



Andhra Pradesh State Skill Development Corporation



Andhra Pradesh State Skill Development Corporation



Web designing using ReactJs

Bootstrap



Bootstrap

Bootstrap is freely available for everyone. The main features of bootstrap is, it is very simple and easy to use, huge JavaScript plugins are available, and easily design mobile friendly websites.

Features

- Easy to use
 - *Anybody with just basic knowledge of HTML and CSS can start using Bootstrap*
- Responsive features
 - *Bootstrap's responsive CSS adjusts to phones, tablets, and desktops*
- Mobile-Friendly
 - *Mobile-first approach: In Bootstrap 3, mobile-first styles are part of the core framework*
- Simple Integration
 - *Bootstrap can be simply integrated along with distinct other platforms and frameworks, on existing sites and new ones too and one more thing you can also utilize particular elements of Bootstrap along with your current CSS.*
- Pre-styled Components
 - *Bootstrap approaches with pre-styled components for alerts, dropdowns, nav bars, etc.*
- Customizable Bootstrap
 - *The Bootstrap can be customized as per the designs of your project.*
- Browser compatibility
 - *Bootstrap is compatible with all modern browsers (Chrome, Firefox, Internet Explorer, Safari, and Opera)*
- Great grid system
 - *Bootstrap is built on responsive 12-column grids, layouts and components. Whether you need a fixed grid or a responsive, it's only a matter of a few changes.*
- Extensive list of components
 - *Whether you need drop down menus, pagination or alert boxes, Bootstrap has got your covered. Some of the components pre styled are; Dropdowns, Button Groups, Navigation Bar, Breadcrumbs, Labels & Badges, Alerts, Progress Bar, And many others.*
- Base styling for most HTML elements
 - *A website has many different elements such as headings, lists, tables, buttons, forms, etc. The HTML elements for which styles are provided are; Typography Code, Tables, Forms, Buttons, Images, Icons*



- Good documentation
 - Not only does Bootstrap offer styling for almost every element a typical website or web application requires, it also provides a great documentation with examples and demo that only make it more easier for even someone new.

How to integrate bootstrap to our project

- We can download bootstrap from www.getbootstrap.com
- If you wanna run the bootstrap from online, we can use CDN and JSDelivr
- When we are going to use CDN and JSDelivr , we need to have internet connectivity whenever we are going to run our project.
- We can use either bootstrap files for offline or CDN and JSDelivr but we need to use the source files in the head section of the HTML document.

Color codes

Convey meaning through color with a handful of color utility classes. Includes support for styling links with hover states, too. We have various color codes in bootstrap. The following codes describe the color codes along with the colors.

- primary => Blue color
- secondary => Grey color (Light black color)
- info => Sky blue color
- danger => Red color
- success => Green color
- warning => Orange color
- dark => Black color
- white => White color

```
$theme-colors: (  
  "primary":    $primary,  
  "secondary":  $secondary,  
  "success":    $success,  
  "info":       $info,  
  "warning":    $warning,  
  "danger":     $danger,  
  "light":      $light,  
  "dark":       $dark  
) ;
```

Background color

- Easily set the background by using .bg-* color classes
- <h2 class="bg-primary"> Blue colored background for the text </h2>



```
<div class="bg-primary"></div>
<div class="bg-secondary"></div>
<div class="bg-success"></div>
<div class="bg-danger"></div>
<div class="bg-warning"></div>
<div class="bg-info"></div>
<div class="bg-light"></div>
<div class="bg-dark"></div>
```

Text color

- Colorize text with color utilities.

```
<h2 class="text-primary"> Blue colored text </h2>

<p class="text-primary">.text-primary</p>
<p class="text-secondary">.text-secondary</p>
<p class="text-success">.text-success</p>
<p class="text-danger">.text-danger</p>
<p class="text-warning bg-dark">.text-warning</p>
<p class="text-info bg-dark">.text-info</p>
<p class="text-light bg-dark">.text-light</p>
<p class="text-dark">.text-dark</p>
<p class="text-body">.text-body</p>
<p class="text-muted">.text-muted</p>
<p class="text-white bg-dark">.text-white</p>
<p class="text-black-50">.text-black-50</p>
<p class="text-white-50 bg-dark">.text-white-50</p>
```

Links

- You can use the .link-* classes to colorize links. Unlike the .text-* classes, these classes have a :hover and :focus state.

```
<a href="#" class="link-primary">Primary link</a>
<a href="#" class="link-secondary">Secondary link</a>
<a href="#" class="link-success">Success link</a>
<a href="#" class="link-danger">Danger link</a>
<a href="#" class="link-warning">Warning link</a>
<a href="#" class="link-info">Info link</a>
<a href="#" class="link-light">Light link</a>
<a href="#" class="link-dark">Dark link</a>
```

Components

- We also use theme colors as predefined colors for some components, for example, buttons.

```
<button type="button" class="btn btn-primary">Primary</button>
<button type="button" class="btn btn-secondary">Secondary</button>
<button type="button" class="btn btn-success">Success</button>
```



```
<button type="button" class="btn btn-danger">Danger</button>
<button type="button" class="btn btn-warning">Warning</button>
<button type="button" class="btn btn-info">Info</button>
<button type="button" class="btn btn-light">Light</button>
<button type="button" class="btn btn-dark">Dark</button>
```

Grid system

The Bootstrap Grid System is used for layout, specifically Responsive Layouts. ... The Grid is made up of groupings of Rows & Columns inside 1 or more Containers. The Bootstrap Grid can be used alone, without the Bootstrap JavaScript and other CSS Components. You just need to download and reference the "bootstrap-grid".

For implementing bootstrap grid system, we have to use following classes

- `row` (*For representing Rows*)
- `col` (*For representing Columns*)

Example

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

    <div class="col">
        row-1 col-2
    </div>

    <div class="col">
        row-1 col-3
    </div>
</div>
```

- Bootstrap grid system is calculated with 12 columns each and every column is having 1 grid of width.

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

We can change the width of each and every grid by mentioning the width. `col-2` is for representing column width with 2 grids.

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



```
    1 of 3
</div>
<div class="col-6">
    2 of 3 (wider)
</div>
<div class="col">
    3 of 3
</div>
</div>

<div class="row">
    <div class="col">
        1 of 3
    </div>
    <div class="col-5">
        2 of 3 (wider)
    </div>
    <div class="col">
        3 of 3
    </div>
</div>
```

Use col-{breakpoint}-auto classes to size columns based on the natural width of their content.

```
<div class="row justify-content-md-center">
    <div class="col col-lg-2">
        1 of 3
    </div>
    <div class="col-md-auto">
        Variable width content
    </div>
    <div class="col col-lg-2">
        3 of 3
    </div>
</div>
<div class="row">
    <div class="col">
        1 of 3
    </div>
    <div class="col-md-auto">
        Variable width content
    </div>
    <div class="col col-lg-2">
        3 of 3
    </div>
</div>
```



Responsive Grid system (*Break points*)

By using breakpoints, we can make the grid system as responsive, we can access the grid system in various screen resolutions. We have different classes for different screen resolutions.

- .col => Extra small scale device
- .col-sm => Small scale devices (>=576px)
- .col-md => Medium scale devices (>=768px)
- .col-lg => Large scale devices (>=992px)
- .col-xl => eXtra Large Scale devices (>=1200px)

Example

```
<!-- Row 1 -->
<div class="row justify-content-center">
    <div class="col-md-6 col-sm-12 col-lg-4 ">
        Row-1 Col-1
    </div>

    <div class="col-md-6 col-sm-12 col-lg-4 ">
        Row-1 Col-1
    </div>
</div>

<!-- Row 2 -->
<div class="row justify-content-center">
    <div class="col-md-6 col-sm-12 col-lg-4 ">
        Row-2 Col-1
    </div>

    <div class="col-md-6 col-sm-12 col-lg-4 ">
        Row-2 Col-1
    </div>
</div>
```

From the above example, we used two rows, each and every row is having two columns in it. We used various class names for providing responsiveness for every column. col-md is for medium scale devices (Smart phones), col-sm is for small-scale devices (Old moto e series mobiles), col-lg is for large-scale devices (Desktops)

Tables

We can style the HTML tables by using bootstrap predefined classes. The following are the available classes for styling tables.



- .table
- .table-dark
 - We can provide a dark background for table and table-row.

```
<tr class="table-dark">
    <th scope="row">Dark</th>
    <td>Cell</td>
    <td>Cell</td>
</tr>
```

- .table-bordered
 - Add **.table-bordered** for borders on all sides of the table and cells.
- .table-borderless
 - Add **.table-borderless** for generating borderless table.
- .thead-dark
- .thead-light
- .table-active
 - Highlight a table row or cell by adding a **.table-active** class.

```
<table class="table">
    <thead>
        <tr>
            <th scope="col">#</th>
            <th scope="col">First Name</th>
            <th scope="col">Last Name</th>
            <th scope="col">Role</th>
        </tr>
    </thead>

    <tbody>
        <tr class="table-active">
            <th scope="row">1</th>
            <td>Hanuman</td>
            <td>Kumar</td>
            <td>Multi-skill Trainer</td>
        </tr>
        <tr>
            <th scope="row">2</th>
            <td>Rajesh</td>
            <td>Nalleboina</td>
            <td>Multi-skill Trainer</td>
        </tr>
        <tr>
            <th scope="row">3</th>
            <td colspan="2" class="table-active">Swamy kankipati</td>
        </tr>
    </tbody>

```



- ```
<td>Multi-skill Trainer</td>
</tr>
</tbody>
</table>
```
- .table-striped
    - Use **.table-striped** to add zebra-striping to any table row within the **<tbody>**.

```
<table class="table table-striped"></table>
```

  - .table-hover
    - We can observe the background color of each and every row when we are going to hover it.

```
<table class="table table-striped table-hover"></table>
```

  - Add .table-sm to make any .table more compact by cutting all cell padding in half.

```
<table class="table table-sm"></table>
```

  - Border styles, active styles, and table variants are not inherited by nested tables.
    - To prevent any styles from leaking to nested tables, we use the child combinator (>) selector in our CSS. Since we need to target all the tds and ths in the thead, tbody, and tfoot, our selector would look pretty long without it. As such, we use the rather odd looking **.table > :not(caption) > \* > \*** selector to target all tds and ths of the .table, but none of any potential nested tables.
    - Note that if you add s as direct children of a table, those will be wrapped in a by default, thus making our selectors work as intended.\_

```
<table class="table table-striped">
 <thead></thead>
 <tbody>
 <tr>
 <td colspan="4">
 <table class="table mb-0"></table>
 </td>
 </tr>
 </tbody>
</table>
```

## Attributes for table

- colspan
- rowspan
- scope="col|group|colgroup|rowgroup" ##### Table-responsive-{breakpoint}



We can make the tables as responsive by applying the following class names for the parent element of the tables. We have to create a parent element for the table.\*

```
.table-responsive * .table-responsive-sm * .table-responsive-md *
.table-responsive-lg
```

```
<div class="table-responsive">
 <table class="table">
 <thead>
 <tr>
 <th scope="col">#</th>
 <th scope="col">First Name</th>
 <th scope="col">Last Name</th>
 <th scope="col">Role</th>
 </tr>
 </thead>

 <tbody>
 <tr class="table-active">
 <th scope="row">1</th>
 <td>Hanuman</td>
 <td>Kumar</td>
 <td>Multi-skill Trainer</td>
 </tr>
 <tr>
 <th scope="row">2</th>
 <td>Rajesh</td>
 <td>Nalleboina</td>
 <td>Multi-skill Trainer</td>
 </tr>
 <tr>
 <th scope="row">3</th>
 <td colspan="2" class="table-active">Swamy
kankipati</td>
 <td>Multi-skill Trainer</td>
 </tr>
 </tbody>
 </table>
</div>
```

Example:

```
<!DOCTYPE html>
<html>
 <head>
 <title>:: Bootstrap Tables ::</title>
 <link rel="stylesheet" type="text/css"
href="css/bootstrap.min.css">
 <script type="text/javascript"
src="js/bootstrap.min.js"></script>
```



```
<meta name="viewport"
content="width=device-width,initial-scale=1">
</head>
<body>
<div class="container mt-5">
<div class="table-responsive-md">
<table class="table table-bordered">
<thead class="thead-dark">
<tr class="bg-primary">
<th scope="col">S.No</th>
<th>Continent</th>
<th>First_Name</th>
<th>Last_Name</th>
<th>Age</th>
<th colspan="2">Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td scope="row">1</td>
<td rowspan="2">Non Asia</td>
<td>Warren</td>
<td>Buffet</td>
<td>69</td>
<td>Edit</td>
<td>Delete</td>
</tr>
<tr>
<td>2</td>
<td>Jeff</td>
<td>Bezzos</td>
<td>65</td>
<td>Edit</td>
<td>Delete</td>
</tr>
<tr>
<td>3</td>
<td>Asia</td>
<td>Mukesh</td>
<td>Ambani</td>
<td>60</td>
<td>Edit</td>
<td>Delete</td>
</tr>
</tbody>
</table>
</div>
</div>
```



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

## Cards

---

A card is a flexible and extensible content container. It includes options for headers and footers, a wide variety of content, contextual background colors, and powerful display options. For making a layout or division as a card, we have to use following class names

- .card
- .card-header
- .card-body
- .card-footer
- .card-img-top
- .card-title
- .card-subtitle
- .card-text
- .card-link
- .card-group
- .card-deck

Basically a card container divided into three parts: card-header, card-body and card-footer.

```
<div class="card">
 <div class="card-header"> Header </div>
 <div class="card-body"> Body </div>
 <div class="card-footer"> </div>
</div>
```

We can apply styles to the content inside the card-body by using the classes card-title, card-subtitle, card-text, card-img, card-link and card-group.

```
<div class="card-body">
 <h1 class="card-title"> Title text </h1>
 <p class="card-text"> Card content </p>
</div>
```

Example:

```
<!DOCTYPE html>
<html>
 <head>
 <title>:: Profile card ::</title>
 <link rel="stylesheet" type="text/css"
href="css/bootstrap.min.css">
```



```
<script type="text/javascript"
src="js/bootstrap.min.js"></script>
<meta name="viewport"
content="width=device-width,initial-scale=1">
</head>
<body>
<div class="container mt-5">
<div class="card-deck">
<div class="card" style="width:18rem; border: 1px solid lightgray;">

<div class="card-body">
<h1 class="card-title">Kalyan Chakravarthi</h1>
<h3 class="card-subtitle">Software Developer</h3>
<p class="card-text">Working in APSSDC</p>
<button class="btn btn-primary">Click Me</button>
</div>
</div>

<div class="card" style="width:20rem; border: 1px solid lightgray;">

<div class="card-body">
<h1 class="card-title">Kalyan Chakravarthi</h1>
<h3 class="card-subtitle">Software Developer</h3>
<p class="card-text">Working in APSSDC</p>
<button class="btn btn-primary">Click Me</button>
</div>
</div>

<div class="card" style="width:20rem; border: 1px solid lightgray;">

<div class="card-body">
<h1 class="card-title">Kalyan Chakravarthi</h1>
<h3 class="card-subtitle">Software Developer</h3>
<p class="card-text">Working in APSSDC</p>
<button class="btn btn-primary">Click Me</button>
</div>
</div>

<div class="card" style="width:20rem; border: 1px solid lightgray;">

<div class="card-body">
<h1 class="card-title">Kalyan Chakravarthi</h1>
<h3 class="card-subtitle">Software Developer</h3>
<p class="card-text">Working in APSSDC</p>
<button class="btn btn-primary">Click Me</button>
</div>
</div>
```



```
<p class="card-text">Working in APSSDC</p>
<button class="btn btn-primary"><a href="table.html"
class="text-white">Click Me</button>
</div>
</div>
</body>
</html>
```

## Bootstrap forms

---

- Textual form controls like <input>s, <select>s, and <textarea>s are styled with the .form-control class. Included are styles for general appearance, focus state, sizing, and more.

```
<form>
 <div class="form-group">
 <label for="exampleFormControlInput1">Email address</label>
 <input type="email" class="form-control"
id="exampleFormControlInput1" placeholder="name@example.com">
 </div>
 <div class="form-group">
 <label for="exampleFormControlSelect1">Example select</label>
 <select class="form-control" id="exampleFormControlSelect1">
 <option>1</option>
 <option>2</option>
 <option>3</option>
 <option>4</option>
 <option>5</option>
 </select>
 </div>
 <div class="form-group">
 <label for="exampleFormControlSelect2">Example multiple
select</label>
 <select multiple class="form-control"
id="exampleFormControlSelect2">
 <option>1</option>
 <option>2</option>
 <option>3</option>
 <option>4</option>
 <option>5</option>
 </select>
 </div>
 <div class="form-group">
 <label for="exampleFormControlTextarea1">Example
textarea</label>
 <textarea class="form-control"
```



```
id="exampleFormControlTextareal" rows="3"></textarea>
 </div>
</form>
```

## File-input

```
<form>
 <div class="form-group">
 <label for="exampleFormControlFile1">Example file input</label>
 <input type="file" class="form-control-file"
id="exampleFormControlFile1">
 </div>
</form>
```

- Set heights using classes like `.form-control-lg` and `.form-control-sm`.
- Add the `readonly` boolean attribute on an input to prevent modification of the input's value. Read-only inputs appear lighter (just like disabled inputs), but retain the standard cursor.

```
<input class="form-control" type="text" placeholder=" Readonly input
here..." readonly>
```

## Range

- Set horizontally scrollable range inputs using `.form-control-range`.

```
<form>
 <div class="form-group">
 <label for="formControlRange">Example Range input</label>
 <input type="range" class="form-control-range"
id="formControlRange">
 </div>
</form>
```

## Checkboxes & Radios

- Default checkboxes and radios are improved upon with the help of `.form-check`, a single class for both input types that improves the layout and behavior of their HTML elements. \* Checkboxes are for selecting one or several options in a list, while radios are for selecting one option from many.
- Disabled checkboxes and radios are supported. The `disabled` attribute will apply a lighter color to help indicate the input's state.
- Checkboxes and radios use are built to support HTML-based form validation and provide concise, accessible labels. As such, our `label`s and `input`s are sibling elements as opposed to `label` within `input`. This is slightly more verbose as you must specify `id` and `for` attributes to relate the `label` and `input`.

```
<div class="form-check">
 <input class="form-check-input" type="checkbox" value=""
id="defaultCheck1">
```



```
<label class="form-check-label" for="defaultCheck1">
 Default checkbox
</label>
</div>
<div class="form-check">
 <input class="form-check-input" type="checkbox" value="" id="defaultCheck2" disabled>
 <label class="form-check-label" for="defaultCheck2">
 Disabled checkbox
 </label>
</div>
```

## Alerts

---

- Alerts are available for any length of text, as well as an optional dismiss button. For proper styling, use one of the eight required contextual classes ( .alert-success ).

```
<div class="alert alert-secondary" role="alert">
 This is a secondary alert—check it out!
</div>
<div class="alert alert-success" role="alert">
 This is a success alert—check it out!
</div>
```

## Alert-links

- Use the .alert-link utility class to quickly provide matching colored links within any alert.

```
<div class="alert alert-danger" role="alert">
 This is a danger alert with an example link. Give it a click if you like.
</div>
<div class="alert alert-warning" role="alert">
 This is a warning alert with an example link. Give it a click if you like.
</div>
```

## Alert-content

```
<div class="alert alert-success" role="alert">
 <h4 class="alert-heading">Well done!</h4>
 <p>Aww yeah, you successfully read this important alert message. This example text is going to run a bit longer so that you can see how spacing within an alert works with this kind of content.</p>
 <hr>
 <p class="mb-0">Whenever you need to, be sure to use margin utilities
```



```
to keep things nice and tidy.</p>
</div>
```

## Dismissible alerts

Using the alert JavaScript plugin, it's possible to dismiss any alert inline. Here's how:

- Be sure you've loaded the alert plugin, or the compiled Bootstrap JavaScript.
- If you're building our JavaScript from source, it requires util.js. The compiled version includes this.
- Add a dismiss button and the .alert-dismissible class, which adds extra padding to the right of the alert and positions the .close button.
- On the dismiss button, add the data-dismiss="alert" attribute, which triggers the JavaScript functionality. Be sure to use the element with it for proper behavior across all devices.
- To animate alerts when dismissing them, be sure to add the .fade and .show classes.

```
<div class="alert alert-warning alert-dismissible fade show"
role="alert">
 Holy guacamole! You should check in on some of those
fields below.
 <button type="button" class="close" data-dismiss="alert"
aria-label="Close">
 ×
 </button>
</div>
```

By using the above code snippet, we can provide a closing button for dismissing the alert. But we have to include bootstrap.min.js in the head section of the HTML document.