Responsive Web Development With Bootstrap

Course Content

Unit-1

Bootstrap Scaffolding: Introduction to Bootstrap , file structure, global styles, default Grid System, Fluid Grid system ,Container Layouts, Responsive Design.

Unit-2

Bootstrap CSS : Typography, Code, tables, forms, buttons, images, icons

Unit-3

Bootstrap Layout Components: Dropdown Menus, Button Groups, Button with dropdowns, Navigation Elements, Navbar, Pagination, Labels and Badges, Typographic Elements

Unit-4

Bootstrap JavaScript Plugins: Programmatic API, Transitions, Modal, Dropdown, Scrollspy,Toggleable Tabs, Tooltips, Popover, Alerts, Buttons, Collapse.

Unit-5

Carousel, Typeahead, **Using Bootstrap:** Customizing Bootstrap using LESS, text snippets, themes, Built with Bootstrap.

Bootstrap

Bootstrap is an **open source product** from **Mark Otto and Jacob Thornton**, when it was initially released, were both employees at Twitter.

There was a need to **standardize the frontend toolsets of engineers across the company**.

Most popular and used front-end framework with HTML and CSS templates for UI interface elements such as buttons and forms.

With bootstrap you can build complex models in no time which is easy and customizable.

Why we need to use it?

- **Responsive Grid-** It has a responsive layout and grid system which dynamically aligns with screen resolutions.
- **Responsive images-** It resizes based on current screen resolution by adding a responsive class to it.
- **Community-**being an open source framework it has amazing support the developers and contributors. This helps to use it widely without any trouble.

Launch: Introduced in August 2011, Bootstrap quickly gained popularity.

Evolution: Originally a CSS framework, it has expanded to include JavaScript plugins and icons.

Responsive Design: Enables responsive web design, ensuring websites look good on all devices.

Grid System: Features a robust 12-column grid layout, standardizing design across various screen sizes.

Customization: Offers a build tool on its website for customizing CSS and JavaScript features according to project needs.

Frontend Development: Streamlines frontend development by providing a solid foundation for design and coding practices.

Ease of Use: Getting started is straightforward—simply include the necessary CSS and JavaScript files in your project.

Community Support: A large community and extensive documentation support users in leveraging Bootstrap effectively.

Bootstrap

Build responsive, mobile-first projects on the web with the world's most popular front-end component library.

Bootstrap is an open source toolkit for developing with HTML, CSS, and JS. Quickly prototype your ideas or build your entire app with our Sass variables and mixins, responsive grid system, extensive prebuilt components, and powerful plugins built on jQuery.

Get started Download

Currently v4.3.1

Bootstrap.css

Bootstrap.css is a **CSS framework that arranges and manages the layout of a website.** While HTML works with the content and structure of a web page, CSS deals with the layout itself.

Bootstrap.js

This file is the core part of Bootstrap. **It consists of JavaScript files that are responsible for the website's interactivity.**

Glyphicons

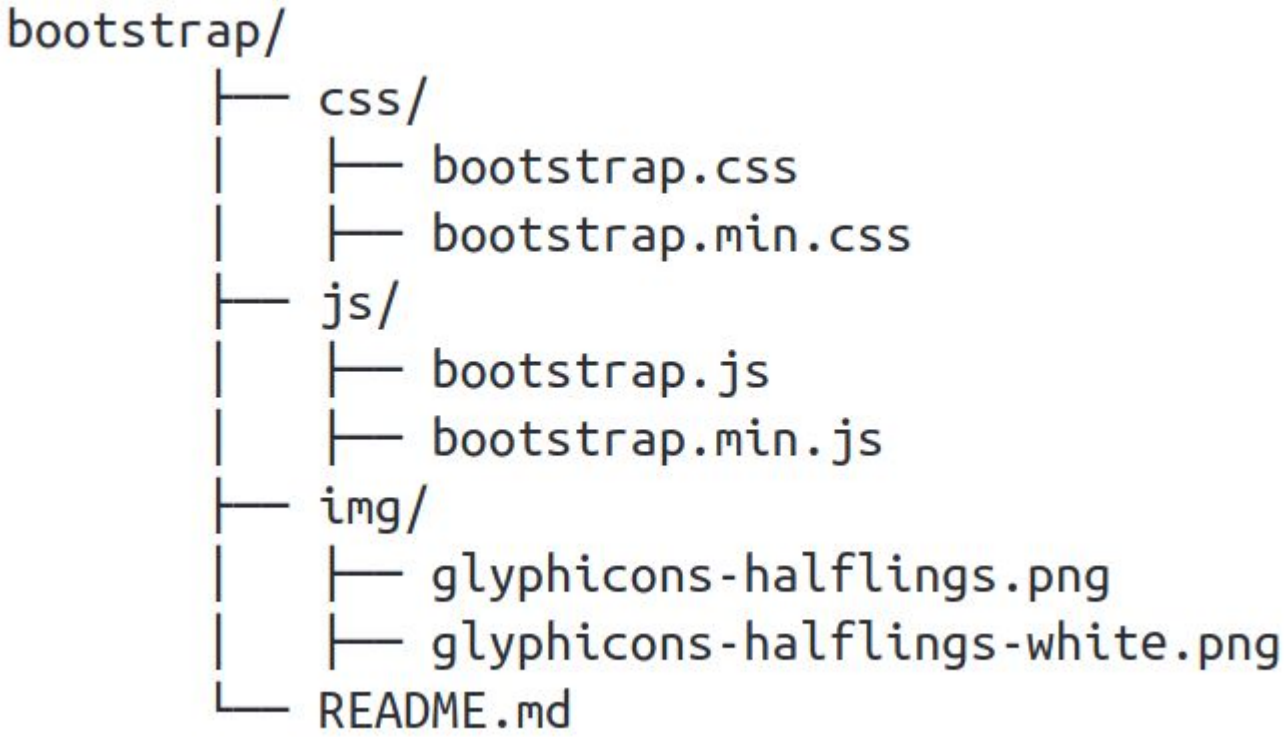
Icons are an **integral part of the front-end of a website**, as they often **display actions and data within the user interface.**

Bootstrap uses icons called Glyphicons, which include a Glyphicons Halflings set. Although the design is basic, they perform their essential functions, and they're free to use.

```
<html lang="en">

<head>
<meta charset="utf-8" />
<meta http-equiv="X-UA-Compatible" content="IE=edge" />
<meta name="viewport" content="width=device-width, initial-scale=1" />
<title>Bootstrap 101 Template</title>
<link href="css/bootstrap.min.css" rel="stylesheet" />
</head>

<body>
<h1>Hello, world!</h1>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/
jquery.min.js"></script>
<script src="js/bootstrap.min.js"></script>
</body>

</html>
```

- Using **Bootstrap via a CDN (Content Delivery Network)** allows you to easily include Bootstrap's CSS and JavaScript files in your project without hosting them locally.
- A **CDN** (Content Delivery Network) is a network of servers distributed across various geographical locations, designed to deliver content to users more efficiently.

Bootstrap Structure



- The Bootstrap download includes three folders: css, js, and img.
- For simplicity, add these to the root of your project.
- Minified versions of the CSS and JavaScript are also included.
- It is not necessary to include both the uncompressed and the minified versions.

Normally, a web project looks something like this:

Basic HTML Template

```
<!DOCTYPE html>
<html>
        <head>
                <title>Bootstrap 101 Template</title>
        </head>
        <body>
                <h1>Hello, world!</h1>
        </body>
</html>
```

With Bootstrap, we include the link to the CSS stylesheet and the JavaScript:

```
<!DOCTYPE html>
<html>
        <head>
                <title>Bootstrap 101 Template</title>
                <link href="css/bootstrap.min.css" rel="stylesheet">
        </head>
        <body>
                <h1>Hello, world!</h1>
                <script src="js/bootstrap.min.js"></script>
        </body>
</html>
```

Bootstrap 3 version

```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Bootstrap demo</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet"
integrity="sha384-QWTKZyjpPEjISv5WaRU9OFeRpok6YctnYmDr5pNlyT2bRjXh0JMhjY6hW+ALEwIH" crossorigin="anonymous">
  </head>
  <body>
    <button class="btn btn-primary">
    <span class="glyphicon glyphicon-star"></span> Star
</button>
  </body>
</html>
```

Bootstrap 5 version

```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Bootstrap demo</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet"
integrity="sha384-QWTKZyjpPEjISv5WaRU9OFeRpok6YctnYmDr5pNlyT2bRjXh0JMhjY6hW+ALEwIH" crossorigin="anonymous">
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.11.3/font/bootstrap-icons.min.css">

  </head>
  <body>
   <i class="bi bi-1-circle"></i>
  </body>
</html>
```

Global styles:

1. Normalization

Use of Normalize.css:

- Normalize.css helps standardize styles across different browsers, making sure that elements look more consistent. This means you won't have to deal with variations in default styling, such as margins, padding, and font sizes.

Example: If you have a heading element (<h1>), it might have different default margins in different browsers. Normalize.css ensures it has consistent styling.

Link: https://getbootstrap.com/docs/3.4/css/#overview-normalize

Normalize.css

For improved cross-browser rendering, we use Normalize.css, a project by Nicolas Gallagher and Jonathan Neal.

Containers

Bootstrap requires a containing element to wrap site contents and house our grid system. You may choose one of two containers to use in your projects. Note that, due to `padding` and more, neither container is nestable.

Use `.container` for a responsive fixed width container.

```
<div class="container">
  ...
</div>
```

Use `.container-fluid` for a full width container, spanning the entire width of your viewport.

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

2. Body Styles

Margin Removal:

- **What it does**: Bootstrap removes the default margin from the <body> element. This allows content to extend to the edges of the viewport.

Example:

```
<body>
    <h1>Hello, Bootstrap!</h1>
    <p>This content is flush against the edges of the browser window.</p>
</body>
```

In this example, the heading and paragraph will appear without any margin around the body.

Background Color:

- **What it is**: The default background color of the body is set to white.

Example:

```
<body style="background-color: white;">
    <h1>Welcome!</h1>
    <p>This background is white by default.</p>
</body>
```

3. Typography: This is in Bootstrap refers to the set of styles applied to text elements to ensure they are visually appealing, readable, and consistent across the web application. Bootstrap provides a robust typographic system that includes various settings for font sizes, weights, line heights, and other attributes.

Base Typography Settings:

- Bootstrap uses variables like @baseFontFamily, @baseFontSize, and @baseLineHeight to create a typographic system.

```
<body>
    <h1>This is a heading</h1>
    <p>This is a paragraph with consistent line height and font size.</p>
</body>
```

In this example, all text adheres to Bootstrap's typographic settings, ensuring a cohesive look.

4. Link Styles

Global Link Color: Bootstrap defines a default link color using the variable @linkColor.

This is a link

The link will be styled in the default Bootstrap link color (usually a shade of blue).

Hover Effects:

- Links are underlined only when hovered over, reducing visual clutter.

Example:

Hover over this link

In this case, the link will only show an underline when hovered, making it clear that it’s interactive.

```html
<!doctype html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Bootstrap Global Styles Example</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet">
</head>
<body>
    <div class="container">
        <h1>Welcome to Bootstrap</h1>
        <p>This paragraph shows Bootstrap's default styling. The text has a consistent font size and line height, thanks to Bootstrap's typographic base.</p>
        <a href="#" class="link-primary">This is a primary link. Hover to see the underline effect!</a>
    </div>
</body>
</html>
```

file:///C:/Users/Admin/OneDrive/Desktop/2.html

Welcome to Bootstrap

This paragraph shows Bootstrap's default styling. The text has a consistent font size and line height, thanks to Bootstrap's typographic base.

This is a primary link. Hover to see the underline effect!

Example 1:

```
<!doctype html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Bootstrap Global Styles Example</title>
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css"
rel="stylesheet">
</head>
<body>
    <div class="container">
    <h1>Welcome to Bootstrap</h1>
```

```
<p>This paragraph shows Bootstrap's default styling. The text has a consistent font size and line height, thanks to Bootstrap's typographic base.This paragraph shows Bootstrap's default styling. The text has a consistent font size and line height, thanks to Bootstrap's typographic base.This paragraph shows Bootstrap's default styling. The text has a consistent font size and line height, thanks to Bootstrap's typographic base.</p>

    <a href="#" class="link-primary">This is a primary link. Hover to see the underline effect!</a>

    </div>

</body>

</html>
```


file:///C:/Users/Admin/OneDrive/Desktop/2.html
Welcome to Bootstrap
This paragraph shows Bootstrap's default styling. The text has a consistent font size and line height, thanks to Bootstrap's typographic base.This paragraph shows Bootstrap's default styling. The text has a consistent font size and line height, thanks to Bootstrap's typographic base.This paragraph shows Bootstrap's default styling. The text has a consistent font size and line height, thanks to Bootstrap's typographic base.
This is a primary link. Hover to see the underline effect!

Example 2:

```
<!doctype html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Bootstrap Margin Example</title>
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css"
rel="stylesheet">
</head>
<body>
    <div class="container">
    <h1>Welcome to My Bootstrap Page</h1>
```

```
<p>This paragraph is flush against the edges of the browser window. There is no margin around the body, making the content look snug and clean.</p>

    <button class="btn btn-primary">Click Me!</button>

    </div>

</body>

</html>
```


file:///C:/Users/Admin/OneDrive/Desktop/2.html
Welcome to My Bootstrap Page
This paragraph is flush against the edges of the browser window. There is no margin around the body, making the content look snug and clean.
Click Me!

Default Grid System

- **12 Columns**: The grid is based on a 12-column layout, allowing for various combinations of columns.
- **Container Widths**: Without responsive features, the default container width is 940px. With responsive CSS, it adjusts to either 724px or 1170px based on the viewport size.
- **Responsive Behavior**: Below 767px (typical for tablets and smaller devices), columns stack vertically, making it easier to read on smaller screens.
- **Column Width**: At the default width, each column is 60 pixels wide.
- **Offsets**: Columns have a 20-pixel offset to the left, ensuring proper spacing and alignment.
- **Fluid Columns**: The grid is designed to be fluid, adapting to different screen sizes while maintaining structure.
    - **Example:** Using classes like **`.col-md-4 and .col-sm-6,`** you can specify different layouts for different screen sizes, ensuring columns resize and stack as needed.

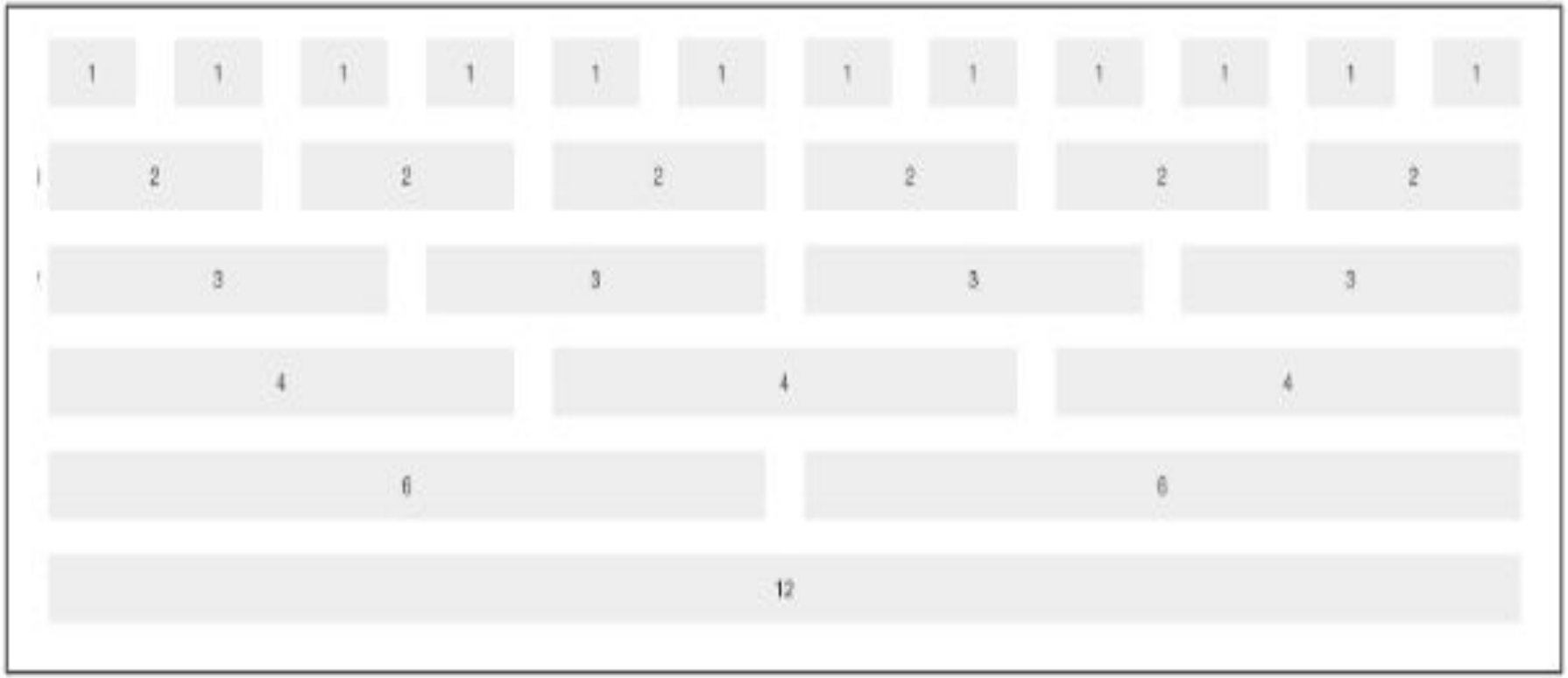


Figure 1-1. Default grid

- Example: On a desktop, you might have a layout like this:

```
| Col 1 | Col 2 | Col 3 |
```

On a tablet (below 767px), it stacks:

```
| Col 1 |
| Col 2 |
| Col 3 |
```

Example: If you want to create a layout with an offset, you could do:

```
|    Col 1    | Col 2 | Col 3 |
```

Here, Col 1 is offset by 20 pixels to the left.

```
<div class="container">
  <div class="row">
    <div class="col-md-4">Column 1</div>
    <div class="col-md-4">Column 2</div>
    <div class="col-md-4">Column 3</div>
  </div>
</div>
```

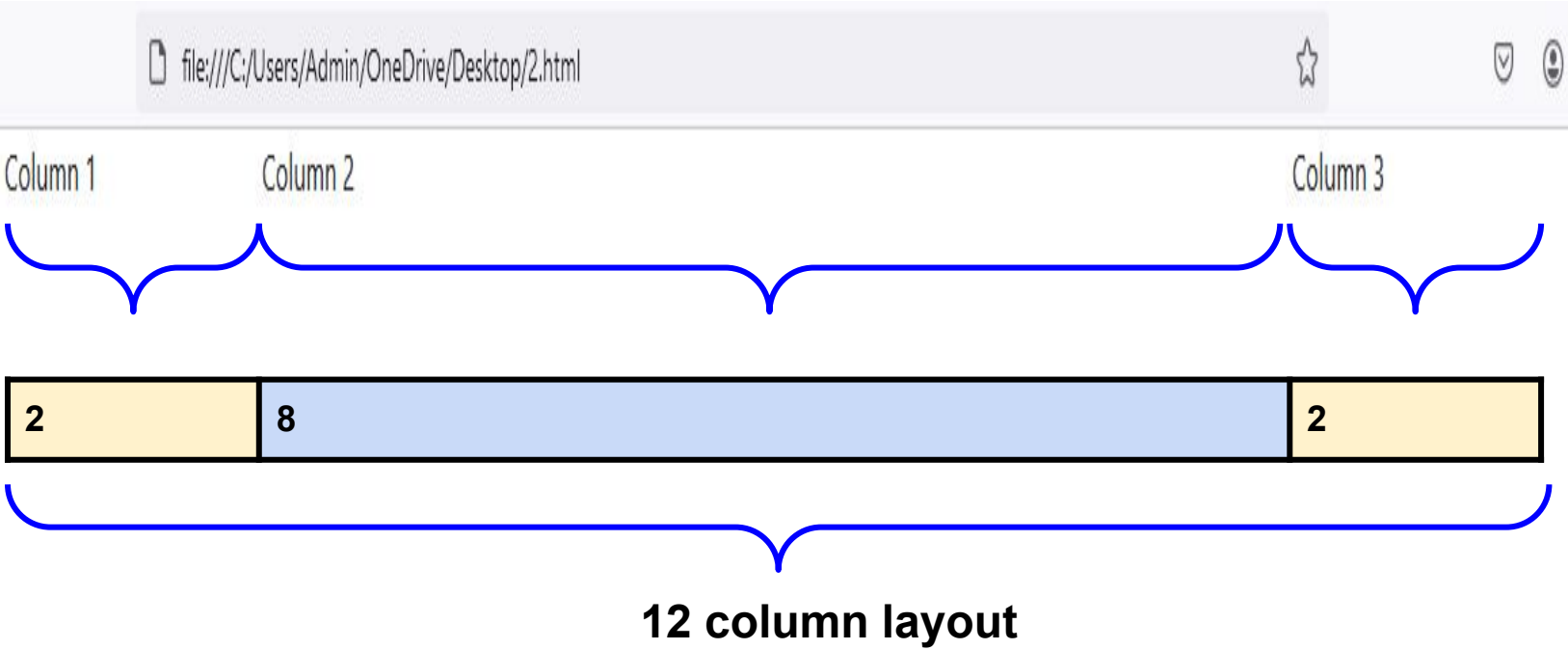
On larger screens: This will display three equal columns side by side.
On screens smaller than 768px: The columns will stack vertically.

Basic Grid HTML

- To create a simple layout, create a container with a ** ** that has a class of .row and add the appropriate amount of **.col-md*** columns.
- Since we have a 12-column grid, we just need the amount of **.col-md*** columns to equal 12. We could use a 3-6-3 layout, 4-8, 3-5-4, 2-8-2…

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Bootstrap Grid Example</title>
    <!-- Bootstrap CSS -->
    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
</head>
<body>
<div class="container">
    <div class="row">
```

```
  <div class="col-md-2">Column 1</div>
        <div class="col-md-8">Column 2</div>
        <div class="col-md-2">Column 3</div>
        </div>
</div>
</body>
</html>
```


file:///C:/Users/Admin/OneDrive/Desktop/2.html
Column 1
Column 2
Column 3
2
8
2
12 column layout

Offsetting Columns

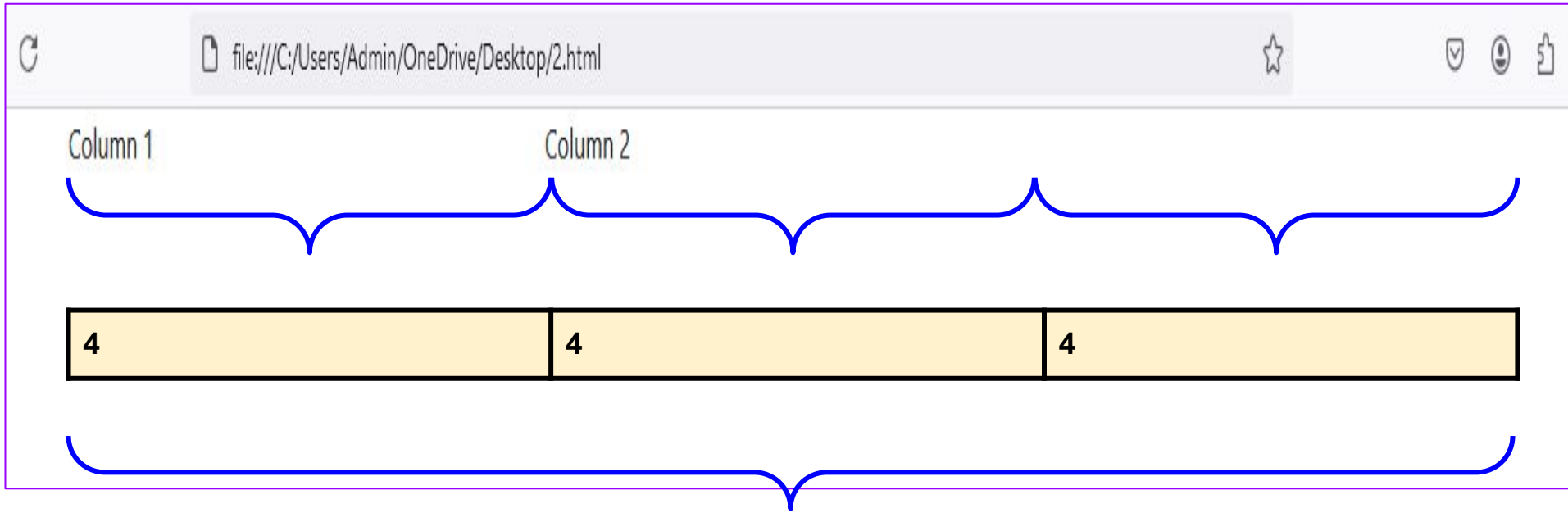
You can move columns to the right using the .offset* class. Each class moves the span over that width.

```
<div class="container">
    <div class="row">
    <div class="col-md-4">Column 1</div>
    <div class="col-md-4 offset-md-2">Column 2 (Offset by 2)</div>
    </div>
```

Example without offset

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Bootstrap Grid Example</title>
    <!-- Bootstrap CSS -->
    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
</head>
<body>
<div class="container">
    <div class="row">
```

```
  <!-- Column 1 (no offset) -->
        <div class="col-md-4">Column 1</div>
        <!-- Column 2  (no offset) -->
        <div class="col-md-4">Column 2</div>
        </div>
</div>
</body>
</html>
```

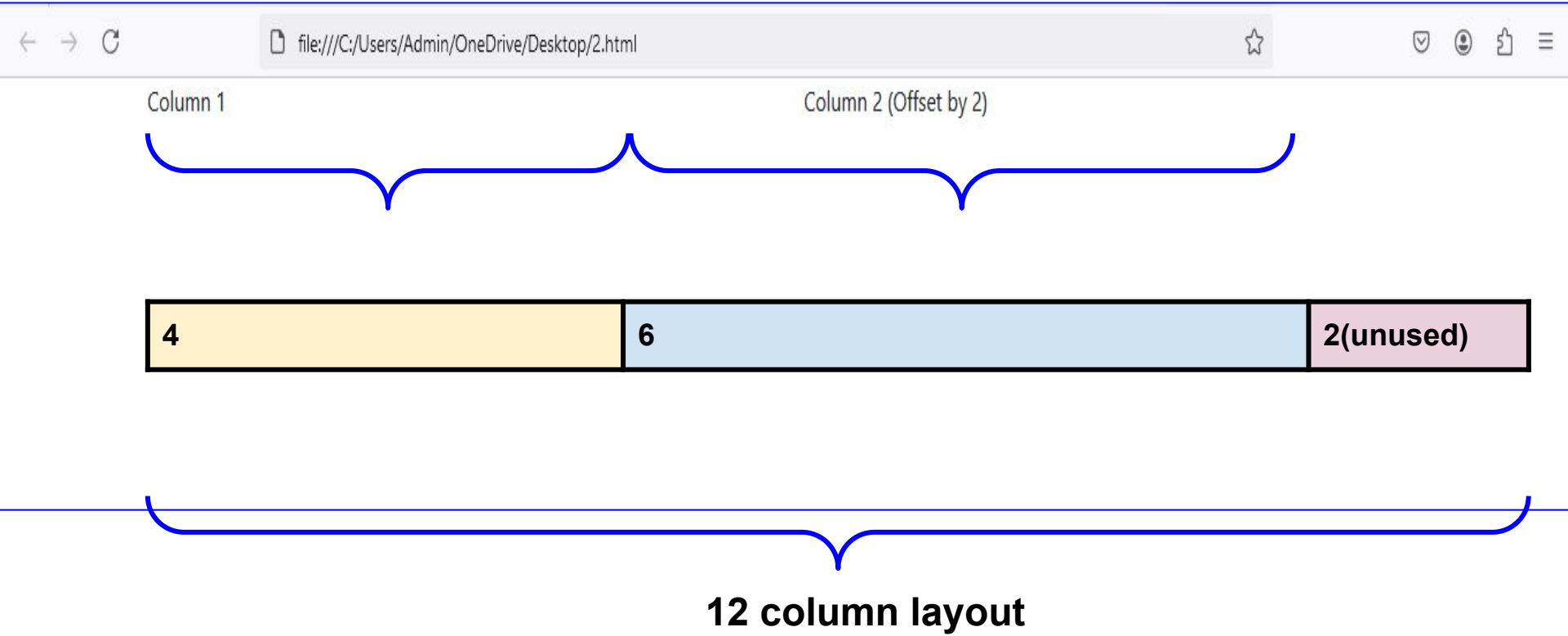


12 column layout

Example with offset

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Bootstrap Grid Example</title>
    <!-- Bootstrap CSS -->
    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
</head>
<body>
<div class="container">
    <div class="row">
```

```
  <!-- Column 1 (no offset) -->
        <div class="col-md-4">Column 1</div>
        <!-- Column 2 (offset by 2 columns) -->
        <div class="col-md-4 offset-md-2">Column 2 (Offset by 2)</div>
        </div>
</div>
</body>
</html>
```


file:///C:/Users/Admin/OneDrive/Desktop/2.html
Column 1
Column 2 (Offset by 2)
4
6
2(unused)
12 column layout

Nesting Columns

In Bootstrap 5, to nest your content within the default grid, simply add a new .row inside a .**col-md-*.** Ensure that the total column spans in the nested row equal the number of columns in the parent container. For example, if the parent uses .col-md-4, the nested row can use columns that add up to 4.

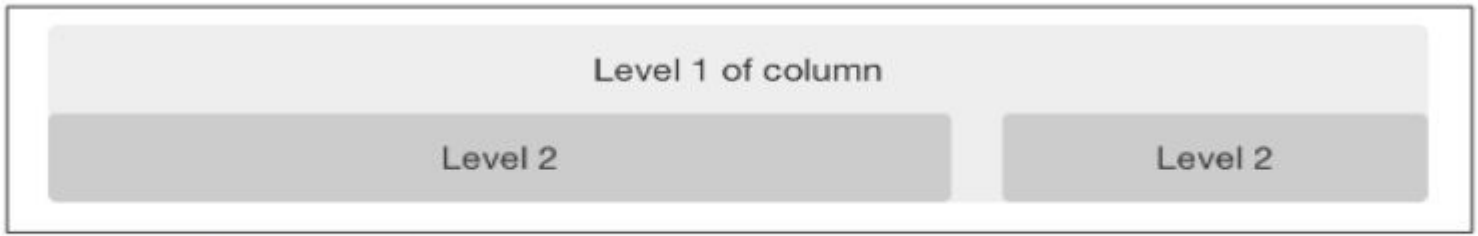


Figure 1-3. Nesting grid

```
<!DOCTYPE html>
<html lang="en">
<head>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <title>Bootstrap Nested Columns Example</title>
        <!-- Bootstrap CSS -->
        <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
</head>
<body>
<div class="container">
        <div class="row">
        <div class="col-md-4">Column 1</div>
```

```
<div class="col-md-4">Column 2
        <div class="row">
        <div class="col-md-6">Nested Column 1</div>
        <div class="col-md-6">Nested Column 2</div>
        </div>
    </div>
    <div class="col-md-4">Column 3</div>
    </div>
</div>
</body>
</html>
```

Column 1- 4
Column 2- 4 –nested column 1- 6
–nested column 2- 6
(nested columns size is 6+6=12 which exceeds the parent column i.e., column 2 is 4. So, automatically it adjusts to parent column size)
Column 3- 4

File C:/Users/student/Desktop/1.html

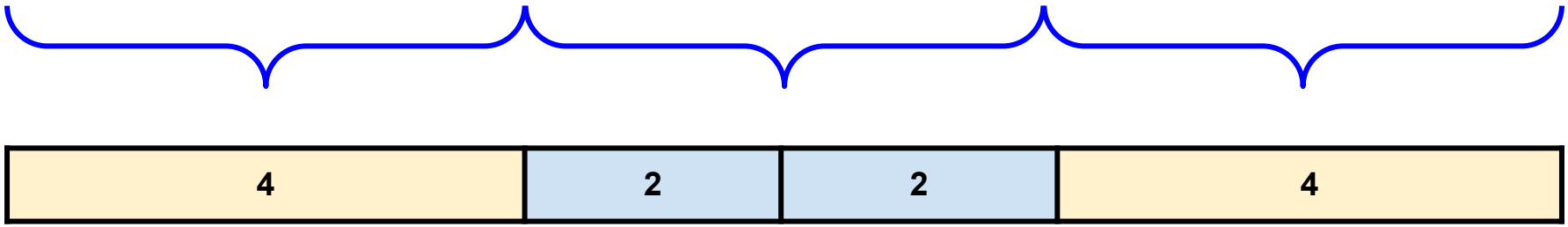
Column 1

Column 2

Nested Column 1

Nested Column 2

Column 3



Fluid Grid System

- The fluid grid system uses percentages instead of pixels for column widths.
- It has the same responsive capabilities as our fixed grid system, ensuring proper proportions for key screen resolutions and devices. You can make any row “fluid” by changing .row to .row-fluid.
- The column classes stay exactly the same, making it easy to flip between fixed and fluid grids.
- To offset, you operate in the same way as the fixed grid system—add .offset* to any column to shift by your desired number of columns.

```
<div class="row-fluid">
<div class="col-md-4 fluid-column">Fluid Column 1</div>
<div class="col-md-4 offset-md-2 fluid-column">Fluid Column 2 (Offset 2)</div>
</div>
```

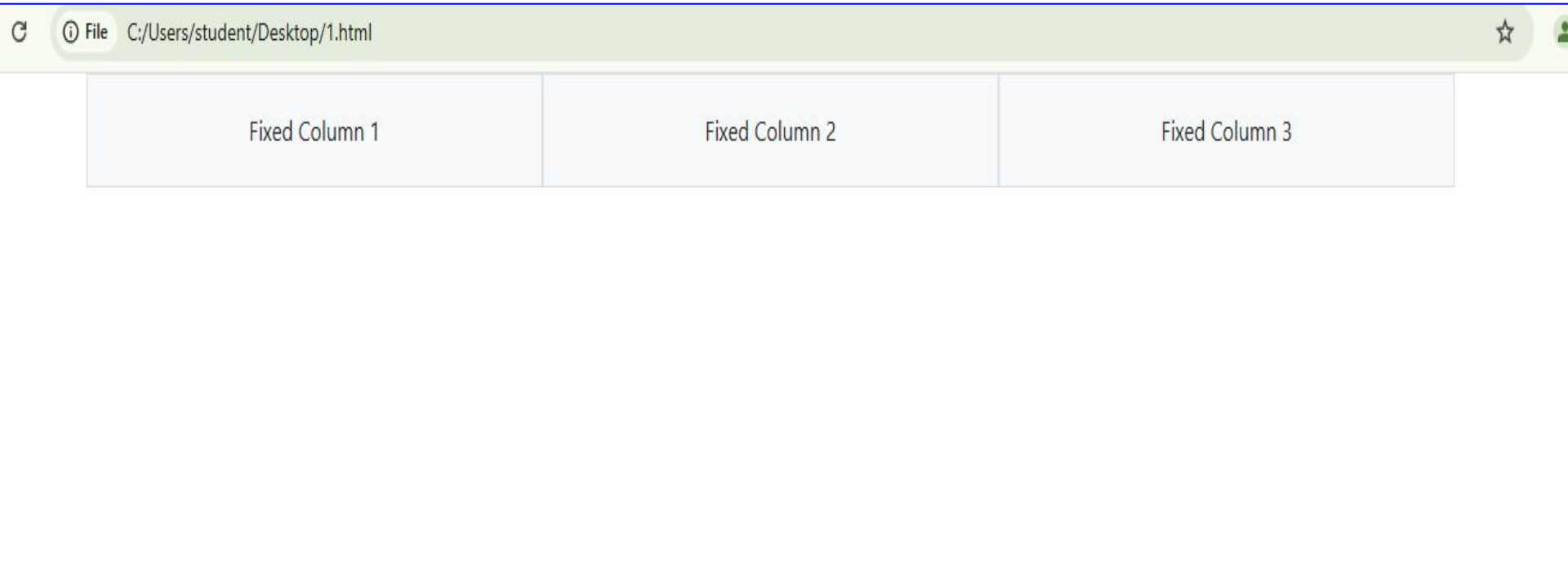
`row-fluid` is relevant for Bootstrap 3 and earlier. In Bootstrap 4 and later, you would use just **`row`** for a fluid layout.

Example

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Bootstrap Fluid Grid Example</title>
    <!-- Bootstrap CSS -->
    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
    <style>
    .fluid-column {
    background-color: #f8f9fa;
    border: 1px solid #dee2e6;
```

```
padding: 20px;
        text-align: center;
        }
        </style>
</head>
<body>
<div class="container">
        <!-- Using .row for fixed layout -->
        <div class="row">
        <div class="col-md-4 fluid-column">Fixed Column 1</div>
        <div class="col-md-4 fluid-column">Fixed Column 2</div>
        <div class="col-md-4 fluid-column">Fixed Column 3</div>
        </div>
```

```
</div>
</body>
</html>
```


File
C:/Users/student/Desktop/1.html
Fixed Column 1
Fixed Column 2
Fixed Column 3

Nesting a fluid grid

Nesting a fluid grid in Bootstrap requires an understanding that each nested .row resets the column count back to 12. When working with a fluid grid, this means you should use percentages for the nested columns to ensure they fit within the parent column. Here's a detailed explanation with examples.

Key Points about Nesting Fluid Grids:

1. **Column Reset**: Each nested .row resets the column count to 12. So, if you're inside a parent column that takes up a certain number of columns, you must ensure the nested columns add up to 12.
2. **Use of Percentages**: To fill the space effectively, use the appropriate column classes that sum to 12 based on the parent column's width.

Example of Nesting Fluid Columns:

Here's how to create a nested fluid grid:

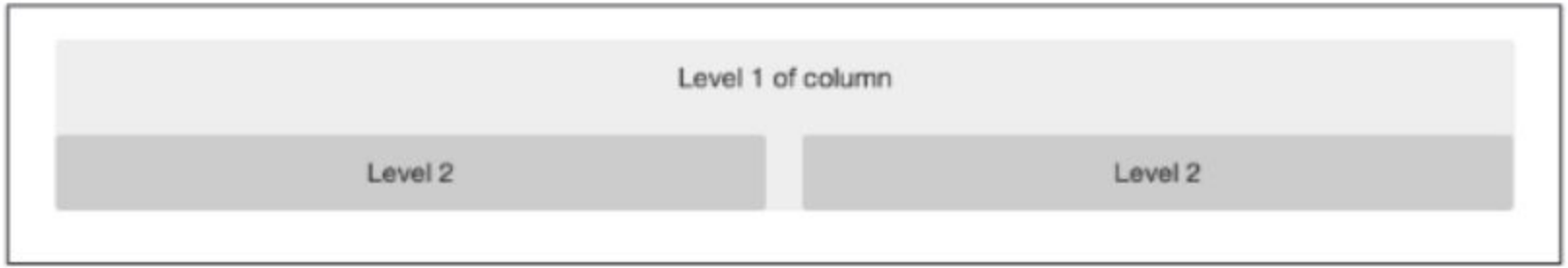


Figure 1-4. Nesting fluid grid

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Nesting Fluid Grid Example</title>
    <!-- Bootstrap CSS -->
    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
    <style>
    .nested-column {
    background-color: #e9ecef;
    border: 1px solid #ced4da;
    padding: 20px;
    text-align: center;
    }
```

```
        .parent-column {
        background-color: #f8f9fa;
        border: 1px solid #dee2e6;
        padding: 20px;
        text-align: center;
        }
        </style>
</head>
<body>

<div class="container">
        <div class="row">
```

```
<div class="col-md-6 parent-column">Parent Column (6)
        <div class="row">
                <div class="col-md-6 nested-column">Nested Column 1 (6)</div>
                <div class="col-md-6 nested-column">Nested Column 2 (6)</div>
        </div>
        </div>
        <div class="col-md-6 parent-column">Parent Column (6)</div>
        </div>
</div>
</body>
</html>
```


File
C:/Users/student/Desktop/1.html
Parent Column (6)
Nested Column 1 (6)
Nested Column 2 (6)
Parent Column (6)

Container Layouts

- **Fixed-Width Container**:
  - Use ** ** to create a fixed-width layout that is centered on the page.
  - The width of the container adjusts based on the viewport size (responsive).
- **Example**:

```
<div class="container">
    <h1>Welcome to My Site</h1>
    <p>This is a fixed-width, centered layout.</p>
</div>
```

Widths:

- In Bootstrap 4 and 5, the maximum widths for a fixed-width container are typically:
  - **Smaller devices (≥576px)**: 540px
  - **Medium devices (≥768px)**: 720px
  - **Large devices (≥992px)**: 960px
  - **Extra-large devices (≥1200px)**: 1140px
  - **Extra-extra-large devices (≥1400px)**: 1320px

Fluid Container:

- Use ** ** for a layout that takes up the full width of the viewport.
- This is particularly useful for applications and admin dashboards where you want content to stretch across the screen.

Example:

```
<div class="container-fluid">
    <h1>Dashboard</h1>
    <p>This is a fluid layout that fills the entire width of the viewport.</p>
</div>
```

Responsive Design

- **Viewport Meta Tag**:
  - To enable responsive features in Bootstrap, include the following <meta> tag in the **<head>** section of your HTML. This tag ensures proper scaling and responsiveness on mobile devices.
- **Example**:

<meta name="viewport" content="width=device-width, initial-scale=1.0">

Viewport- visible area of the web page that is displayed on the screen
Initial scale=1.0 - no zoom- actual size

Including Bootstrap CSS:

- To use Bootstrap's styles, you need to link to the Bootstrap CSS file. If you want the responsive features, also link to the responsive CSS file (in Bootstrap 3).

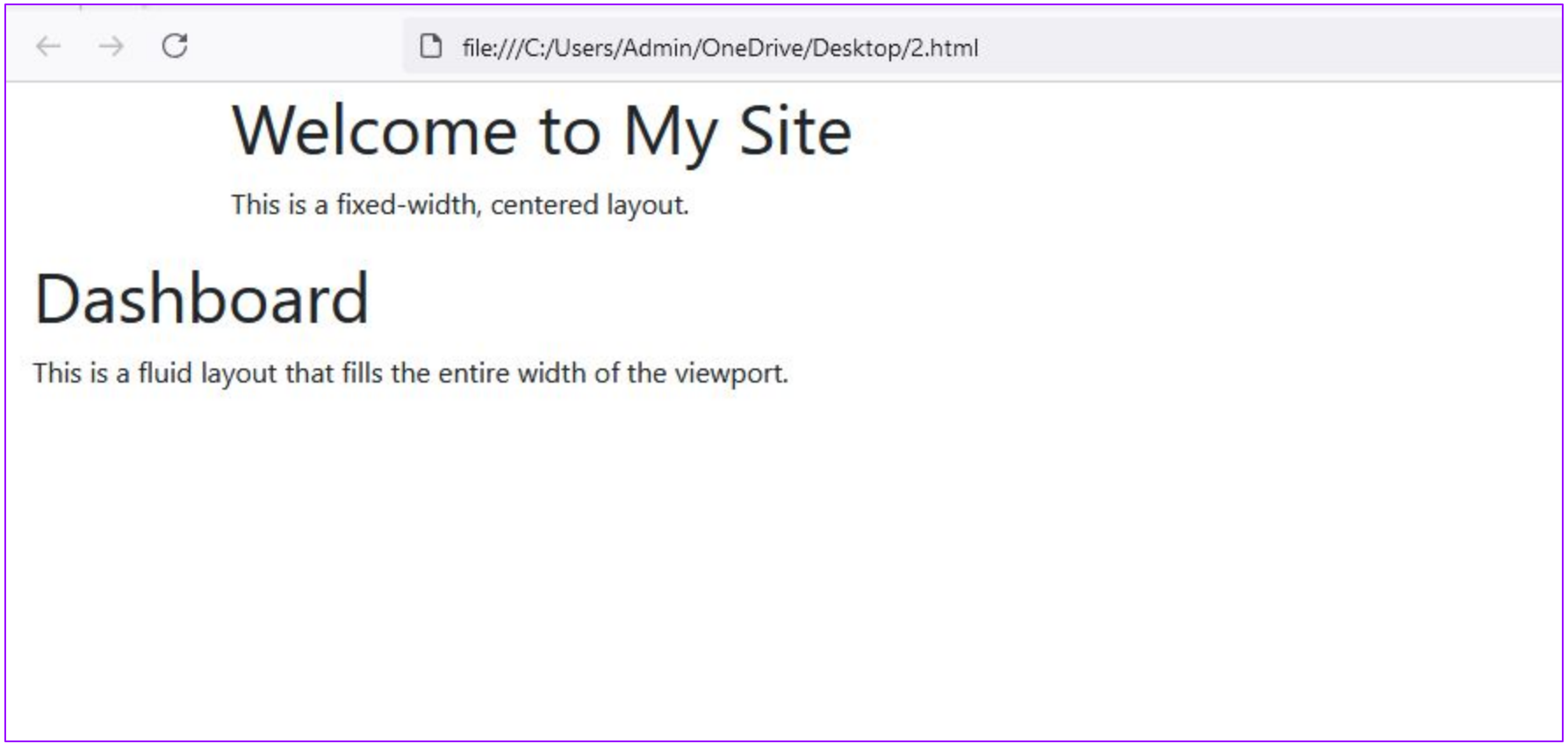
<link href="/css/bootstrap.min.css" rel="stylesheet">

or

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet">

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Bootstrap Grid Example</title>
    <!-- Bootstrap CSS -->
    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
</head>
<body>
    <div class="container">
    <h1>Welcome to My Site</h1>
```

```
        <p>This is a fixed-width, centered layout.</p>
        </div>
        <div class="container-fluid">
        <h1>Dashboard</h1>
        <p>This is a fluid layout that fills the entire width of the viewport.</p>
        </div>
</body>
</html>
```


file:///C:/Users/Admin/OneDrive/Desktop/2.html
Welcome to My Site
This is a fixed-width, centered layout.
Dashboard
This is a fluid layout that fills the entire width of the viewport.

Container: Use ** ** for a centered, fixed-width layout.

Fluid Container: Use ** ** for a full-width layout.

Responsive Design: Include the viewport meta tag and link to the Bootstrap CSS files to enable responsive features.

What is Responsive Design?

Responsive design is an approach to web design that ensures a website's content adjusts smoothly across different devices (desktops, tablets, and smartphones).

Different devices receive tailored layouts:

- **Desktop**: Wider layouts optimized for larger screens.
- **Tablets**: Layouts that adapt to portrait and landscape orientations.
- **Smartphones**: Layouts that accommodate narrower widths, often stacking content vertically.

How Bootstrap Achieves Responsive Design

CSS Media Queries:

- Bootstrap uses media queries to apply different styles based on the width of the browser viewport. This allows for fluid and flexible layouts.

Breakpoints:

- Bootstrap supports several breakpoints that define how the layout changes at various screen sizes. The main breakpoints are:
  - **Large desktop**: @media (min-width: 1200px)
  - **Tablets**: @media (min-width: 768px) and (max-width: 979px)
  - **Phones**:
    - Landscape phones: @media (max-width: 767px)
    - Portrait phones: @media (max-width: 480px)

Fluid Layouts:

- On smaller screens, columns stack vertically, each taking up 100% width.

Bootstrap 3

```
/* Large desktop */
@media (min-width: 1200px) {
        .column {
        width: 70px; /* Wider columns for large desktops */
        }
}
```

```
/* Portrait tablet to landscape and desktop */
@media (min-width: 768px) and (max-width: 979px) {
    .column {
    width: 42px; /* Columns for tablets */
    }
}
```

```
/* Landscape phone to portrait tablet */
@media (max-width: 767px) {
    .column {
    width: 100%; /* Full width for phones */
    }
}
```

```
/* Extra small devices */
@media (max-width: 480px) {
        .column {
        background-color: lightgray; /* Example of a style change */
        }
}
```

Organizing CSS for Responsive Design

1. **Single CSS File**:
   - You can include all rules in one CSS file using media queries, but this may make the file large.
2. **Separate CSS Files**:
   - For larger projects, you may want to separate styles into different CSS files based on breakpoints to optimize loading times.

```
<link rel="stylesheet" href="base.css">

<link rel="stylesheet" media="(min-width: 1200px)" href="large.css">

<link rel="stylesheet" media="(min-width: 768px) and (max-width: 979px)" href="tablet.css">

<link rel="stylesheet" media="(max-width: 767px)" href="phone.css">

<link rel="stylesheet" media="(max-width: 480px)" href="phone.css">
```

large.css

```
.column {
    width: 70%; /* Example width for large screens */
    background-color: lightblue;
}
```

Bootstrap 5:

In **Bootstrap 5**, the handling of responsive design is simplified, and you typically don’t need to create separate CSS files like **large.css or phone.css.** Instead, Bootstrap's built-in classes and utilities handle most of the responsive styling for you.

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet">