode. These layouts will become horizontal i.e. changed to one column layout on mobiles and tablets in portrait mode.



The more difficult Exercise.



Exercise 2. You can customize the layout for extra extra large devices such as a large desktop screen by adding the class `.col-xxl-3` on each column, as every row in that layout contains 4 columns (i.e. 4x3 grids layout)

Bootstrap Responsive Layout

| | | |
|-------|-------|-------|
| Box 1 | Box 2 | Box 3 |
| Box 4 | Box 5 | Box 6 |
| Box 7 | Box 8 | Box 9 |

Bootstrap Responsive Layout

| | |
|-------|-------|
| Box 1 | Box 2 |
| Box 3 | Box 4 |
| Box 5 | Box 6 |

Bootstrap Responsive Layout

| |
|-------|
| Box 1 |
| Box 2 |
| Box 3 |

Creating Navbar with Bootstrap

You can use the Bootstrap navbar component to create responsive navigation header for your website or application. These responsive navbar will be collapsed on devices having small viewports like mobile phones but expand when user click the toggle button. However, it will be horizontal as normal on the medium and large devices such as laptop or desktop.

You can also create different variations of the navbar such as navbars with dropdown menus and search boxes as well as fixed positioned navbar with much less effort. The following example will show you how to create a simple static navbar with navigation links.

```
<nav class="navbar navbar-expand-lg navbar-light bg-light">
  <div class="container-fluid">
    <a href="#" class="navbar-brand">Brand</a>
    <button type="button" class="navbar-toggler" data-bs-toggle="collapse" data-bs-
target="#navbarCollapse">
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarCollapse">
      <div class="navbar-nav">
        <a href="#" class="nav-item nav-link active">Home</a>
        <a href="#" class="nav-item nav-link">Profile</a>
        <a href="#" class="nav-item nav-link">Messages</a>
        <a href="#" class="nav-item nav-link disabled" tabindex="-1">Reports</a>
      </div>
      <div class="navbar-nav ms-auto">
        <a href="#" class="nav-item nav-link">Login</a>
      </div>
    </div>
  </div>
</nav>
```

Brand Home Profile Messages Reports

Login

Brand



Brand



Home

Profile

Messages

Reports

Login

Adding Logo Images to Navbars

You can also place your logo image inside the navbar, instead of plain text. However, you need to set the logo height manually to fit it properly inside the navbar, as shown here:

```
<nav class="navbar navbar-expand-lg navbar-light bg-light">
  <div class="container-fluid">
    <a href="#" class="navbar-brand">
      
    </a>
    <button type="button" class="navbar-toggler" data-bs-toggle="collapse" data-bs-target="#navbarCollapse">
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarCollapse">
      <div class="navbar-nav">
```

```

    <a href="#" class="nav-item nav-link active">Home</a>
    <a href="#" class="nav-item nav-link">Profile</a>
    <a href="#" class="nav-item nav-link">Messages</a>
    <a href="#" class="nav-item nav-link disabled" tabindex="-1">Reports</a>
  </div>

  <div class="navbar-nav ms-auto">
    <a href="#" class="nav-item nav-link">Login</a>
  </div>
</div>
</div>
</nav>

```



Adding Dropdowns to Navbar

You can also include dropdowns and search box within the navbars, as shown here:

```

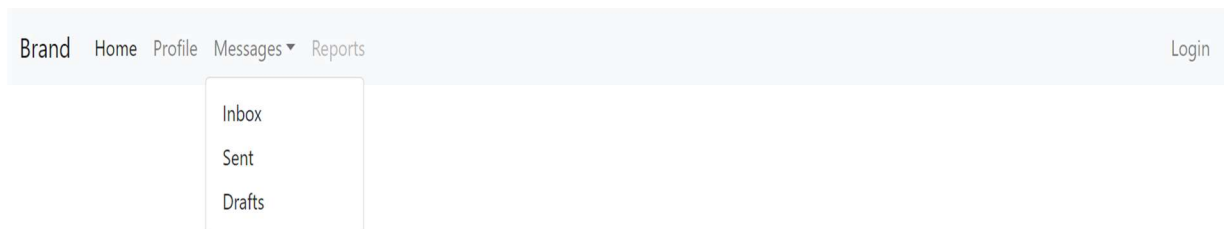
<nav class="navbar navbar-expand-lg navbar-light bg-light">
  <div class="container-fluid">
    <a href="#" class="navbar-brand">Brand</a>
    <button type="button" class="navbar-toggler" data-bs-
toggle="collapse" data-bs-target="#navbarCollapse">
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarCollapse">
      <div class="navbar-nav">
        <a href="#" class="nav-item nav-link active">Home</a>
        <a href="#" class="nav-item nav-link">Profile</a>
        <div class="nav-item dropdown">
          <a href="#" class="nav-link dropdown-toggle" data-
bs-toggle="dropdown">Messages</a>
          <div class="dropdown-menu">
            <a href="#" class="dropdown-item">Inbox</a>

```

```

        <a href="#" class="dropdown-item">Sent</a>
        <a href="#" class="dropdown-item">Drafts</a>
      </div>
    </div>
    <a href="#" class="nav-item nav-link disabled"
tabindex="-1">Reports</a>
  </div>
  <div class="navbar-nav ms-auto">
    <a href="#" class="nav-item nav-link">Login</a>
  </div>
</div>
</div>
</nav>

```



Changing the Color Scheme of Navbars

You can also change the color scheme of the navbar by using the `.navbar-light` for the light background colors, or `.navbar-dark` for the dark background colors. Then, customize it with the background color utility classes, such as `.bg-dark`, `.bg-primary`, and so on.

Alternatively, you can also apply the CSS `background-color` property on the `.navbar` element yourself to customize the navbar theme, as shown in the following example:

```

<nav class="navbar navbar-dark bg-dark">

  <!-- Navbar content -->

</nav>

<nav class="navbar navbar-dark bg-primary">

  <!-- Navbar content -->

</nav>

```

```
<nav class="navbar navbar-light" style="background-color: #ddeeff;">

  <!-- Navbar content -->

</nav>
```

Bootstrap Fixed Navbars

Bootstrap also provides mechanism to create navbar that is fixed to the top, fixed to the bottom, or stickied to the top (i.e. scrolls with the page until it reaches the top, then stays there).

Navbar Fixed to the Top

Apply the position utility class `.fixed-top` to the `.navbar` element to fix the navbar at the top of the viewport, so that it won't scroll with the page. Here's an example:

```
<nav class="navbar fixed-top navbar-expand-lg navbar-dark bg-dark">
  <!-- Navbar content -->
</nav>
```

Navbar Stickied to the Top

You can also create sticky top navbar that scrolls with the page until it reaches the top, then stays there, by simply using the `.sticky-top` class on the `.navbar` element, like this:

```
<nav class="navbar sticky-top navbar-expand-lg navbar-dark bg-dark">
  <!-- Navbar content -->
</nav>
```

Using Icons in Bootstrap 5

Bootstrap now includes over 1,300 high quality icons, which are available in SVGs, SVG sprite, or web fonts format. You can use them with or without Bootstrap in any project. The advantage of using font icons is, you can create icons of any color just through applying the CSS `color` property. Also, to change the size of icons you can simply use the CSS `font-size` property.

Now, let's see how to include and use Bootstrap icons on a web page.

Including Bootstrap Icons in a Web Page

The simplest way to include Bootstrap icons in a web page is using the CDN link. This CDN link basically points to a remote CSS file that includes all the necessary classes to generate font icons.

You can include Bootstrap icons in a Bootstrap template as well as in a simple web page without using the Bootstrap framework. Let's take a look at the following example:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1">
  <title>Including Bootstrap Icons in HTML</title>
  <!-- Bootstrap CSS -->
  <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@
5.1.3/dist/css/bootstrap.min.css">
  <!-- Bootstrap Font Icon CSS -->
  <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap-
icons@1.5.0/font/bootstrap-icons.css">
</head>
<body>
  <h1><i class="bi-globe"></i> Hello, world!</h1>
</body>
</html>
```

How to Use Bootstrap Icons in Your Code

To use Bootstrap icons in your code you'll require an `<i>` tag with an individual icon class `.bi-*` applied on it. The general syntax for using Bootstrap icons is:

```
<i class="bi-class-name"></i>
```

Where ***class-name*** is the name of the particular icon class, e.g. search, person, calendar, star, globe, facebook, twitter, and so on.

See the list of all Bootstrap icons classes -> <https://icons.getbootstrap.com>

Using Font Awesome Icons in Bootstrap

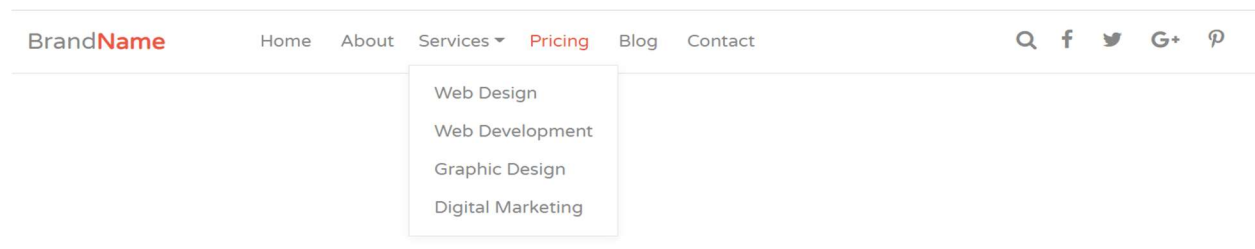
You can also use external icon libraries in Bootstrap. One of the most popular and highly compatible external icon library for Bootstrap is Font Awesome. It provides over 675 icons which are available in SVG, PNG, as well as in web font format for better usability and scalability.

You can simply use the freely available font-awesome CDN link to include it in your project. Let's take a look at the following example to understand how it basically works:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1">
  <title>Including Font Awesome Icons in Bootstrap</title>
  <!-- Bootstrap CSS -->
  <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.
min.css">
  <!-- Font Awesome CSS -->
  <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/font-awesome/4.7.0/css/font-
awesome.min.css">
</head>
<body>
  <h1><i class="fa fa-globe"></i> Hello, world!</h1>
</body>
</html>
```

See the list of all Font Awesome Icons -> <https://fontawesome.com/v5/icons/users-class?s=solid>

Exercise 3. Create navbar looks like this.



Let this navbar will be sticky to the top. You should add to the pages Samone of section with height is more then page height.

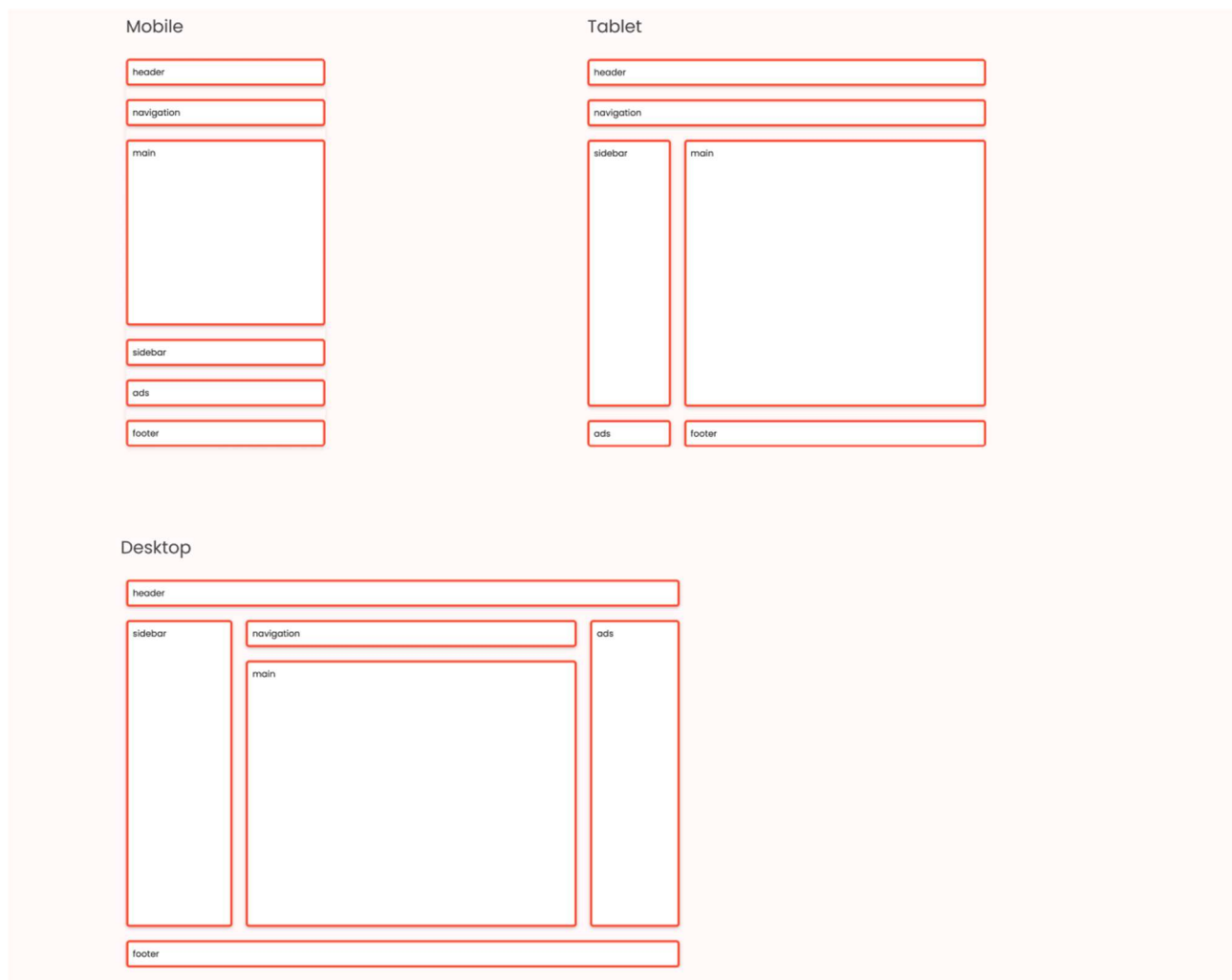
Now own work - see the other components available in Bootstrap5.

Exercise 4. In directory exercise you find example one-land page. Try to create this landpage by own using my example as inspiration.

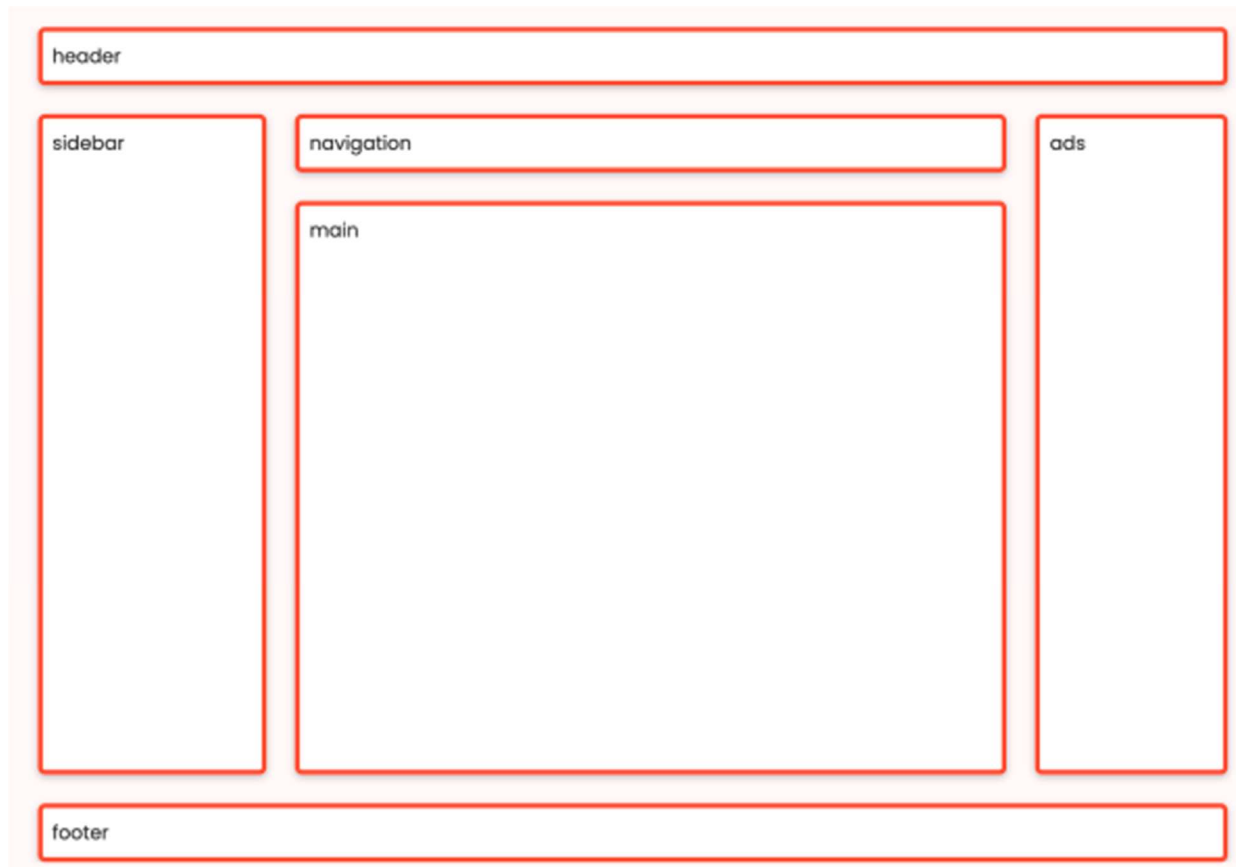
Homework. (max 10pkt)

Design a simple landing page. Let yours page will be RWD (3pkt).

Below you see how should



looks layout for different size of viewport. Page consist of header, navigation, sidebar, main area, ads and footer.



Header consist of image banner with add text on the top of banner. (1pkt)

Ads consists of some of advertising (two or three) (1pkt)

Let sidebar is fixed and contains single card with personal data. (1 pkt)

SIDEBAR

Home

About Me

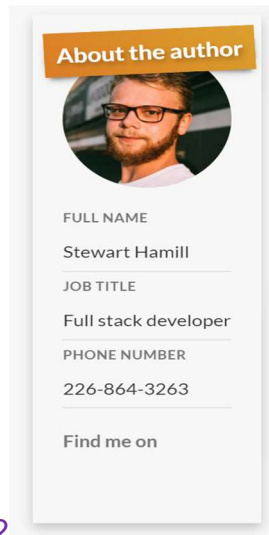
Portfolio

Contact Me



Anna Kowalska

example 1



Example2

Let navigation will be always in viewport. (1pkt)

Tips. Check **position: sticky** properties.

Main have 4 sections which a linked to navigation items. When you click in item the screen content is scrolled to selected section. (1 pk)

Tips. Use **scroll-behavior: smooth**

Minimum one section should contain gallery of pictures.

A sample one item in this section should look like the following



Lorem ipsum dolor sit amet.

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

In one there is a form about knowing web technology. (1pkt)

1 pkt for design effect.