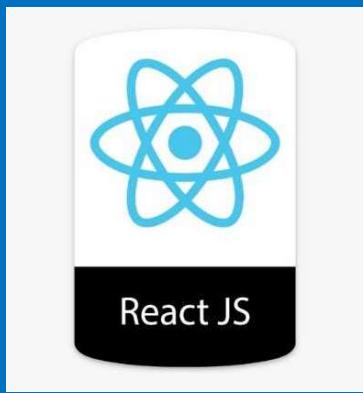


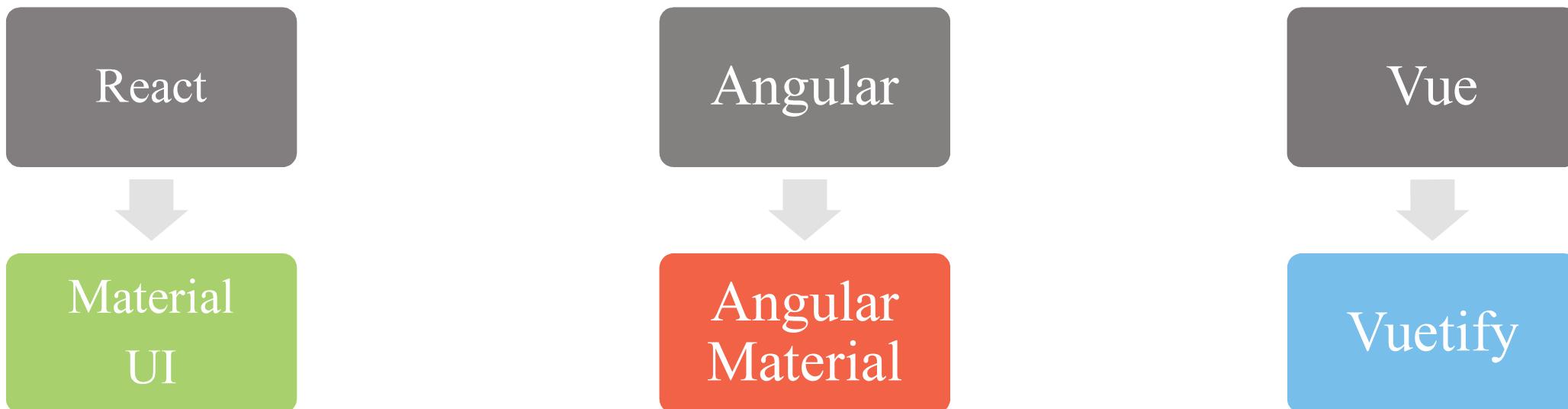
1 Creating Beautiful Apps with Material UI



Material UI Introduction

What is Material UI ?

- UI Component Library
- Provides us with components to build awesome user interfaces in quick time
- Implementation of Google's Material Design Specification
- Using These We Can Build Clean and Elegant UI



Introduction

- Material UI is an open-source React component library that implements Google's [Material Design](#).
- It includes a comprehensive collection of prebuilt components that are ready for use in production right out of the box.
- Material UI is beautiful by design and features a suite of customization options that make it easy to implement your own custom design system on top of our components.

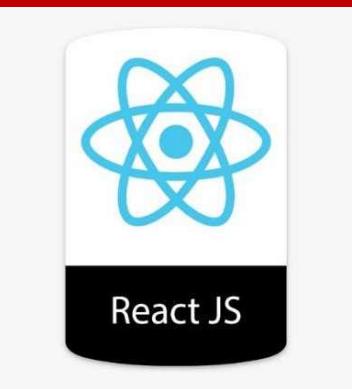
Advantages of Material UI

- **Ship faster:** Over 2,500 open-source [contributors](#) have poured countless hours into these components. Focus on your core business logic instead of reinventing the wheel—we've got your UI covered.
- **Beautiful by default:** we're meticulous about our implementation of [Material Design](#), ensuring that every Material UI component meets the highest standards of form and function, but diverge from the official spec where necessary to provide multiple great options.
- **Customizability:** the library includes an extensive set of intuitive customizability features. [The templates](#) in our store demonstrate how far you can go with customization.

Material UI vs. Base UI

- **Material UI** and Base UI feature many of the same UI components, but Base UI comes without any default styles or styling solutions.
- Material UI is *comprehensive* in that it comes packaged with default styles, and is optimized to work with Emotion (or styled-components).
- Base UI, by contrast, could be considered the "skeletal" or "headless" counterpart to Material UI—in fact, future versions of Material UI will use Base UI components and hooks for its foundational structure.

- **Cross-team collaboration:** Material UI's intuitive developer experience reduces the barrier to entry for back-end developers and less technical designers, empowering teams to collaborate more effectively. The [design kits](#) streamline your workflow and boost consistency between designers and developers.
- **Trusted by thousands of organizations:** Material UI has the largest UI community in the React ecosystem. It's almost as old as React itself—its history stretches back to 2014—and we're in this for the long haul. You can count on the community's support for years to come (e.g. [Stack Overflow](#)).



Material UI Installation

Default installation

Run one of the following commands to add Material UI to your project:

npm

```
npm install @mui/material @emotion/react @emotion/styled
```

yarn

```
yarn add @mui/material @emotion/react @emotion/styled
```

With Styled - Components

With styled-components

Material UI uses [Emotion](#) as its default styling engine. If you want to use [styled-components](#) instead, run one of the following commands:

npm

```
npm install @mui/material @mui/styled-engine-sc styled-components
```

yarn

```
yarn add @mui/material @mui/styled-engine-sc styled-components
```

Visit the [Styled engine guide](#) for more information about how to configure styled-components.

Peer dependencies

Please note that `react` and `react-dom` are peer dependencies too:

```
"peerDependencies": {  
  "react": "^17.0.0 || ^18.0.0",  
  "react-dom": "^17.0.0 || ^18.0.0"  
},
```

Robo Fonts

The screenshot shows the Material UI documentation for the Roboto font. The top navigation bar includes the MUI logo, 'MUI CORE', 'Material UI v5.13.6', a search bar, and various icons. A banner at the top right announces 'MUI X v6 is out! Discover what's new and get started now!' with a 'Ctrl+K' keyboard shortcut. The main content area features a section titled 'Roboto font' with a note that Material UI uses it by default. It provides installation instructions for npm and yarn, and shows code snippets for both. Below this, a note explains that Fontsource can be configured to load specific subsets, weights, and styles. The sidebar on the left lists 'Getting started' (Overview, Installation, Usage, Example projects, Templates, Learn, Design resources, FAQs, Supported components, Supported platforms, Support), 'Components' (Component API), and 'Diamond Sponsors' (Octopus Deploy, doT, zesty). The right sidebar contains a message about MUI's solidarity with Ukraine, a 'CONTENTS' section listing various documentation pages, and a 'Robo font' section with its own sub-links for npm, yarn, Google Web Fonts, and CDN.

MUI CORE
Material UI v5.13.6

Diamond Sponsors

- Octopus Deploy
- doT
- zesty

Getting started

- Overview
- Installation**
- Usage
- Example projects
- Templates
- Learn
- Design resources
- FAQs
- Supported components
- Supported platforms
- Support

Components

Component API

Roboto font

Material UI is designed to use the [Roboto](#) font by default. You may add it to your project with npm or yarn via [Fontsource](#), or with the Google Fonts CDN.

npm

```
npm install @fontsource/roboto
```

yarn

```
yarn add @fontsource/roboto
```

Then you can import it in your entry point like this:

```
import '@fontsource/roboto/300.css';
import '@fontsource/roboto/400.css';
import '@fontsource/roboto/500.css';
import '@fontsource/roboto/700.css';
```

Fontsource can be configured to load specific subsets, weights and styles. Material UI's default typography configuration relies only on the 300, 400, 500, and 700 font weights.

Google Web Fonts

To install the Roboto font in your project using the Google Web Fonts CDN, add the following

MUI X v6 is out! Discover what's new and get started now!

Search... Ctrl+K

MUI stands in solidarity with the Ukrainian people against the Russian invasion.

Find out how you can help.

CONTENTS

- Default installation
- npm
- yarn
- With styled-components
- npm
- yarn
- Peer dependencies
- Robo font**
- npm
- yarn
- Google Web Fonts
- Icons
- npm
- yarn
- Google Web Fonts
- CDN

Icons

To use the [font Icon component](#) or the prebuilt SVG Material Icons (such as those found in the [icon demos](#)), you must first install the [Material Icons](#) font. You can do so with npm or yarn, or with the Google Web Fonts CDN.

npm

```
npm install @mui/icons-material
```

yarn

```
yarn add @mui/icons-material
```

CDN

You can start using Material UI right away with minimal front-end infrastructure by installing it via CDN, which is a great option for rapid prototyping. Follow [this CDN example](#) to get started.

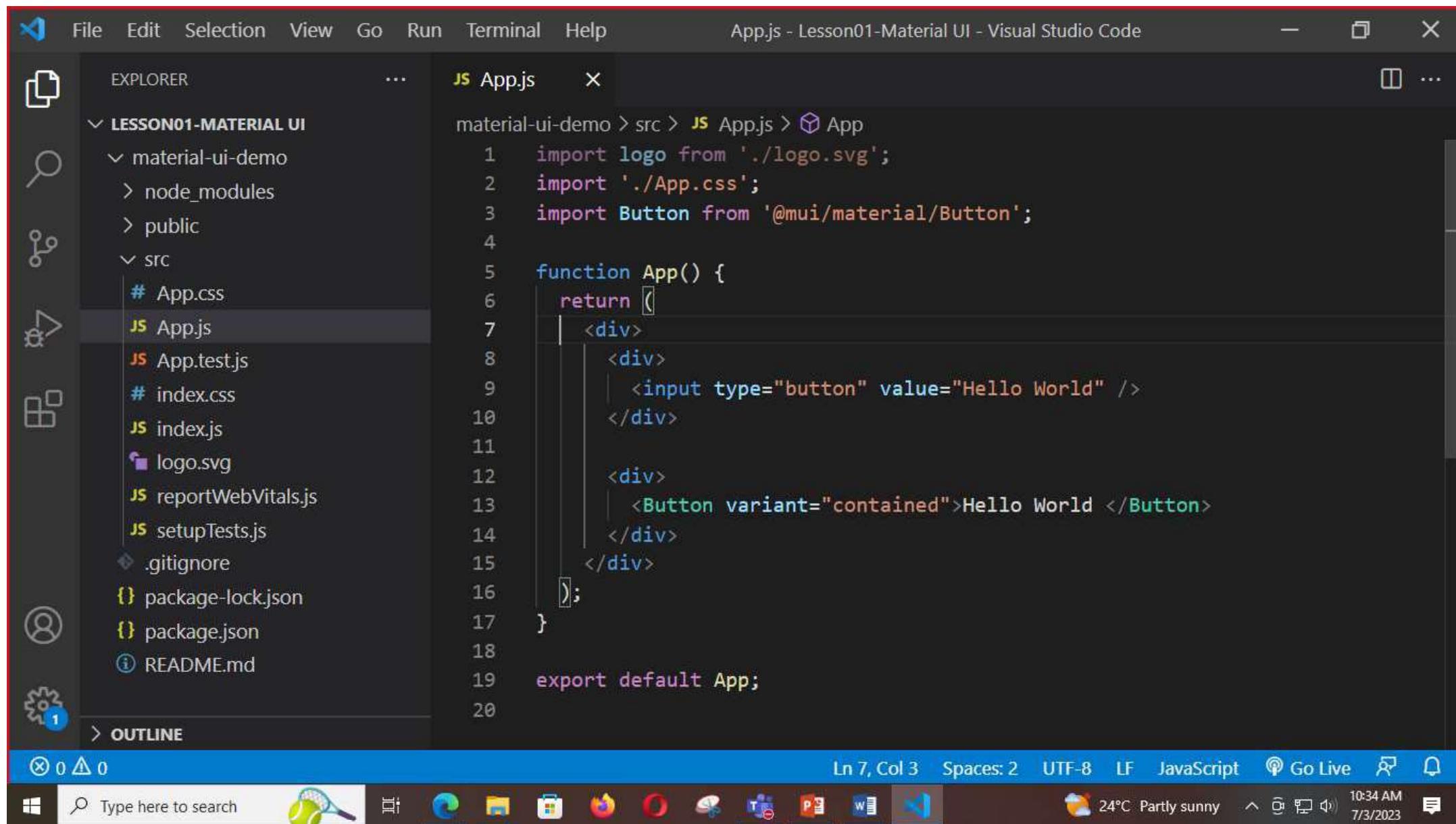
We do *not* recommend using this approach in production. It requires the client to download the entire library—regardless of which components are actually used—which negatively impacts performance and bandwidth utilization.

Two Universal Module Definition (UMD) files are provided:

- one for development: <https://unpkg.com/@mui/material@latest/umd/material-ui.development.js>
- one for production: <https://unpkg.com/@mui/material@latest/umd/material-ui.production.min.js>

The UMD links use the `latest` tag to point to the latest version of the library. This pointer is *unstable* and subject to change as we release new versions. You should consider pointing to a specific version, such as `v5.0.0`.

Lets See How to Add Material to React Application

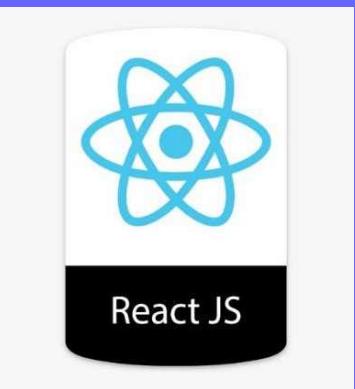


The screenshot shows the Visual Studio Code interface with the following details:

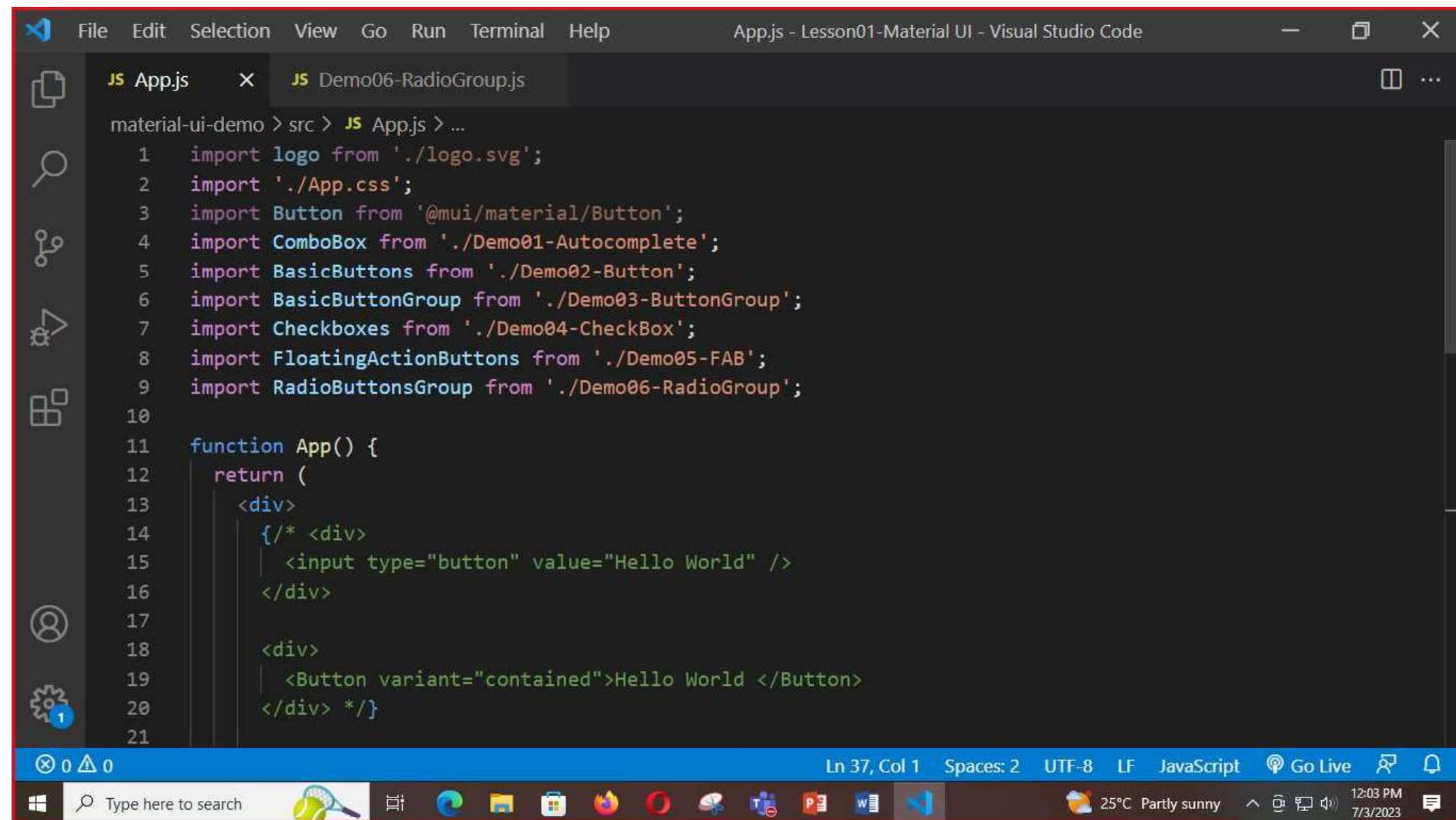
- File Menu:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** App.js - Lesson01-Material UI - Visual Studio Code.
- Explorer Panel:** Shows the project structure under "LESSON01-MATERIAL UI". The "src" folder contains "App.js", "index.css", "index.js", "logo.svg", "reportWebVitals.js", "setupTests.js", ".gitignore", "package-lock.json", "package.json", and "README.md".
- Code Editor:** The "App.js" file is open, displaying the following code:

```
material-ui-demo > src > JS App.js > App
1 import logo from './logo.svg';
2 import './App.css';
3 import Button from '@mui/material/Button';

4
5 function App() {
6   return (
7     <div>
8       <div>
9         <input type="button" value="Hello World" />
10      </div>
11
12      <div>
13        <Button variant="contained">Hello World </Button>
14      </div>
15    </div>
16  );
17}
18
19 export default App;
```
- Bottom Status Bar:** ShowsLn 7, Col 3, Spaces: 2, UTF-8, LF, JavaScript, Go Live, a clock (10:34 AM), weather (24°C Partly sunny), date (7/3/2023), and a message icon.
- Taskbar:** Shows icons for File Explorer, Search, Task List, and others.



Input Material UI Comps



File Edit Selection View Go Run Terminal Help App.js - Lesson01-Material UI - Visual Studio Code

JS App.js X JS Demo06-RadioGroup.js

```
material-ui-demo > src > JS App.js > ...
1 import logo from './logo.svg';
2 import './App.css';
3 import Button from '@mui/material/Button';
4 import ComboBox from './Demo01-Autocomplete';
5 import BasicButtons from './Demo02-Button';
6 import BasicButtonGroup from './Demo03-ButtonGroup';
7 import Checkboxes from './Demo04-CheckBox';
8 import FloatingActionButtons from './Demo05-FAB';
9 import RadioButtonsGroup from './Demo06-RadioGroup';

10
11 function App() {
12   return (
13     <div>
14       /* <div>
15         <input type="button" value="Hello World" />
16       </div>
17
18       <div>
19         <Button variant="contained">Hello World </Button>
20       </div> *)
21 }
```

Ln 37, Col 1 Spaces: 2 UTF-8 LF JavaScript ⚡ Go Live 🔍 🔔

0 0 0 Type here to search 25°C Partly sunny 12:03 PM 7/3/2023

File Edit Selection View Go Run Terminal Help App.js - Lesson01-Material UI - Visual Studio Code

JS App.js X JS Demo06-RadioGroup.js

material-ui-demo > src > JS App.js > ...

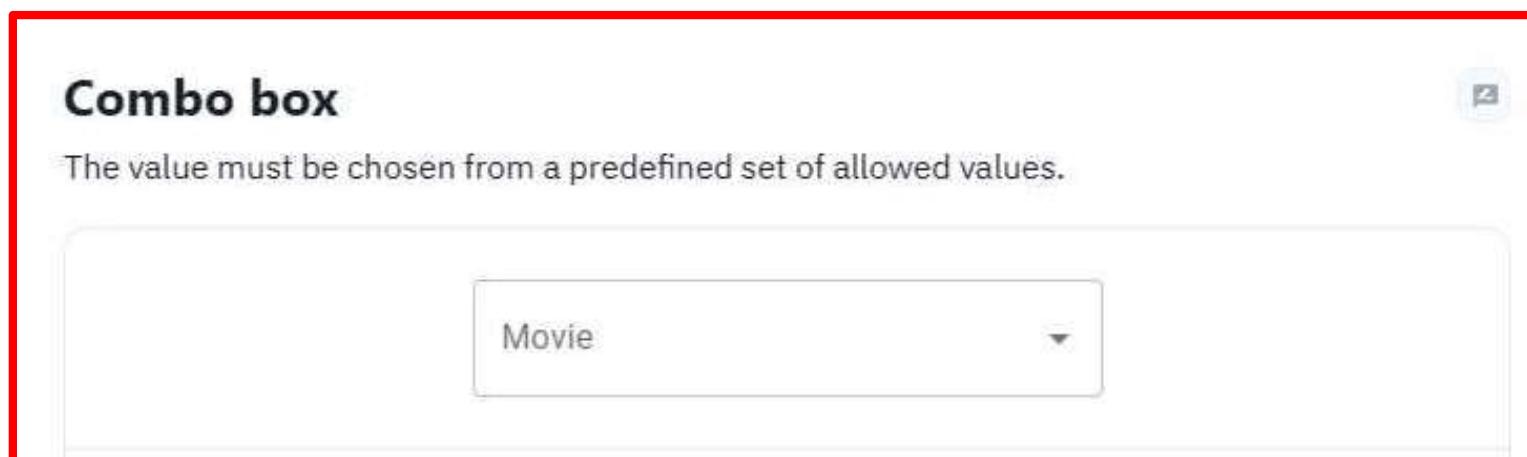
```
22 <ComboBox></ComboBox>
23
24 <BasicButtons></BasicButtons>
25
26 <BasicButtonGroup></BasicButtonGroup>
27
28 <Checkboxes></Checkboxes>
29
30 <FloatingActionButton (alias) function RadioButtonsGroup(): React.JSX.Element
31 import RadioButtonsGroup
32 <RadioButtonsGroup></RadioButtonsGroup>
33
34 </div>
35 );
36 }
37
38 export default App;
39
```

Ln 37, Col 1 Spaces: 2 UTF-8 LF JavaScript Go Live 12:03 PM 7/3/2023

Type here to search

Auto Complete

- The autocomplete is a normal text input enhanced by a panel of suggested options.
- The value for the textbox must be chosen from a predefined set of allowed values, e.g., a location field must contain a valid location name: combo box.
- The textbox may contain any arbitrary value, but it is advantageous to suggest possible values to the user, e.g., a search field may suggest similar or previous searches to save the user time: free solo.



The screenshot shows a Visual Studio Code interface with a dark theme. The top menu bar includes File, Edit, Selection, View, Go, Run, Terminal, and Help. The title bar displays "Demo01-Autocomplete.js - Lesson01-Material UI - Visual Studio Code". The left sidebar contains icons for file explorer, search, symbols, and other development tools. The main editor area has tabs for "App.js", "Demo01-Autocomplete.js" (which is currently active), and "package.json". The code in "Demo01-Autocomplete.js" is a React component named "ComboBox" that uses Material-UI's Autocomplete and TextField components. It defines a list of top 100 films from IMDb. The status bar at the bottom shows file counts (0△0), current position (Ln 14, Col 5), and encoding (UTF-8). The taskbar at the bottom includes icons for various applications like File Explorer, Edge, Task View, and others, along with system status indicators for weather (25°C Partly sunny) and date/time (11:08 AM 7/3/2023).

```
material-ui-demo > src > Demo01-Autocomplete.js > ComboBox
1 import * as React from 'react';
2 import TextField from '@mui/material/TextField';
3 import Autocomplete from '@mui/material/Autocomplete';
4
5 export default function ComboBox() {
6   return (
7     <Autocomplete
8       disablePortal
9       id="combo-box-demo"
10      options={top100Films}
11      sx={{ width: 300 }}
12      renderInput={(params) => <TextField {...params} label="Movie" />}
13    />
14  );
15}
16
17 // Top 100 films as rated by IMDb users. http://www.imdb.com/chart/top
18 const top100Films = [
19   { label: 'The Shawshank Redemption', year: 1994 },
20   { label: 'The Godfather', year: 1972 },
21   { label: 'The Godfather: Part II', year: 1974 },
```

Button

- Buttons allow users to take actions, and make choices, with a single tap.
- Buttons communicate actions that users can take. They are typically placed throughout your UI, in places like:
 - Modal windows
 - Forms
 - Cards
 - Toolbars

File Edit Selection View Go Run Terminal Help Demo02-Button.js - Lesson01-Material UI - Visual Studio Code

JS App.js JS Demo01-Autocomplete.js JS Demo02-Button.js X

material-ui-demo > src > JS Demo02-Button.js > BasicButtons

```
1 import * as React from 'react';
2 import Stack from '@mui/material/Stack';
3 import Button from '@mui/material/Button';

4
5 export default function BasicButtons() {
6     return (
7         <Stack spacing={2} direction="row">
8             <Button variant="text">Text</Button>
9             <Button variant="contained">Contained</Button>
10            <Button variant="outlined">Outlined</Button>
11
12            <Button>Primary</Button>
13            <Button disabled>Disabled</Button>
14            <Button href="#text-buttons">Link</Button>
15        </Stack>
16    );
17}
18
```

Java Full Stack Aspire Team 3 Training — 02:11:25



Kousil Lakkapragada

Ln 16, Col 1 Spaces: 4 UTF-8 CRLF JavaScript Go Live ⚡ 🔔 11:38 AM 25°C Partly sunny 7/3/2023

Type here to search

ButtonGroup

- The ButtonGroup component can be used to group related buttons.

Basic button group

The buttons can be grouped by wrapping them with the `ButtonGroup` component. They need to be immediate children.



ONE TWO THREE

The screenshot shows a Visual Studio Code interface with the following details:

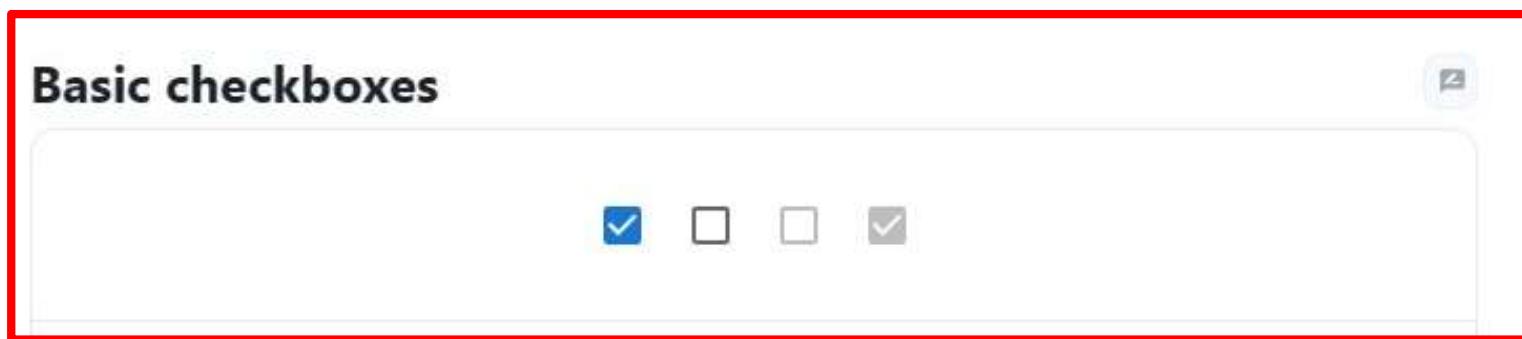
- Title Bar:** Demo03-ButtonGroup.js - Lesson01-Material UI - Visual Studio Code
- File Explorer:** Shows the project structure: material-ui-demo > src > Demo03-ButtonGroup.js > BasicButtonGroup.
- Code Editor:** The file Demo03-ButtonGroup.js contains the following code:

```
1 import * as React from 'react';
2 import Button from '@mui/material/Button';
3 import ButtonGroup from '@mui/material/ButtonGroup';

4
5 export default function BasicButtonGroup() {
6   return (
7     <ButtonGroup variant="contained" aria-label="outlined primary button group">
8       <Button>One</Button>
9       <Button>Two</Button>
10      <Button>Three</Button>
11    </ButtonGroup>
12  );
13}
```
- Bottom Status Bar:** ShowsLn 13, Col 2, Spaces: 4, UTF-8, CRLF, JavaScript, Go Live, system icons, and the date/time 11:44 AM 7/3/2023.

Checkbox

- Checkboxes allow the user to select one or more items from a set.
- Checkboxes can be used to turn an option on or off.
- If you have multiple options appearing in a list, you can preserve space by using checkboxes instead of on/off switches. If you have a single option, avoid using a checkbox and use an on/off switch instead.

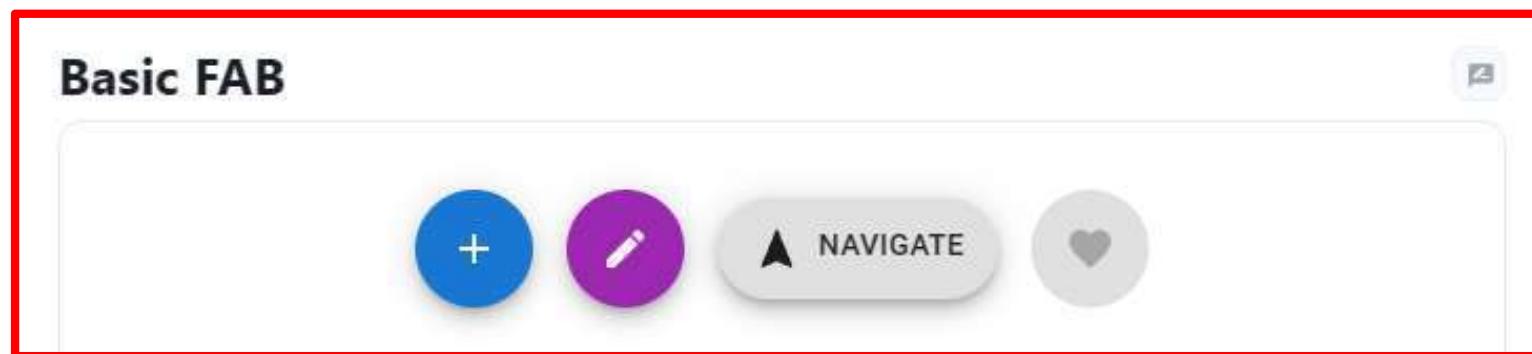


```
material-ui-demo > src > Demo04-CheckBox.js > Checkboxes
1 import * as React from 'react';
2 import Checkbox from '@mui/material/Checkbox';
3
4 const label = { inputProps: { 'aria-label': 'Checkbox demo' } };
5
6 export default function Checkboxes() {
7   return (
8     <div>
9       <Checkbox {...label} defaultChecked />
10      <Checkbox {...label} />
11      <Checkbox {...label} disabled />
12      <Checkbox {...label} disabled checked />
13    </div>
14  );
15}
```

The screenshot shows a Visual Studio Code window with the title "Demo04-CheckBox.js - Lesson01-Material UI - Visual Studio Code". The left sidebar contains icons for file operations like copy, search, and refresh. The main editor area displays a JavaScript file with code demonstrating different states of MUI checkboxes. The code uses the `@mui/material/Checkbox` component and includes props like `defaultChecked` and `checked` to show how they affect the appearance of the checkboxes. The status bar at the bottom shows the current line (Ln 12, Col 44), character count (7 selected), encoding (UTF-8), and file type (JavaScript). It also displays the date and time (11:59 AM, 7/3/2023) and system information (Windows 10, 25°C, Partly sunny).

Floating Action Button

- A Floating Action Button (FAB) performs the primary, or most common, action on a screen.
- A floating action button appears in front of all screen content, typically as a circular shape with an icon in its center. FABs come in two types: regular, and extended.
- Only use a FAB if it is the most suitable way to present a screen's primary action. Only one component is recommended per screen to represent the most common action.



File Edit Selection View Go Run Terminal Help Demo05-FAB.js - Lesson01-Material UI - Visual Studio Code

JS App.js JS Demo03-ButtonGroup.js JS Demo04-CheckBox.js JS Demo05-FAB.js X

material-ui-demo > src > JS Demo05-FAB.js > FloatingActionButtons

```
3 import Fab from '@mui/material/Fab';
4 import AddIcon from '@mui/icons-material/Add';
5 import EditIcon from '@mui/icons-material/Edit';
6 import FavoriteIcon from '@mui/icons-material/Favorite';
7 import NavigationIcon from '@mui/icons-material/Navigation';

8
9 export default function FloatingActionButtons() {
10   return (
11     <Box sx={{ '&' : {not(style): { m: 1 } } }}>
12       <Fab color="primary" aria-label="add">
13         <AddIcon />
14       </Fab>
15       <Fab color="secondary" aria-label="edit">
16         <EditIcon />
17       </Fab>
18       <Fab variant="extended">
19         <NavigationIcon sx={{ mr: 1 }} />
20         Navigate
21       </Fab>
22       <Fab disabled aria-label="like">
23         <FavoriteIcon />
24       </Fab>
25     </Box>
26   );
27 }
```

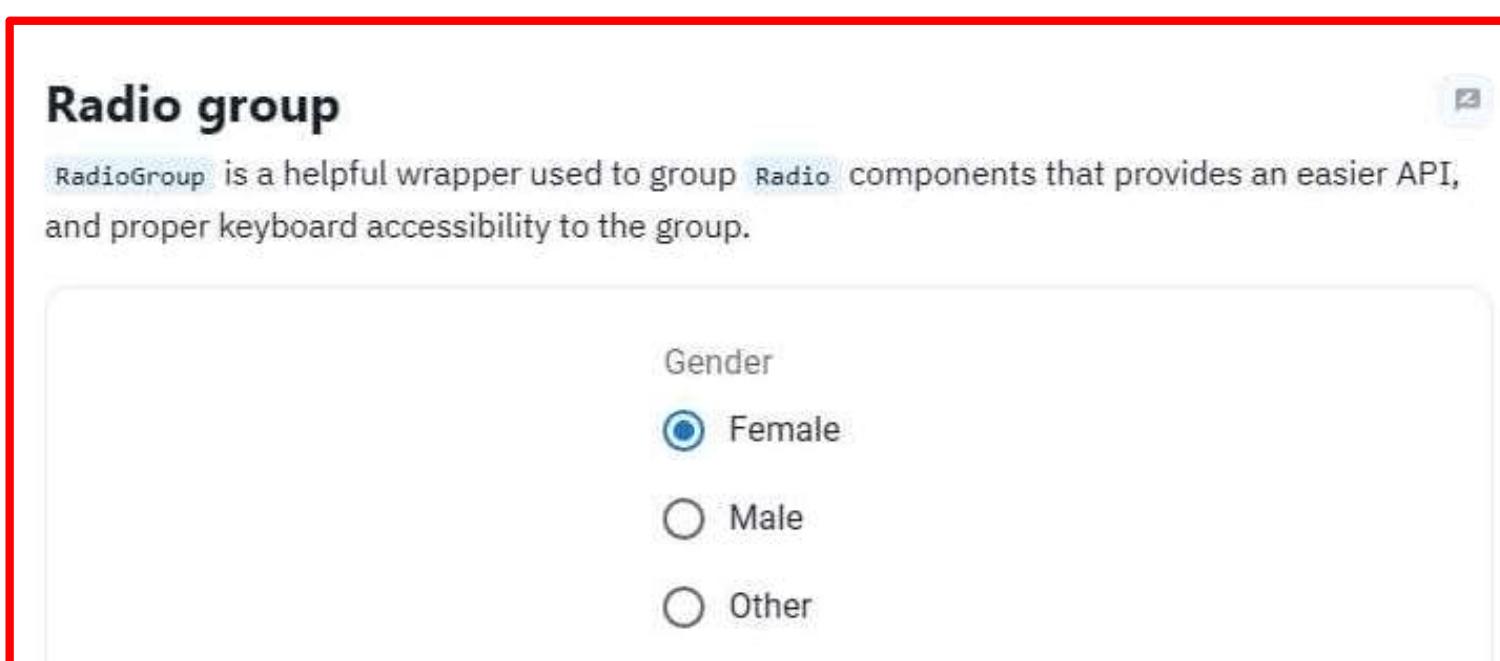
Ln 17, Col 13 Spaces: 4 UTF-8 CRLF JavaScript Go Live ⌂ ⓘ

Type here to search 11:59 AM 7/3/2023

25°C Partly sunny

Radio Group

- The Radio Group allows the user to select one option from a set.
- Use radio buttons when the user needs to see all available options. If available options can be collapsed, consider using a [Select component](#) because it uses less space.
- Radio buttons should have the most commonly used option selected by default.



The screenshot shows a Visual Studio Code interface with the following details:

- Title Bar:** Demo06-RadioGroup.js - Lesson01-Material UI - Visual Studio Code
- File Explorer:** Shows two files: App.js and Demo06-RadioGroup.js.
- Code Editor:** Displays the following code for a `RadioButtonsGroup` component:

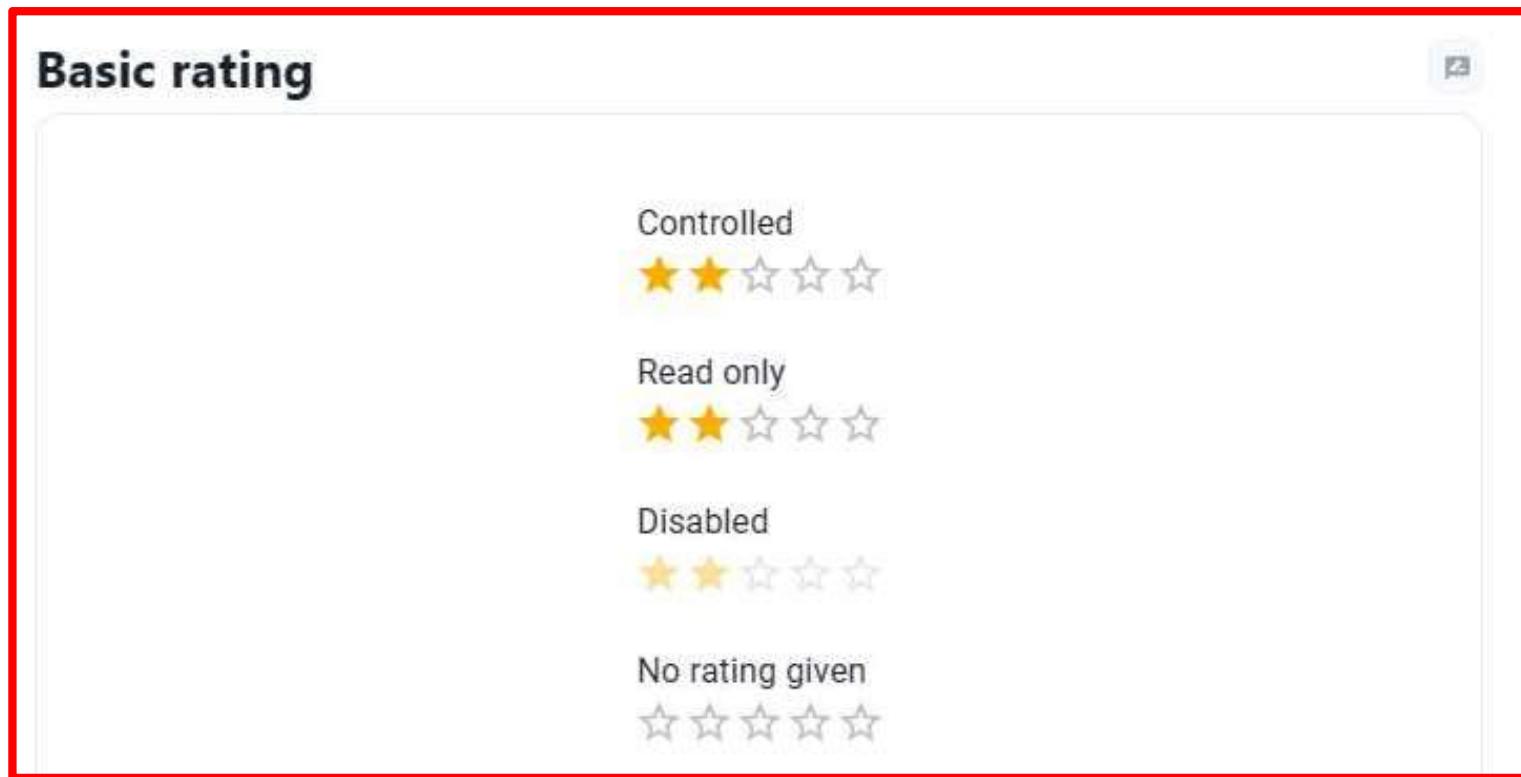
```
material-ui-demo > src > Demo06-RadioGroup.js > RadioButtonsGroup
1 import * as React from 'react';
2 import Radio from '@mui/material/Radio';
3 import RadioGroup from '@mui/material/RadioGroup';
4 import FormControlLabel from '@mui/material/FormControlLabel';
5 import FormControl from '@mui/material/FormControl';
6 import FormLabel from '@mui/material/FormLabel';
7
8 export default function RadioButtonsGroup() {
9     return (
10         <FormControl>
11             <FormLabel id="demo-radio-buttons-group-label">Gender</FormLabel>
12             <RadioGroup
13                 aria-labelledby="demo-radio-buttons-group-label"
14                 defaultValue="female"
15                 name="radio-buttons-group"
16             >
17                 <FormControlLabel value="female" control={<Radio />} label="Female" />
18                 <FormControlLabel value="male" control={<Radio />} label="Male" />
19                 <FormControlLabel value="other" control={<Radio />} label="Other" />
20             </RadioGroup>
21         </FormControl>
22     );
23 }
```

The code uses the `@mui/material` library to create a radio button group with three options: Female, Male, and Other. The `FormControl` and `FormLabel` components are also used.

Bottom Status Bar: ShowsLn 8, Col 42 (17 selected) Spaces: 4 UTF-8 CRLF JavaScript Go Live 12:04 PM 7/3/2023

Rating

- Ratings provide insight regarding others' opinions and experiences, and can allow the user to submit a rating of their own.



Demo07-Rating.js - Lesson01-Material UI - Visual Studio Code

File Edit Selection View Go Run Terminal Help

JS App.js JS Demo06-RadioGroup.js JS Demo07-Rating.js X

material-ui-demo > src > JS Demo07-Rating.js > BasicRating

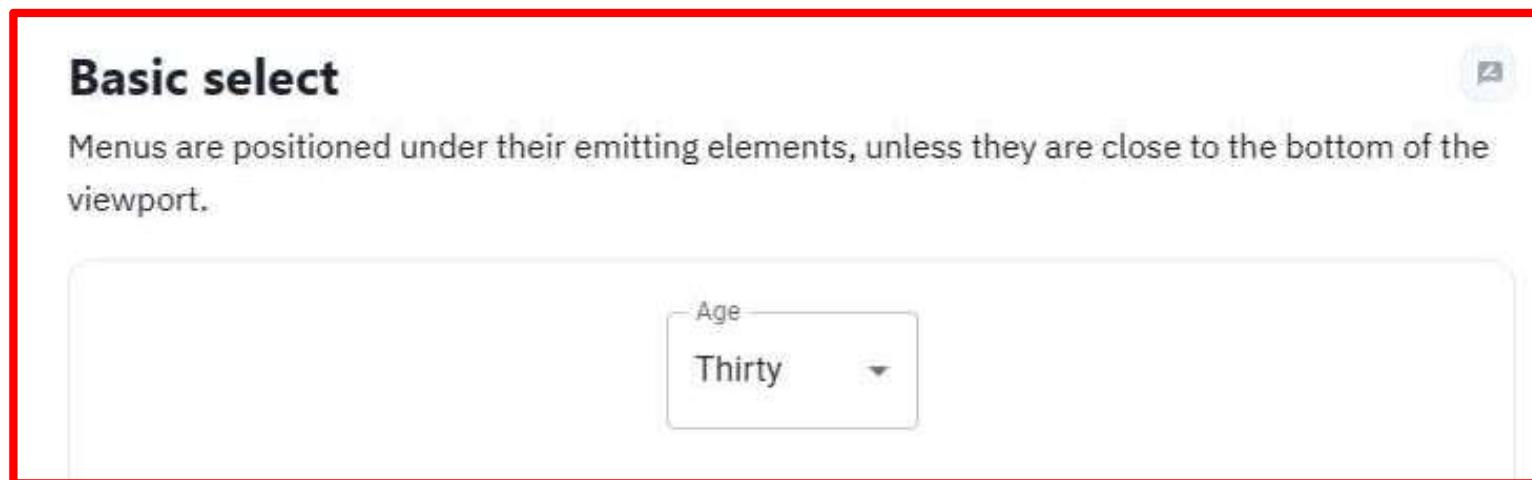
```
9  return (
10    <Box
11      sx={{
12        '& > legend': { mt: 2 },
13      }}
14    >
15      <Typography component="legend">Controlled</Typography>
16      <Rating
17        name="simple-controlled"
18        value={value}
19        onChange={(event, newValue) => {
20          setValue(newValue);
21        }}
22      />
23      <Typography component="legend">Read only</Typography>
24      <Rating name="read-only" value={value} readOnly />
25      <Typography component="legend">Disabled</Typography>
26      <Rating name="disabled" value={value} disabled />
27      <Typography component="legend">No rating given</Typography>
28      <Rating name="no-value" value={null} />
29    </Box>
```

Ln 7, Col 57 (4 selected) Spaces: 4 CRLF JavaScript Go Live 12:09 PM 7/3/2023

Type here to search

Select

- Select components are used for collecting user provided information from a list of options.



File Edit Selection View Go Run Terminal Help Demo08-Select.js - Lesson01-Material UI - Visual Studio Code

JS App.js JS Demo07-Rating.js JS Demo08-Select.js X

```
material-ui-demo > src > JS Demo08-Select.js > BasicSelect
  6 import Select from '@mui/material/Select';
  7
  8 export default function BasicSelect() {
  9   const [age, setAge] = React.useState('');
 10
 11   const handleChange = (event) => {
 12     setAge(event.target.value);
 13   };
 14
 15   return (
 16     <Box sx={{ minWidth: 120 }}>
 17       <FormControl fullWidth>
 18         <InputLabel id="demo-simple-select-label">Age</InputLabel>
 19         <Select
 20           labelId="demo-simple-select-label"
 21           id="demo-simple-select"
 22           value={age}
 23           label="Age"
 24           onChange={handleChange}
 25         >
 26           <MenuItem value={10}>Ten</MenuItem>
 27           <MenuItem value={20}>Twenty</MenuItem>
 28           <MenuItem value={30}>Thirty</MenuItem>
 29         </Select>
 30       </FormControl>
 31     </Box>
 32   );
 33 }
```

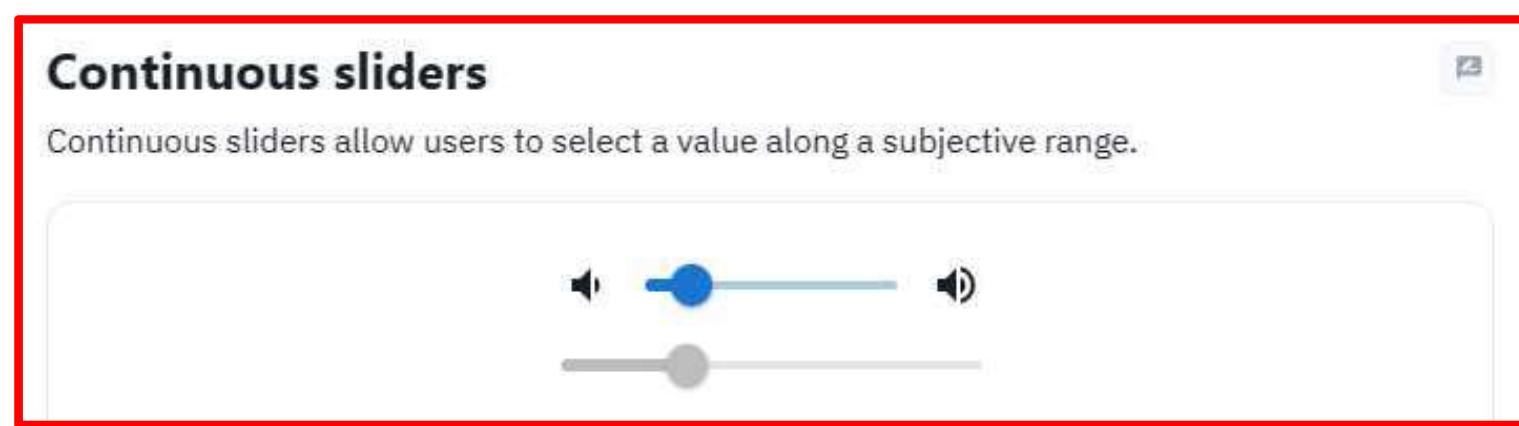
Ln 27, Col 49 Spaces: 4 UTF-8 CRLF JavaScript Go Live

Type here to search

12:17 PM 7/3/2023

Slider

- Sliders allow users to make selections from a range of values.
- Sliders reflect a range of values along a bar, from which users may select a single value. They are ideal for adjusting settings such as volume, brightness, or applying image filters.



File Edit Selection View Go Run Terminal Help Demo09-Slider.js - Lesson01-Material UI - Visual Studio Code

JS App.js JS Demo08-Select.js JS Demo09-Slider.js X

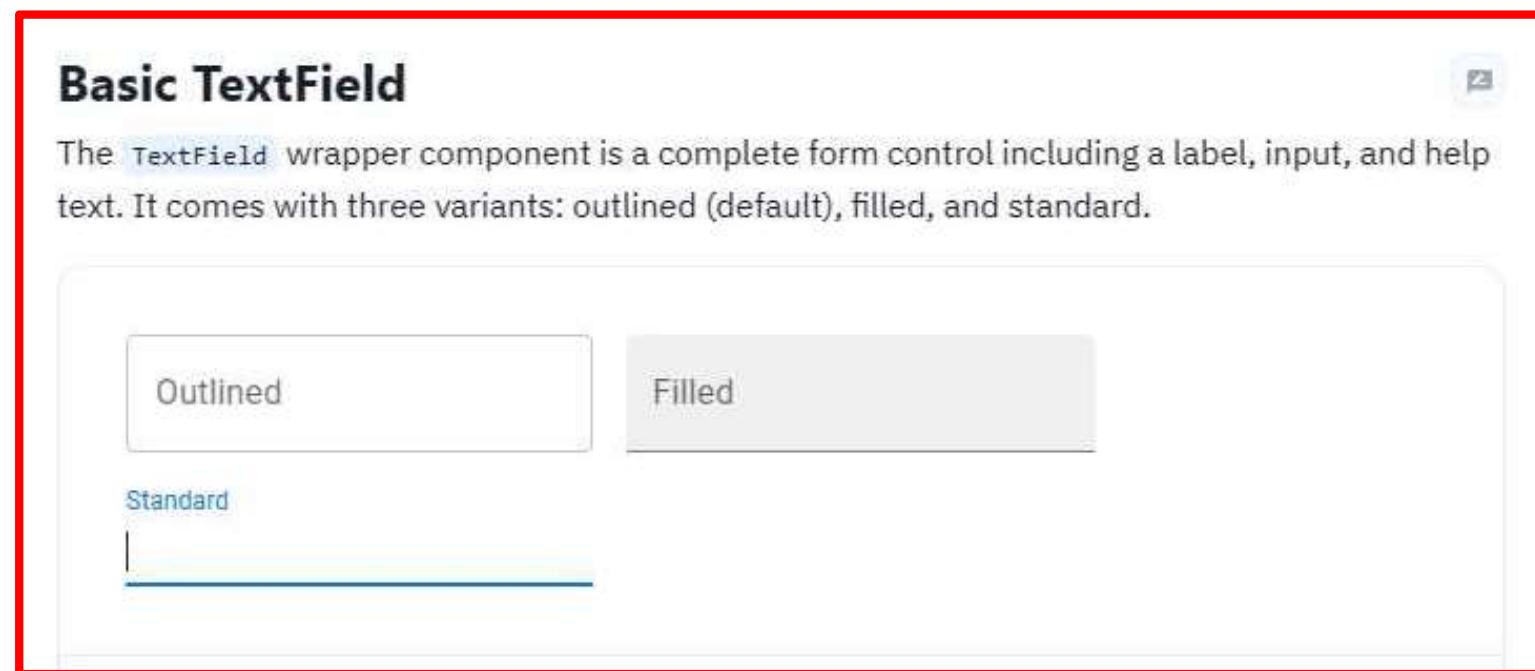
```
material-ui-demo > src > JS Demo09-Slider.js > ContinuousSlider
1 import * as React from 'react';
2 import Box from '@mui/material/Box';
3 import Stack from '@mui/material/Stack';
4 import Slider from '@mui/material/Slider';
5 import VolumeDown from '@mui/icons-material/VolumeDown';
6 import VolumeUp from '@mui/icons-material/VolumeUp';
7
8 export default function ContinuousSlider() {
9   const [value, setValue] = React.useState(30);
10
11   const handleChange = (event, newValue) => {
12     setValue(newValue);
13   };
14
15   return (
16     <Box sx={{ width: 200 }}>
17       <Stack spacing={2} direction="row" sx={{ mb: 1 }} alignItems="center">
18         <VolumeDown />
19         <Slider aria-label="Volume" value={value} onChange={handleChange} />
20         <VolumeUp />
21       </Stack>
22       <Slider disabled defaultValue={30} aria-label="Disabled slider" />
23     </Box>
24   );
25 }
```

Ln 25, Col 2 Spaces: 4 UTF-8 CRLF JavaScript Go Live ⚡ 12:24 PM 7/3/2023

Type here to search

TextField

- Text Fields let users enter and edit text.
- Text fields allow users to enter text into a UI. They typically appear in forms and dialogs.



Demo10-TextField.js - Lesson01-Material UI - Visual Studio Code

File Edit Selection View Go Run Terminal Help

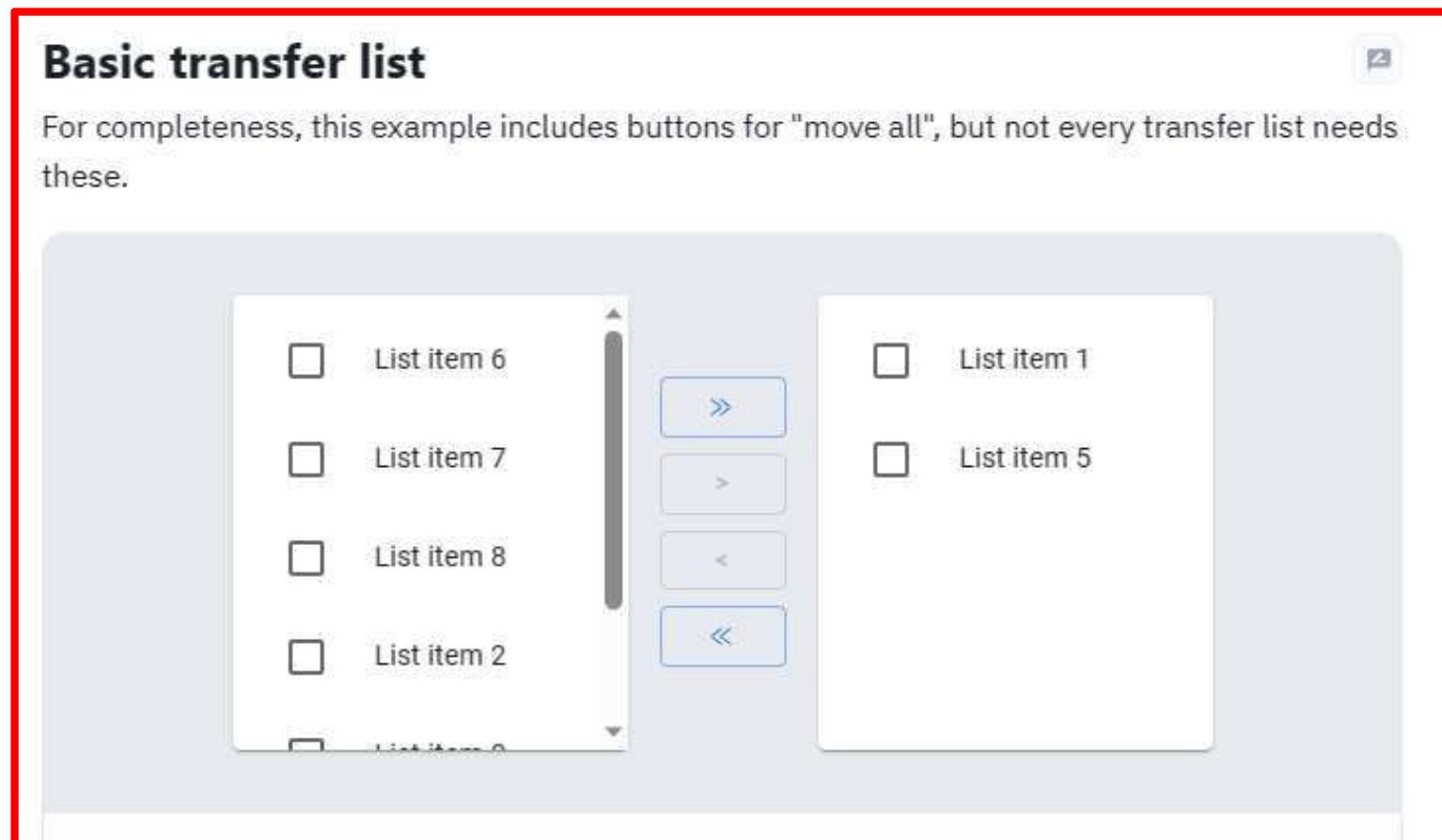
JS App.js JS Demo09-Slider.js JS Demo10-TextField.js X

```
material-ui-demo > src > JS Demo10-TextField.js > BasicTextFields
1 import * as React from 'react';
2 import Box from '@mui/material/Box';
3 import TextField from '@mui/material/TextField';
4
5 export default function BasicTextFields() {
6   return (
7     <Box
8       component="form"
9       sx={{
10         '& > :not(style)': { m: 1, width: '25ch' },
11       }}
12       noValidate
13       autoComplete="off"
14     >
15       <TextField id="outlined-basic" label="Outlined" variant="outlined" />
16       <TextField id="filled-basic" label="Filled" variant="filled" />
17       <TextField id="standard-basic" label="Standard" variant="standard" />
18     </Box>
19   );
20 }
```

Ln 20, Col 2 Spaces: 4 UTF-8 CRLF JavaScript Go Live 12:37 PM 26°C Partly sunny 7/3/2023

Transfer List / Shuttle

- A Transfer List (or "shuttle") enables the user to move one or more list items between lists.



The screenshot shows a Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** Demo11-TransferList.js - Lesson01-Material UI - Visual Studio Code.
- Explorer:** Shows a tree view of files under "LESSON01-MATERIAL UI". The "Demo11-TransferList.js" file is selected.
- Editor:** Displays the code for "Demo11-TransferList.js".
- Bottom Status Bar:** Shows line 143, column 2, spaces: 4, CRLF, JavaScript, Go Live, and system status (26°C, 12:44 PM, 7/3/2023).

```
material-ui-demo > src > Demo11-TransferList.js > TransferList
1 import * as React from 'react';
2 import Grid from '@mui/material/Grid';
3 import List from '@mui/material/List';
4 import ListItem from '@mui/material/ListItem';
5 import ListItemIcon from '@mui/material/ListItemIcon';
6 importListItemText from '@mui/material/ListItemText';
7 import Checkbox from '@mui/material/Checkbox';
8 import Button from '@mui/material/Button';
9 import Paper from '@mui/material/Paper';

10 function not(a, b) {
11   return a.filter((value) => b.indexOf(value) === -1);
12 }
13

14 function intersection(a, b) {
15   return a.filter((value) => b.indexOf(value) !== -1);
16 }
17

18 export default function TransferList() {
19   const [checked, setChecked] = React.useState([]);
20   const [left, setLeft] = React.useState([0, 1, 2, 3]);
```

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** Demo11-TransferList.js - Lesson01-Material UI - Visual Studio Code.
- Explorer View:** Shows the project structure under "LESSON01-MATERIAL UI". The file "Demo11-TransferList.js" is selected.
- Code Editor:** Displays the code for "Demo11-TransferList.js".
- Bottom Status Bar:** Includes icons for search, replace, and file operations, along with status information: Ln 143, Col 2, Spaces: 4, UTF-8, CRLF, JavaScript, Go Live, and a GitHub icon.

```
material-ui-demo > src > Demo11-TransferList.js > TransferList
export default function TransferList() {
  const [checked, setChecked] = React.useState([]);
  const [left, setLeft] = React.useState([0, 1, 2, 3]);
  const [right, setRight] = React.useState([4, 5, 6, 7]);

  const leftChecked = intersection(checked, left);
  const rightChecked = intersection(checked, right);

  const handleToggle = (value) => () => {
    const currentIndex = checked.indexOf(value);
    const newChecked = [...checked];

    if (currentIndex === -1) {
      newChecked.push(value);
    } else {
      newChecked.splice(currentIndex, 1);
    }

    setChecked(newChecked);
  };
}
```

```
material-ui-demo > src > Demo11-TransferList.js > TransferList
64   <List dense component="div" role="list">
65     {items.map((value) => {
66       const labelId = `transfer-list-item-${value}-label`;
67
68       return (
69         <ListItem
70           key={value}
71           role="listitem"
72           button
73           onClick={handleToggle(value)}
74         >
75           <ListItemIcon>
76             <Checkbox
77               checked={checked.indexOf(value) !== -1}
78               tabIndex={-1}
79               disableRipple
80               inputProps={{
81                 'aria-labelledby': labelId,
82               }}
83             />
84           </ListItemIcon>
```

Toggle Button

- A Toggle Button can be used to group related options.
- To emphasize groups of related Toggle buttons, a group should share a common container. The ToggleButtonGroup controls the selected state of its child buttons when given its own value prop.



File Edit Selection View Go Run Terminal Help Demo12-ToggleButton.js - Lesson01-Material UI - Visual Studio Code

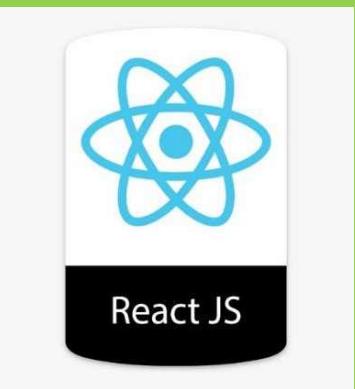
JS App.js JS Demo11-TransferList.js JS Demo12-ToggleButton.js X

material-ui-demo > src > JS Demo12-ToggleButton.js > ToggleButtons

```
9  export default function ToggleButtons() {
10    const [alignment, setAlignment] = React.useState('left');
11
12    const handleAlignment = (event, newAlignment) => {
13      setAlignment(newAlignment);
14    };
15
16    return (
17      <ToggleButtonGroup
18        value={alignment}
19        exclusive
20        onChange={handleAlignment}
21        aria-label="text alignment"
22      >
23        <ToggleButton value="left" aria-label="left aligned">
24          <FormatAlignLeftIcon />
25        </ToggleButton>
26        <ToggleButton value="center" aria-label="centered">
27          <FormatAlignCenterIcon />
28        </ToggleButton>
29        <ToggleButton value="right" aria-label="right aligned">
```

0 △ 0 Ln 32, Col 68 (8 selected) Spaces: 4 UTF-8 CRLF JavaScript Go Live 12:52 PM 7/3/2023

Type here to search



Data Display

Avatar

- Avatars are found throughout material design with uses in everything from tables to dialog menus.

Image avatars

Image avatars can be created by passing standard `img` props `src` or `srcSet` to the component.



Letter avatars

Avatars containing simple characters can be created by passing a string as `children`.



Demo01-Avatar.js - Lesson01-Material UI - Visual Studio Code

File Edit Selection View Go Run Terminal Help

JS App.js JS Demo01-Avatar.js X

```
material-ui-demo > src > Data-Display > JS Demo01-Avatar.js > ImageAvatars
1 import * as React from 'react';
2 import Avatar from '@mui/material/Avatar';
3 import Stack from '@mui/material/Stack';
4 import Image1 from './images/Image1.jpg'
5
6 export default function ImageAvatars() {
7   return (
8     <Stack direction="row" spacing={2}>
9       {/* <Avatar alt="Remy Sharp" src="Image1.jpg" /> */}
10      <Avatar alt="Remy Sharp" src={Image1} />
11      <Avatar alt="Travis Howard" src="/Data-Display/images/2.jpg" />
12      <Avatar alt="Cindy Baker" src="/Data-Display/images/3.jpg" />
13    </Stack>
14  );
15}
```

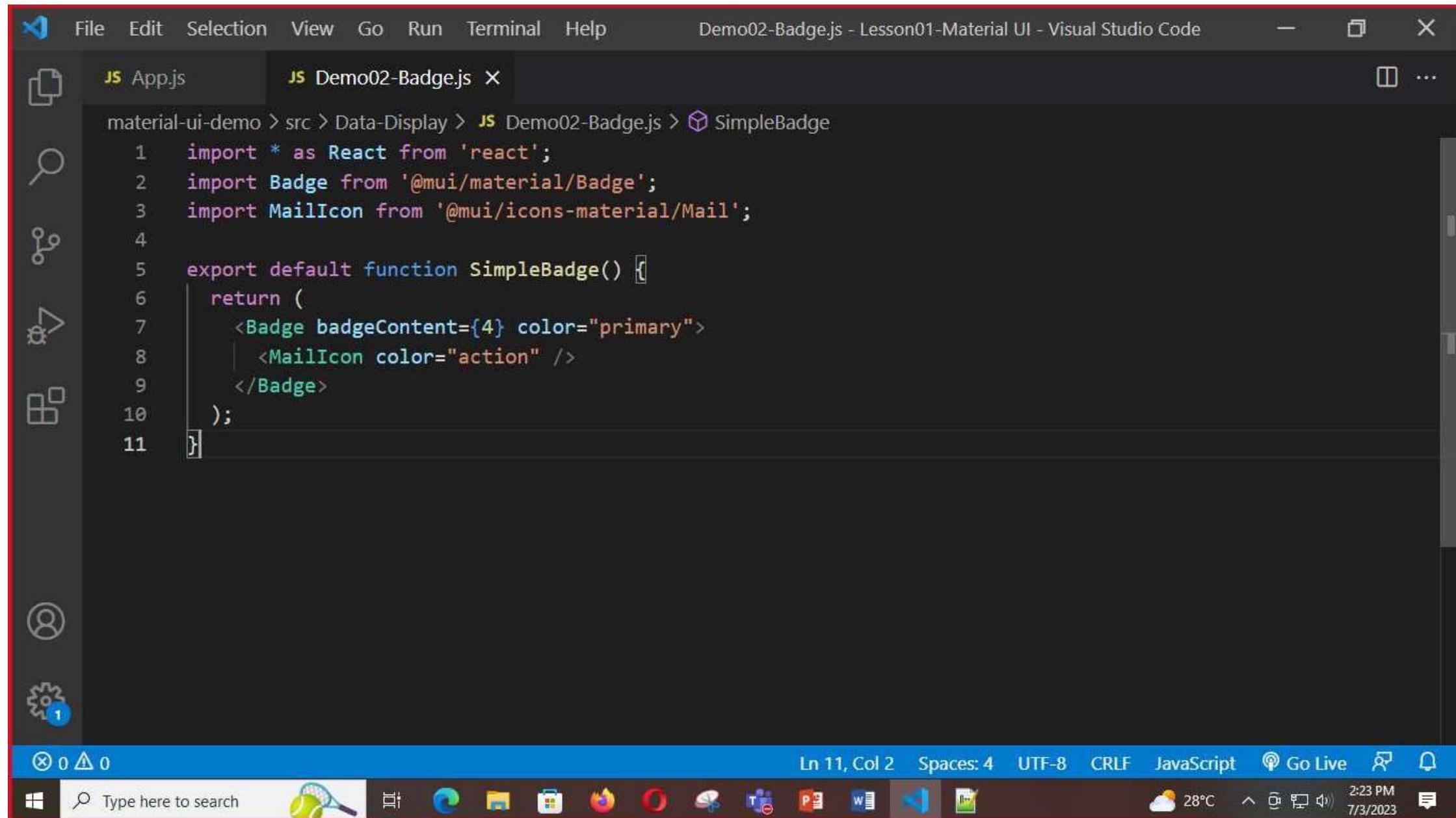
Ln 15, Col 2 Spaces: 4 UTF-8 CRLF JavaScript Go Live ⚡ 1:15 PM 7/3/2023

Badge

- Badge generates a small badge to the top-right of its child(ren).

Basic badge

Examples of badges containing text, using primary and secondary colors. The badge is applied to its children.



Demo02-Badge.js - Lesson01-Material UI - Visual Studio Code

File Edit Selection View Go Run Terminal Help

JS App.js JS Demo02-Badge.js X

material-ui-demo > src > Data-Display > JS Demo02-Badge.js > SimpleBadge

```
1 import * as React from 'react';
2 import Badge from '@mui/material/Badge';
3 import MailIcon from '@mui/icons-material/Mail';
4
5 export default function SimpleBadge() {
6   return (
7     <Badge badgeContent={4} color="primary">
8       <MailIcon color="action" />
9     </Badge>
10  );
11}
```

Ln 11, Col 2 Spaces: 4 UTF-8 CRLF JavaScript Go Live ⚡

Type here to search 2:23 PM 28°C 7/3/2023

Chips

- Chips are compact elements that represent an input, attribute, or action.
- Chips allow users to enter information, make selections, filter content, or trigger actions.
- While included here as a standalone component, the most common use will be in some form of input, so some of the behavior demonstrated here is not shown in context.

Basic chip

The `chip` component supports outlined and filled styling.

Chip Filled

Chip Outlined

A screenshot of the Visual Studio Code interface. The title bar reads "Demo03-Chips.js - Lesson01-Material UI - Visual Studio Code". The left sidebar has icons for file operations, search, and other development tools. The main editor window shows a file named "Demo03-Chips.js" with the following code:

```
material-ui-demo > src > Data-Display > Demo03-Chips.js > BasicChips
1 import * as React from 'react';
2 import Chip from '@mui/material/Chip';
3 import Stack from '@mui/material/Stack';
4
5 export default function BasicChips() {
6   return (
7     <Stack direction="row" spacing={1}>
8       <Chip label="Chip Filled" />
9       <Chip label="Chip Outlined" variant="outlined" />
10    </Stack>
11  );
12}
```

The status bar at the bottom shows "Ln 12, Col 2" and "Spaces: 4". It also includes icons for file operations, a search bar, and system notifications.

Divider

- A divider is a thin line that groups content in lists and layouts.
- Dividers separate content into clear groups.

List dividers

The divider renders as an `<hr>` by default. You can save rendering this DOM element by using the `divider` prop on the `ListItem` component.



```
graph TD; A[Inbox] --- B[Drafts]; B --- C[Trash]; C --- D[Spam]
```

File Edit Selection View Go Run Terminal Help Demo04-Divider.js - Lesson01-Material UI - Visual Studio Code

JS App.js JS Demo02-Badge.js JS Demo03-Chips.js JS Demo04-Divider.js X

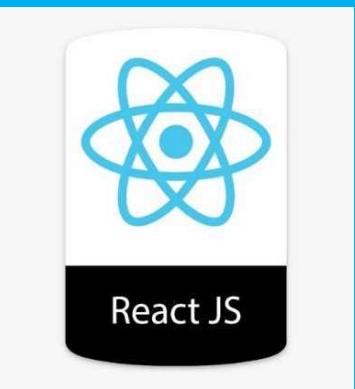
material-ui-demo > src > Data-Display > JS Demo04-Divider.js > ListDividers

```
11    };
12
13    export default function ListDividers() {
14        return (
15            <List sx={style} component="nav" aria-label="mailbox folders">
16                <ListItem button>
17                    <ListItemText primary="Inbox" />
18                </ListItem>
19                <Divider />
20                <ListItem button divider>
21                    <ListItemText primary="Drafts" />
22                </ListItem>
23                <ListItem button>
24                    <ListItemText primary="Trash" />
25                </ListItem>
26                <Divider light />
27                <ListItem button>
28                    <ListItemText primary="Spam" />
29                </ListItem>
30            </List>
31        );
32    }
33
```

Ln 32, Col 2 Spaces: 4 UTF-8 CRLF JavaScript Go Live

Type here to search

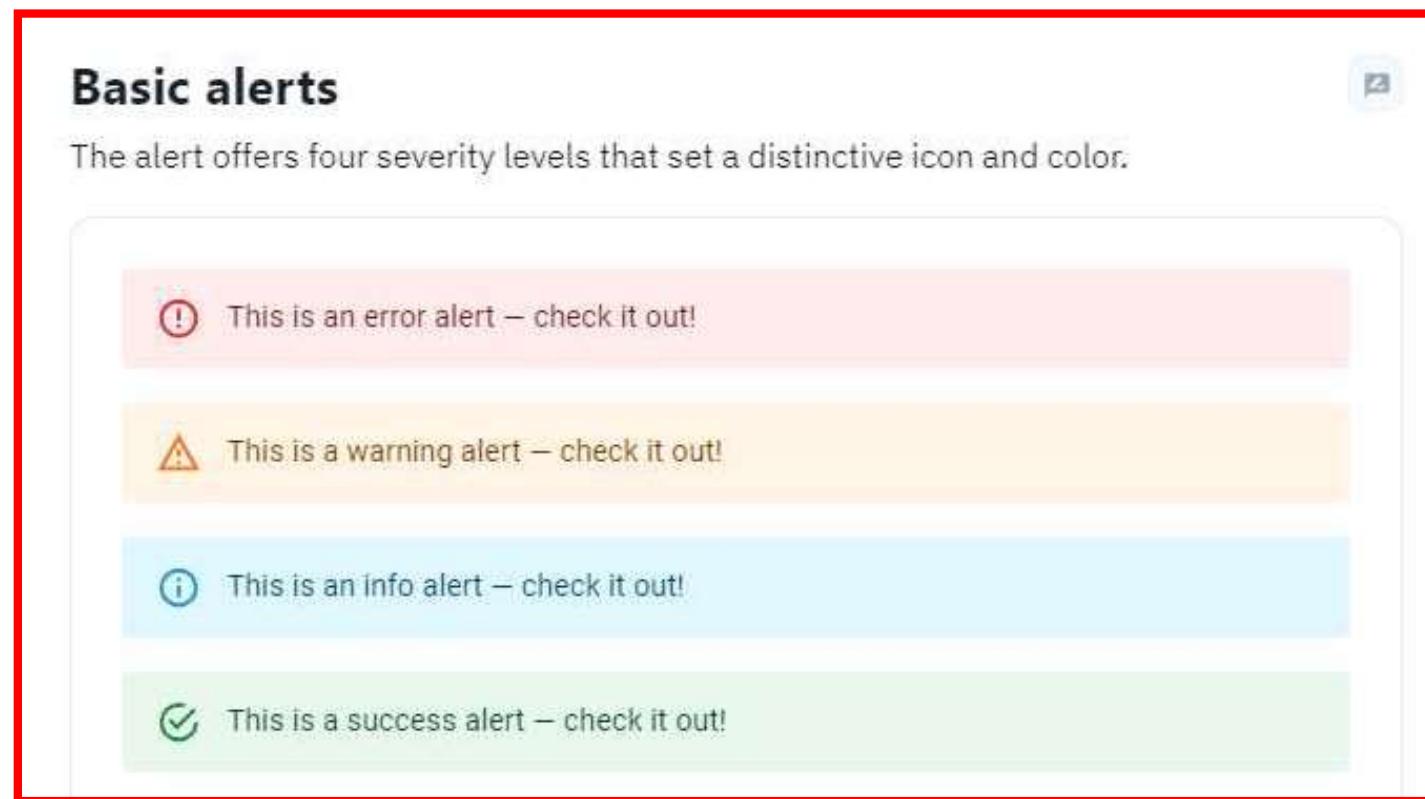
- Guidance and suggestions for using icons with Material UI.
- Material UI provides icon support in three ways:
 1. Standardized [Material Icons](#) exported as React components (SVG icons).
 2. With the [SvgIcon](#) component, a React wrapper for custom SVG icons.
 3. With the [Icon](#) component, a React wrapper for custom font icons.

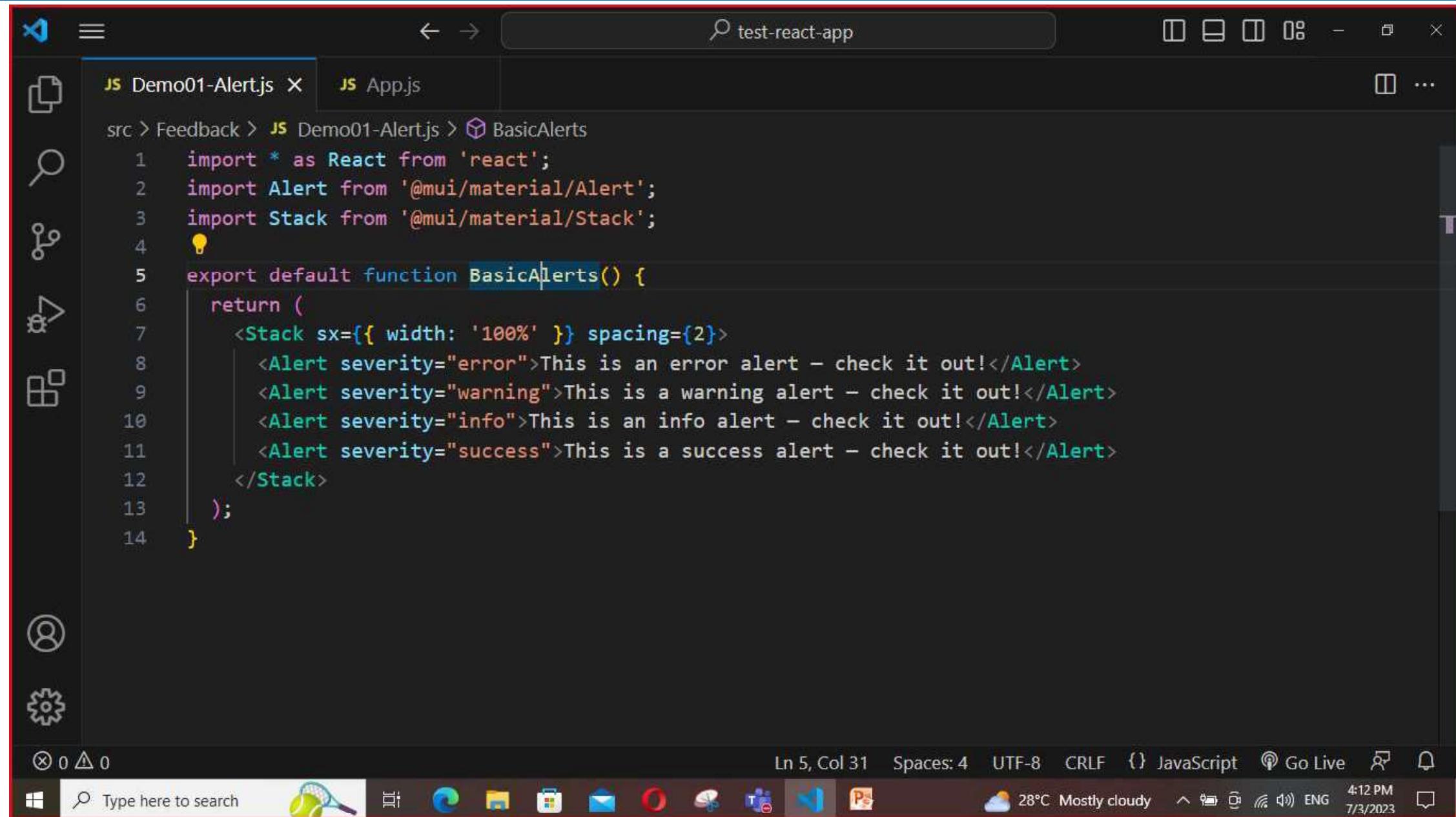


Feedback

Alert

- An alert displays a short, important message in a way that attracts the user's attention without interrupting the user's task.





The screenshot shows a Microsoft Edge browser window with a dark theme. The address bar displays "test-react-app". The left sidebar contains icons for file operations, search, and refresh. The main content area shows two tabs: "Demo01-Alert.js" (selected) and "App.js". The code in "Demo01-Alert.js" is as follows:

```
src > Feedback > Demo01-Alert.js > BasicAlerts
1 import * as React from 'react';
2 import Alert from '@mui/material/Alert';
3 import Stack from '@mui/material/Stack';
4
5 export default function BasicAlerts() {
6   return (
7     <Stack sx={{ width: '100%' }} spacing={2}>
8       <Alert severity="error">This is an error alert – check it out!</Alert>
9       <Alert severity="warning">This is a warning alert – check it out!</Alert>
10      <Alert severity="info">This is an info alert – check it out!</Alert>
11      <Alert severity="success">This is a success alert – check it out!</Alert>
12    </Stack>
13  );
14}
```

The browser status bar at the bottom shows "Ln 5, Col 31" and "Spaces: 4". The taskbar at the bottom includes icons for File, Home, Task View, Edge, File Explorer, Mail, OneDrive, Taskbar settings, Task View settings, and Power. The system tray shows the date and time as "4:12 PM 7/3/2023".

Backdrop

- The Backdrop component narrows the user's focus to a particular element on the screen.
- The Backdrop signals a state change within the application and can be used for creating loaders, dialogs, and more. In its simplest form, the Backdrop component will add a dimmed layer over your application.

Backdr X React A X Amazo X Auto H X CARM X CARM X AUTO X Autois X New ta X React A X +

Sign in

https://mui.com/material-ui/react-backdrop/

Chip Divider Icons Material Icons List Table Tooltip Typography FEEDBACK Alert Backdrop Dialog Progress Skeleton Snackbar SURFACES Accordion App Bar Card Paper

MUI X v6 is out! Discover what's new and get started now!

Search... Ctrl+K

Feedback Bundle size Figma Adobe Sketch

dimmed layer over your application.

Example

The demo below shows a basic Backdrop with a Circular Progress component in the foreground to indicate a loading state. After clicking **Show Backdrop**, you can click anywhere on the page to close it.

SHOW BACKDROP

```
<Button onClick={handleOpen}>Show backdrop</Button>
<Backdrop
  sx={{ color: '#fff', zIndex: (theme) => theme.zIndex.drawer + 1 }}
  open={open}
  onClick={ handleClose }
>
  <CircularProgress color="inherit" />
</Backdrop>
```

MUI stands in solidarity with the Ukrainian people against the Russian invasion.

Find out how you can help.

CONTENTS Example API

4:14 PM 28°C Mostly cloudy ENG 7/3/2023

- Dialogs inform users about a task and can contain critical information, require decisions, or involve multiple tasks.
- A Dialog is a type of **modal** window that appears in front of app content to provide critical information or ask for a decision. Dialogs disable all app functionality when they appear, and remain on screen until confirmed, dismissed, or a required action has been taken.
- Dialogs are purposefully interruptive, so they should be used sparingly.

The screenshot shows a web browser window displaying the Material-UI documentation for React dialogs at <https://mui.com/material-ui/react-dialog/>. The page title is "Basic dialog". The content area shows a modal titled "Set backup account" listing two accounts: "username@gmail.com" and "user02@gmail.com", with an "Add account" button at the bottom. To the right of the modal, there is explanatory text about touch mechanics: "Choosing an option" and "Touching outside of dialog". A note states that touching outside of the dialog closes the menu and the action. On the left, a sidebar lists various UI components like Chip, Divider, Icons, etc., with "Dialog" selected. On the right, a sidebar lists related topics such as Alerts, Transitions, Form dialogs, etc. At the bottom, there is a code snippet for the "Add account" button:

```
<Typography variant="subtitle1" component="div">
  Selected: {selectedValue}
</Typography>
<br />
<Button variant="outlined" onClick={handleClickOpen}>
```

JS Demo01-Alert.js JS App.js JS Demo03-Dialog.js X

src > Feedback > JS Demo03-Dialog.js > SimpleDialogDemo

```
30   return (
31     <Dialog onClose={handleClose} open={open}>
32       <DialogTitle>Set backup account</DialogTitle>
33       <List sx={{ pt: 0 }}>
34         {emails.map((email) => (
35           <ListItem disableGutters>
36             <ListItemIconButton onClick={() => handleListItemClick(email)} key={email}>
37               <ListItemIconAvatar>
38                 <Avatar sx={{ bgcolor: blue[100], color: blue[600] }}>
39                   <PersonIcon />
40                 </Avatar>
41               </ListItemIconAvatar>
42               <ListItemText primary={email} />
43             </ListItemIconButton>
44           </ListItem>
45         )));
46       <ListItem disableGutters>
47         <ListItemIconButton
48           autoFocus
49           onClick={() => handleListItemClick('addAccount')}>
```

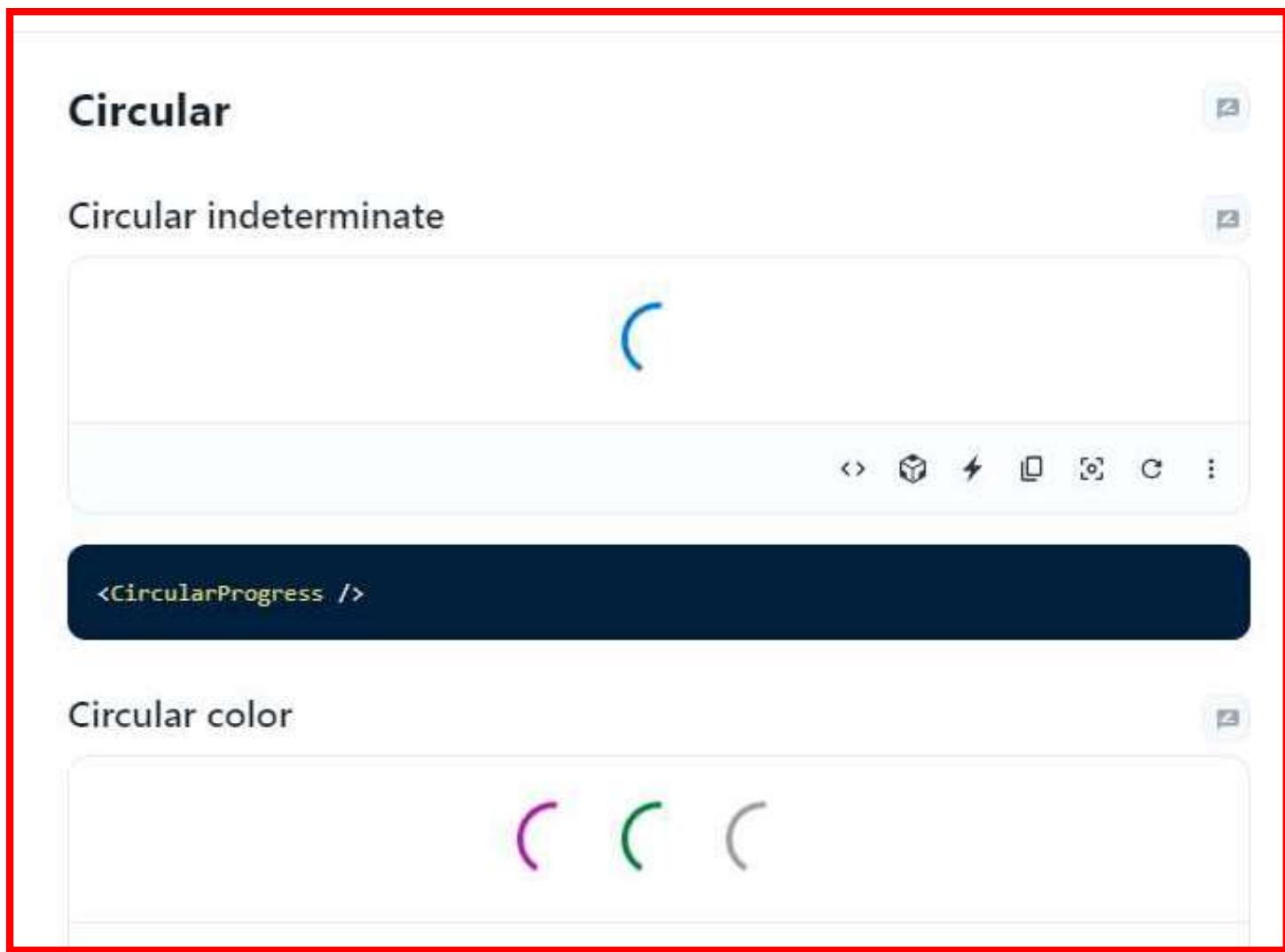
Ln 81, Col 5 Spaces: 4 UTF-8 CRLF {} JavaScript ⚡ Go Live 🔍 📰

⊗ 0 △ 0 Type here to search 🌐 🌐 🌐 🌐 🌐 🌐 🌐 🌐 🌐 🌐 🌐

Windows 10 icon 28°C Mostly cloudy 4:21 PM ENG 7/3/2023

Progress

- Progress indicators commonly known as spinners, express an unspecified wait time or display the length of a process.
- Progress indicators inform users about the status of ongoing processes, such as loading an app, submitting a form, or saving updates.
 - **Determinate** indicators display how long an operation will take.
 - **Indeterminate** indicators visualize an unspecified wait time.



Snackbar

- Snackbars provide brief notifications. The component is also known as a toast.
- Snackbars inform users of a process that an app has performed or will perform. They appear temporarily, towards the bottom of the screen. They shouldn't interrupt the user experience, and they don't require user input to disappear.
- Snackbars contain a single line of text directly related to the operation performed. They may contain a text action, but no icons. You can use them to display notifications.

Simple snackbars

A basic Snackbar that aims to reproduce Google Keep's Snackbar behavior.

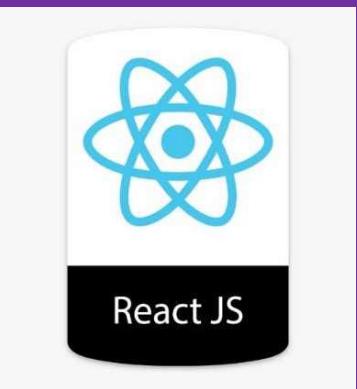
OPEN SIMPLE SNACKBAR



```
<Button onClick={handleClick}>Open simple Snackbar</Button>
<Snackbar
  open={open}
  autoHideDuration={6000}
  onClose={handleClose}
  message="Note archived"
  action={action}
/>
```

A screenshot of Microsoft Visual Studio Code (VS Code) displaying a React component named `Demo05-SnackBar.js`. The code is written in JavaScript and uses the Material UI library. The component contains a button that triggers a `Snackbar` when clicked. The `Snackbar` has a message "Note archived" and an action button. The code is part of a larger application structure, with imports from `Feedback` and `IconButton`.

```
src > Feedback > JS Demo05-SnackBar.js > SimpleSnackbar
  30   color="inherit"
  31   onClick={handleClose}
  32   >
  33     <CloseIcon fontSize="small" />
  34   </IconButton>
  35   </React.Fragment>
  36 );
  37
  38 return (
  39   <div>
  40     <Button onClick={handleClick}>Open simple snackbar</Button>
  41     <Snackbar
  42       open={open}
  43       autoHideDuration={6000}
  44       onClose={handleClose}
  45       message="Note archived"
  46       action={action}
  47     />
  48   </div>
  49 );
```



Surfaces

Accordion

- The accordion component allows the user to show and hide sections of related content on a page.
- An accordion is a lightweight container that may either be used standalone, or be connected to a larger surface, such as a card.



File Edit Selection View Go ... ← → test-react-app

JS Demo01-Alert.js JS App.js JS Demo01-Accordion.js X

src > Surfaces > JS Demo01-Accordion.js > SimpleAccordion

```
9  return (
10 <div>
11   <Accordion>
12     <AccordionSummary
13       expandIcon={<ExpandMoreIcon />}
14       aria-controls="panel1a-content"
15       id="panel1a-header"
16     >
17       <Typography>Accordion 1</Typography>
18     </AccordionSummary>
19     <AccordionDetails>
20       <Typography>
21         Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse
22         malesuada lacus ex, sit amet blandit leo lobortis eget.
23       </Typography>
24     </AccordionDetails>
25   </Accordion>
26   <Accordion>
27     <AccordionSummary
28       expandIcon={<ExpandMoreIcon />}
29       aria-controls="panel2a-content"
30       id="panel2a-header"
31     >
32       <Typography>Accordion 2</Typography>
33     </AccordionSummary>
34     <AccordionDetails>
```

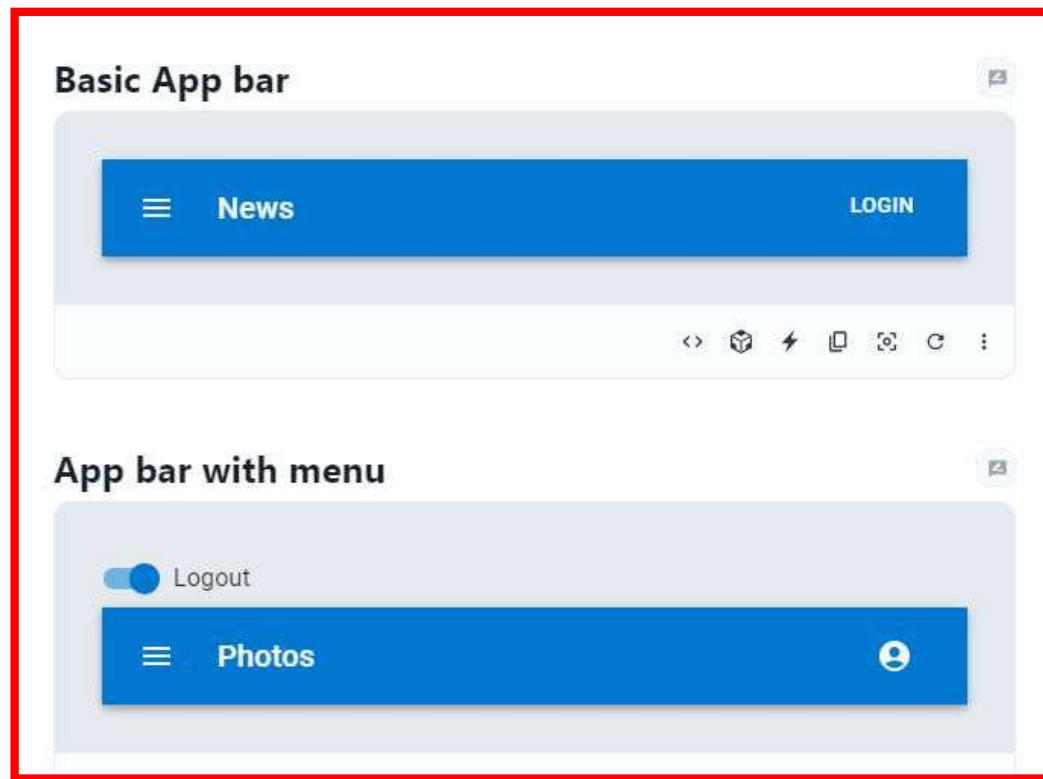
Ln 8, Col 40 (15 selected) Spaces: 4 UTF-8 CRLF {} JavaScript ⚡ Go Live

Type here to search

4:42 PM 28°C Mostly cloudy ENG 7/3/2023

App Bar

- The App Bar displays information and actions relating to the current screen.
- The top App bar provides content and actions related to the current screen. It's used for branding, screen titles, navigation, and actions.



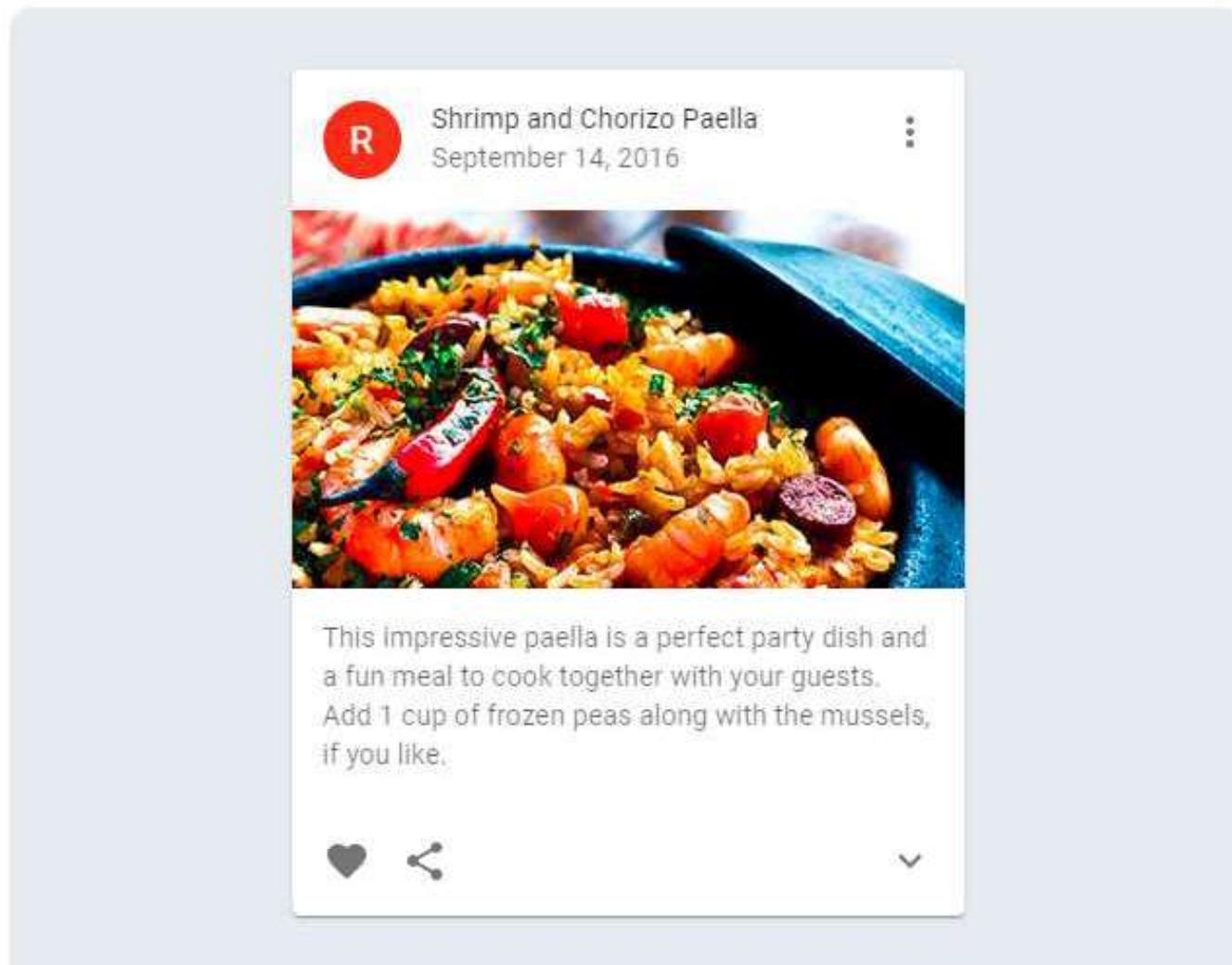
```
File Edit Selection View Go ...
File Explorer Search test-react-app
OPEN EDITORS
TEST-REACT-APP
public
src
Feedback
Demo01-Alert.js
Demo02-Backdrop.js
Demo03-Dialog.js
Demo04-Progress.js
Demo05-SnackBar.js
Surfaces
Demo01-Accordion.js
Demo02-AppBar.js
# App.css
App.js
App.test.js
# index.css
index.js
logo.svg
reportWebVitals.js
setupTests.js
OUTLINE
TIMELINE
Type here to search
JS Demo01-Alert.js JS App.js JS Demo01-Accordion.js JS Demo02-AppBar.js X
src > Surfaces > JS Demo02-AppBar.js > ButtonAppBar
8 import MenuIcon from '@mui/icons-material/Menu';
9
10 export default function ButtonAppBar() {
11   return (
12     <Box sx={{ flexGrow: 1 }}>
13       <AppBar position="static">
14         <Toolbar>
15           <IconButton
16             size="large"
17             edge="start"
18             color="inherit"
19             aria-label="menu"
20             sx={{ mr: 2 }}
21           >
22             <MenuIcon />
23           </IconButton>
24           <Typography variant="h6" component="div" sx={{ flexGrow: 1 }}>
25             News
26           </Typography>
27           <Button color="inherit">Login</Button>
28         </Toolbar>
29       </AppBar>
30     </Box>
31   );
32 }
```

Ln 32, Col 2 Spaces: 4 UTF-8 CRLF {} JavaScript ⚡ Go Live ⌂ 4:45 PM 28°C Mostly cloudy ⌂ ENG 7/3/2023

- Cards contain content and actions about a single subject.
- Cards are surfaces that display content and actions on a single topic.
- They should be easy to scan for relevant and actionable information. Elements, like text and images, should be placed on them in a way that clearly indicates hierarchy.

Complex Interaction

On desktop, card content can expand. (Click the downward chevron to view the recipe.)



A screenshot of a Windows desktop environment. At the top is a Microsoft Edge browser window titled "test-react-app". The browser shows a React application with several tabs open: "Demo01-Alert.js", "App.js", "Demo03-Card.js" (which is the active tab), "Demo01-Accordion.js", and "Demo02-AppBar.js". The "Demo03-Card.js" tab displays code for a card component. The code uses Material UI components like `<Card>`, `<CardHeader>`, `<Avatar>`, `<IconButton>`, and `<CardMedia>`. It also includes styling with `sx` and `styled-components` properties. The file path shown in the browser is `src > Surfaces > Demo03-Card.js`. The browser's status bar at the bottom indicates "Ln 29, Col 41 (16 selected)" and "Spaces: 4". Below the browser is a Windows taskbar with icons for File Explorer, Edge, Mail, OneDrive, Task View, and Power Shell. The system tray shows the date and time as "7/3/2023 4:53 PM". On the left side of the screen, there is a vertical sidebar with icons for File Explorer, Search, Task View, Task List, and a gear icon for settings.

```
src > Surfaces > Demo03-Card.js > RecipeReviewCard
36 return (
37   <Card sx={{ maxWidth: 345 }}>
38     <CardHeader
39       avatar={
40         <Avatar sx={{ bgcolor: red[500] }} aria-label="recipe">
41           R
42         </Avatar>
43       }
44       action={
45         <IconButton aria-label="settings">
46           <MoreVertIcon />
47         </IconButton>
48       }
49       title="Shrimp and Chorizo Paella"
50       subheader="September 14, 2016"
51     />
52     <CardMedia
53       component="img"
54       height="194"
55       image="/static/images/cards/paella.jpg"
56       alt="Paella dish"
57     />
58     <CardContent>
59       <Typography variant="body2" color="text.secondary">
60         This impressive paella is a perfect party dish and a fun meal to cook

```