HAZZA INSTITUTE OF TECHNOLOGY

iqracity

HAZZA INSTITUTE OF TECHNOLOGY

MERN Stack (ES6 + REACT) Course

iqracity

Lab 15

Total Time:
3 hours

Pre-Lab Activities:

- No Pre-Lab Activity

Learning Outcomes:

- Perform the execution, debugging, testing, and profiling of web apps in modern IDEs.

Lab Tasks:

- Getting a text Editor
- Installing React in the local machine
- Accordion
  - Accordion Item #1
  - Accordion Item #2
  - Accordion Item #3

Student Activities:

- Explore text Editor
- Explore Installing React in the local machine
- Explore Accordion
  - Accordion Item #1
  - Accordion Item #2
  - Accordion Item #3

Lab Solution

Reactstrap is a React component library for Bootstrap

Getting Started

Install Reactstrap:

```
npm install reactstrap react react-dom
```

Reactstrap currently requires React 16.8 or higher.

Include Bootstrap

Either:

Import Bootstrap in your application code:

```
npm install --save bootstrap
import 'bootstrap/dist/css/bootstrap.min.css';
```

or include Bootstrap from a CDN URL in your HTML layout:

```
<head>
<link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.0/dist/css/bootstrap.min.css" />
</head>
```

Import components:

```
import React from 'react';
import { Button } from 'reactstrap';

export default (props) => {
  return (
<Button color="danger">Danger!</Button>
  );
};
```

About

Unlike some component libraries, Reactstrap does not embed its own styles, and instead depends on the Bootstrap CSS framework for its styles and theme. This allows you to have consistent styles across your React-based components and static parts of your site, and allows you to include your own custom Bootstrap theme when needed.

Unlike using Bootstrap in HTML, Reactstrap exports all the correct Bootstrap classes automatically, and don't need to use or include Bootstrap's JavaScript files or add data attributes to trigger functionality. Instead, components are defined in React-friendly components with appropriate props for you to control.

So instead of:

```
<!-- HTML -->
```

```
<div class="modal" tabindex="-1">
<div class="modal-dialog">
<div class="modal-content">
<div class="modal-header">
<h5 class="modal-title">Modal title</h5>
<button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>
</div>
<div class="modal-body">
<p>Modal body text goes here.</p>
</div>
</div>
</div>
</div>
```

You can use:

```
// React
import { Modal, ModalBody, ModalHeader } from 'reactstrap';
...
<Modal isOpen={open} toggle={() => setOpen(false)}>
<ModalHeader>
   Modal title
</ModalHeader>
<ModalBody>
   Modal body text goes here.
</ModalBody>
</Modal>
```

Accordion

Build vertically collapsing accordions in combination with our Collapse JavaScript plugin.
Limited time offer: Get 10 free Adobe Stock images.
ads via Carbon

How it works

The accordion uses collapse internally to make it collapsible. To render an accordion that's expanded, add the .open class on the .accordion.
The animation effect of this component is dependent on the prefers-reduced-motion media query. See the reduced motion section of our accessibility documentation.
Example
Click the accordions below to expand/collapse the accordion content.
Accordion Item #1
This is the first item's accordion body. It is shown by default, until the collapse plugin adds the appropriate classes that we use to style each element. These classes control the overall appearance, as well as the showing and hiding via CSS transitions. You can modify any of this with custom CSS or overriding our default variables. It's also worth noting that just about any HTML can go within the .accordion-body, though the transition does limit overflow.

Accordion Item #1
</button>
</h2>
<div id="collapseOne" class="accordion-collapse collapse show" aria-labelledby="headingOne" data-bs-parent="#accordionExample">
<div class="accordion-body">
<strong>This is the first item's accordion body.</strong> It is shown by default, until the collapse plugin adds the appropriate classes that we use to style each element. These classes control the overall appearance, as well as the showing and hiding via CSS transitions. You can modify any of this with custom CSS or overriding our default variables. It's also worth noting that just about any HTML can go within the <code>.accordion-body</code>, though the transition does limit overflow.
</div>
</div>
</div>
<div class="accordion-item">
<h2 class="accordion-header" id="headingTwo">
<button class="accordion-button collapsed" type="button" data-bs-toggle="collapse" data-bs-target="#collapseTwo" aria-expanded="false" aria-controls="collapseTwo">

Accordion Item #2
</button>
</h2>
<div id="collapseTwo" class="accordion-collapse collapse" aria-labelledby="headingTwo" data-bs-parent="#accordionExample">
<div class="accordion-body">
<strong>This is the second item's accordion body.</strong> It is hidden by default, until the collapse plugin adds the appropriate classes that we use to style each element. These classes control the overall appearance, as well as the showing and hiding via CSS transitions. You can modify any of this with custom CSS or overriding our default variables. It's also worth noting that just about any HTML can go within the <code>.accordion-body</code>, though the transition does limit overflow.
</div>
</div>
</div>
<div class="accordion-item">
<h2 class="accordion-header" id="headingThree">
<button class="accordion-button collapsed" type="button" data-bs-toggle="collapse" data-bs-target="#collapseThree" aria-expanded="false" aria-controls="collapseThree">

Accordion Item #3
</button>
</h2>
<div id="collapseThree" class="accordion-collapse collapse" aria-labelledby="headingThree" data-bs-parent="#accordionExample">
<div class="accordion-body">
<strong>This is the third item's accordion body.</strong> It is hidden by default, until the collapse plugin adds the appropriate classes that we use to style each element. These classes control the overall

appearance, as well as the showing and hiding via CSS transitions. You can modify any of this with custom CSS or overriding our default variables. It's also worth noting that just about any HTML can go within the <code>.accordion-body</code>, though the transition does limit overflow.
</div>
</div>
</div>
</div>

Flush

Add .accordion-flush to remove the default background-color, some borders, and some rounded corners to render accordions edge-to-edge with their parent container.

Accordion Item #1

Accordion Item #2

Accordion Item #3

<div class="accordion accordion-flush" id="accordionFlushExample">
<div class="accordion-item">
<h2 class="accordion-header" id="flush-headingOne">
<button class="accordion-button collapsed" type="button" data-bs-toggle="collapse" data-bs-target="#flush-collapseOne" aria-expanded="false" aria-controls="flush-collapseOne">

Accordion Item #1

</button>
</h2>
<div id="flush-collapseOne" class="accordion-collapse collapse" aria-labelledby="flush-headingOne" data-bs-parent="#accordionFlushExample">
<div class="accordion-body">Placeholder content for this accordion, which is intended to demonstrate the <code>.accordion-flush</code> class. This is the first item's accordion body.</div>
</div>
</div>
<div class="accordion-item">
<h2 class="accordion-header" id="flush-headingTwo">
<button class="accordion-button collapsed" type="button" data-bs-toggle="collapse" data-bs-target="#flush-collapseTwo" aria-expanded="false" aria-controls="flush-collapseTwo">

Accordion Item #2

</button>
</h2>
<div id="flush-collapseTwo" class="accordion-collapse collapse" aria-labelledby="flush-headingTwo" data-bs-parent="#accordionFlushExample">
<div class="accordion-body">Placeholder content for this accordion, which is intended to demonstrate the <code>.accordion-flush</code> class. This is the second item's accordion body. Let's imagine this being filled with some actual content.</div>
</div>
</div>
<div class="accordion-item">
<h2 class="accordion-header" id="flush-headingThree">

<button class="accordion-button collapsed" type="button" data-bs-toggle="collapse" data-bs-target="#flush-collapseThree" aria-expanded="false" aria-controls="flush-collapseThree">

Accordion Item #3

</button>
</h2>
<div id="flush-collapseThree" class="accordion-collapse collapse" aria-labelledby="flush-headingThree" data-bs-parent="#accordionFlushExample">
<div class="accordion-body">Placeholder content for this accordion, which is intended to demonstrate the <code>.accordion-flush</code> class. This is the third item's accordion body. Nothing more exciting happening here in terms of content, but just filling up the space to make it look, at least at first glance, a bit more representative of how this would look in a real-world application.</div>
</div>
</div>
</div>

Always open

Omit the data-bs-parent attribute on each .accordion-collapse to make accordion items stay open when another item is opened.
Accordion Item #1
This is the first item's accordion body. It is shown by default, until the collapse plugin adds the appropriate classes that we use to style each element. These classes control the overall appearance, as well as the showing and hiding via CSS transitions. You can modify any of this with custom CSS or overriding our default variables. It's also worth noting that just about any HTML can go within the .accordion-body, though the transition does limit overflow.
Accordion Item #2
Accordion Item #3
<div class="accordion" id="accordionPanelsStayOpenExample">
<div class="accordion-item">
<h2 class="accordion-header" id="panelsStayOpen-headingOne">
<button class="accordion-button" type="button" data-bs-toggle="collapse" data-bs-target="#panelsStayOpen-collapseOne" aria-expanded="true" aria-controls="panelsStayOpen-collapseOne">

Accordion Item #1

</button>
</h2>
<div id="panelsStayOpen-collapseOne" class="accordion-collapse collapse show" aria-labelledby="panelsStayOpen-headingOne">
<div class="accordion-body">
<strong>This is the first item's accordion body.</strong> It is shown by default, until the collapse plugin adds the appropriate classes that we use to style each element. These classes control the overall appearance, as well as the showing and hiding via CSS transitions. You can modify any of this with

custom CSS or overriding our default variables. It's also worth noting that just about any HTML can go within the <code>.accordion-body</code>, though the transition does limit overflow.
</div>
</div>
</div>
<div class="accordion-item">
<h2 class="accordion-header" id="panelsStayOpen-headingTwo">
<button class="accordion-button collapsed" type="button" data-bs-toggle="collapse" data-bs-target="#panelsStayOpen-collapseTwo" aria-expanded="false" aria-controls="panelsStayOpen-collapseTwo">

Accordion Item #2

</button>
</h2>
<div id="panelsStayOpen-collapseTwo" class="accordion-collapse collapse" aria-labelledby="panelsStayOpen-headingTwo">
<div class="accordion-body">
<strong>This is the second item's accordion body.</strong> It is hidden by default, until the collapse plugin adds the appropriate classes that we use to style each element. These classes control the overall appearance, as well as the showing and hiding via CSS transitions. You can modify any of this with custom CSS or overriding our default variables. It's also worth noting that just about any HTML can go within the <code>.accordion-body</code>, though the transition does limit overflow.
</div>
</div>
</div>
<div class="accordion-item">
<h2 class="accordion-header" id="panelsStayOpen-headingThree">
<button class="accordion-button collapsed" type="button" data-bs-toggle="collapse" data-bs-target="#panelsStayOpen-collapseThree" aria-expanded="false" aria-controls="panelsStayOpen-collapseThree">

Accordion Item #3

</button>
</h2>
<div id="panelsStayOpen-collapseThree" class="accordion-collapse collapse" aria-labelledby="panelsStayOpen-headingThree">
<div class="accordion-body">
<strong>This is the third item's accordion body.</strong> It is hidden by default, until the collapse plugin adds the appropriate classes that we use to style each element. These classes control the overall appearance, as well as the showing and hiding via CSS transitions. You can modify any of this with custom CSS or overriding our default variables. It's also worth noting that just about any HTML can go within the <code>.accordion-body</code>, though the transition does limit overflow.
</div>
</div>
</div>
</div>

Accessibility
Please read the collapse accessibility section for more information.
Sass
Variables
Copy

```
$accordion-padding-y:                     1rem;
$accordion-padding-x:                     1.25rem;
$accordion-color:                         $body-color;
$accordion-bg:                            $body-bg;
$accordion-border-width:                  $border-width;
$accordion-border-color:                  rgba($black, .125);
$accordion-border-radius:                 $border-radius;
$accordion-inner-border-radius:           subtract($accordion-border-radius, $accordion-border-width);

$accordion-body-padding-y:                $accordion-padding-y;
$accordion-body-padding-x:                $accordion-padding-x;

$accordion-button-padding-y:              $accordion-padding-y;
$accordion-button-padding-x:              $accordion-padding-x;
$accordion-button-color:                  $accordion-color;
$accordion-button-bg:                     $accordion-bg;
$accordion-transition:                    $btn-transition, border-radius .15s ease;
$accordion-button-active-bg:              tint-color($component-active-bg, 90%);
$accordion-button-active-color:           shade-color($primary, 10%);

$accordion-button-focus-border-color:     $input-focus-border-color;
$accordion-button-focus-box-shadow:       $btn-focus-box-shadow;

$accordion-icon-width:                    1.25rem;
$accordion-icon-color:                    $accordion-button-color;
$accordion-icon-active-color:             $accordion-button-active-color;
$accordion-icon-transition:               transform .2s ease-in-out;
$accordion-icon-transform:                rotate(-180deg);

$accordion-button-icon:
url("data:image/svg+xml,<svg xmlns='http://www.w3.org/2000/svg' viewBox='0 0 16 16'
fill='#{$accordion-icon-color}'><path fill-rule='evenodd' d='M1.646 4.646a.5.5 0 0 1 .708 0L8
10.293l5.646-5.647a.5.5 0 0 1 .708.708l-6 6a.5.5 0 0 1-.708 0l-6-6a.5.5 0 0 1 0-.708z'/></svg>");
$accordion-button-active-icon:  url("data:image/svg+xml,<svg xmlns='http://
```

Reference https://getbootstrap.com/docs/5.1/components/accordion/