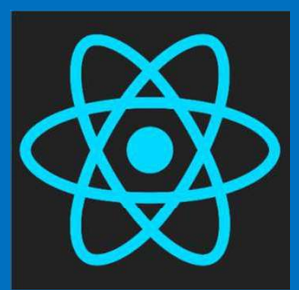
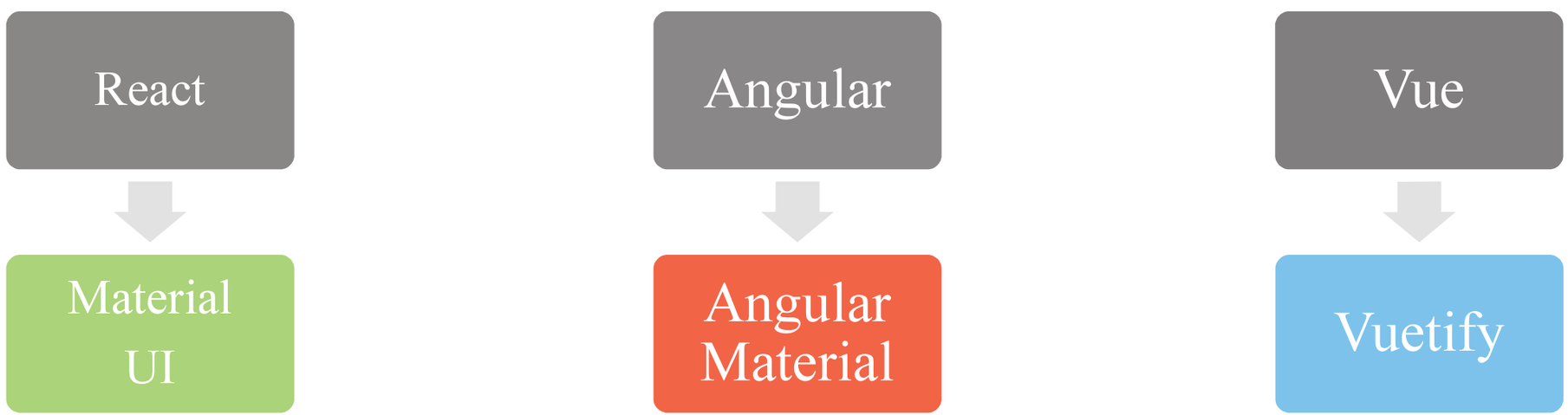
1 Creating Beautiful Apps with Material UI



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.

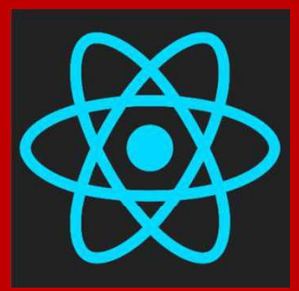
Advantages of Material UI

- **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

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
```

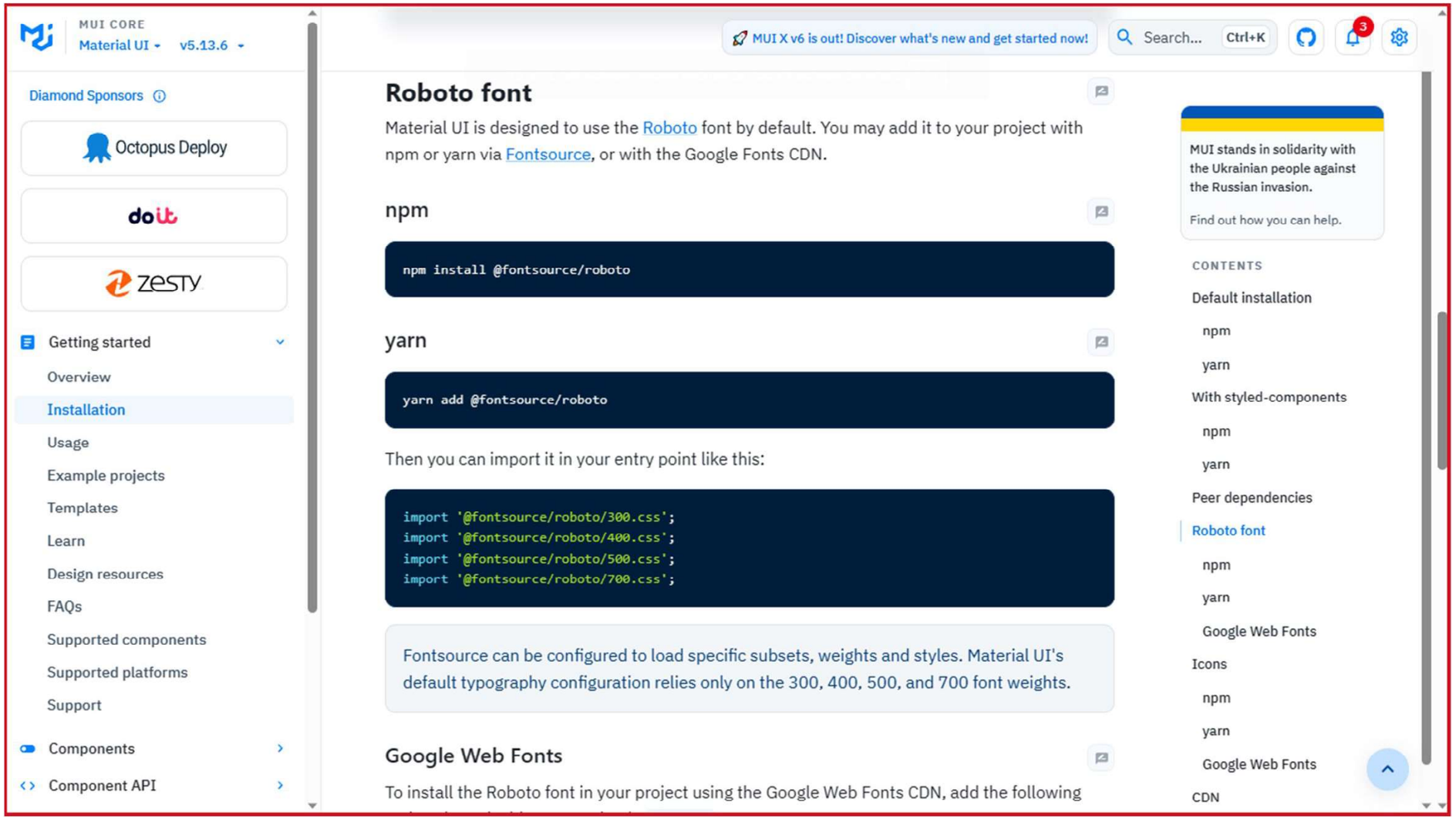
Visit the Styled engine guide for more information about how to configure styled-components.

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


MUI CORE
Material UI
v5.13.6
MUI X v6 is out! Discover what's new and get started now!
Search...
Ctrl+K
Diamond Sponsors
Octopus Deploy
doit
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 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
Roboto 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.

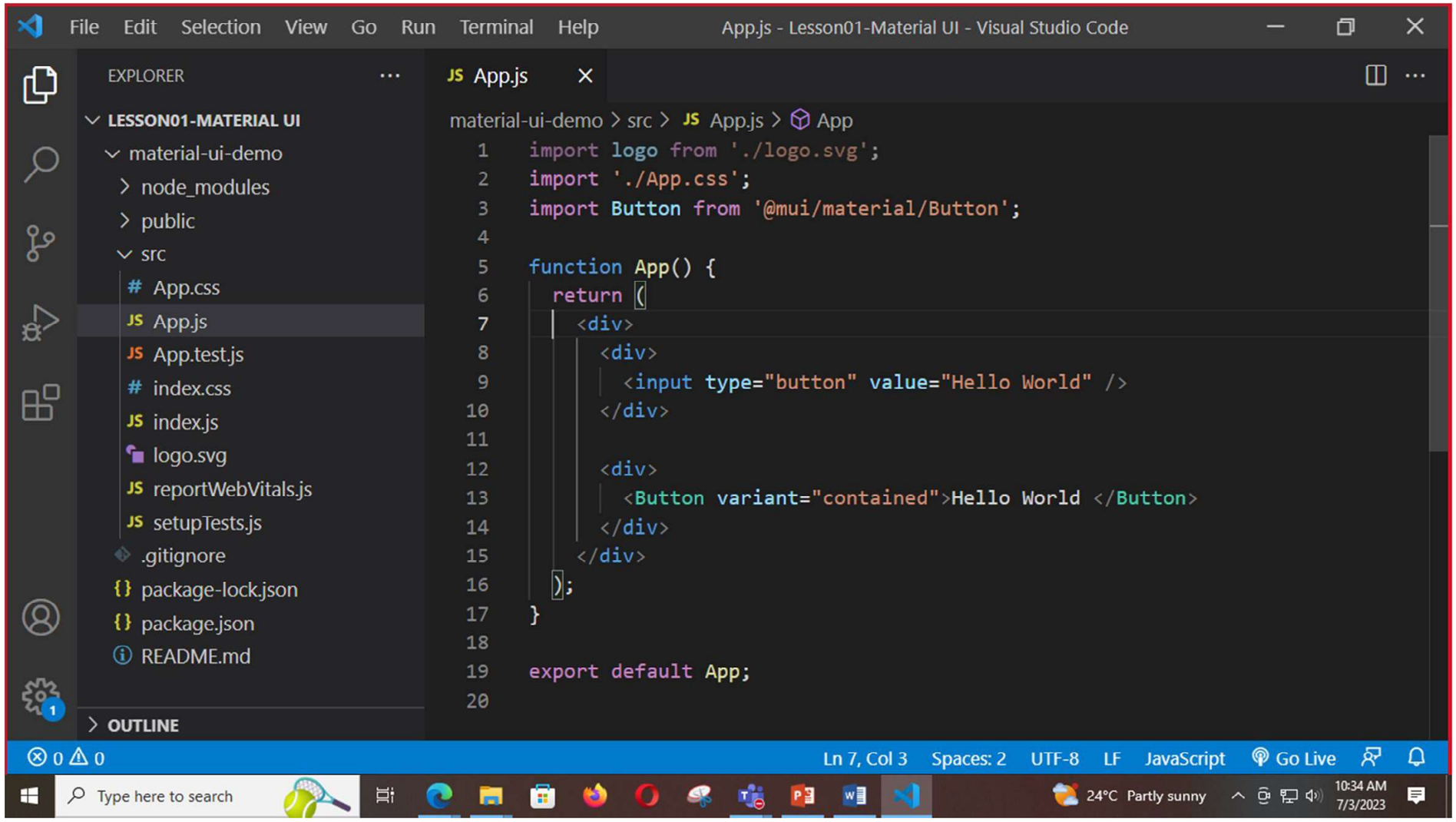
We do *not* recommend using this approach in production. It requires the client to download the entire library—regardless of which components are actually used—which negatively impacts performance and bandwidth utilization.

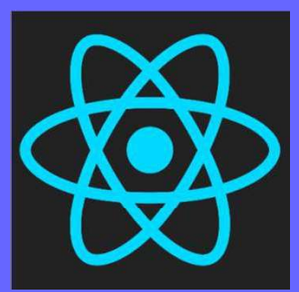
Two Universal Module Definition (UMD) files are provided:

- 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

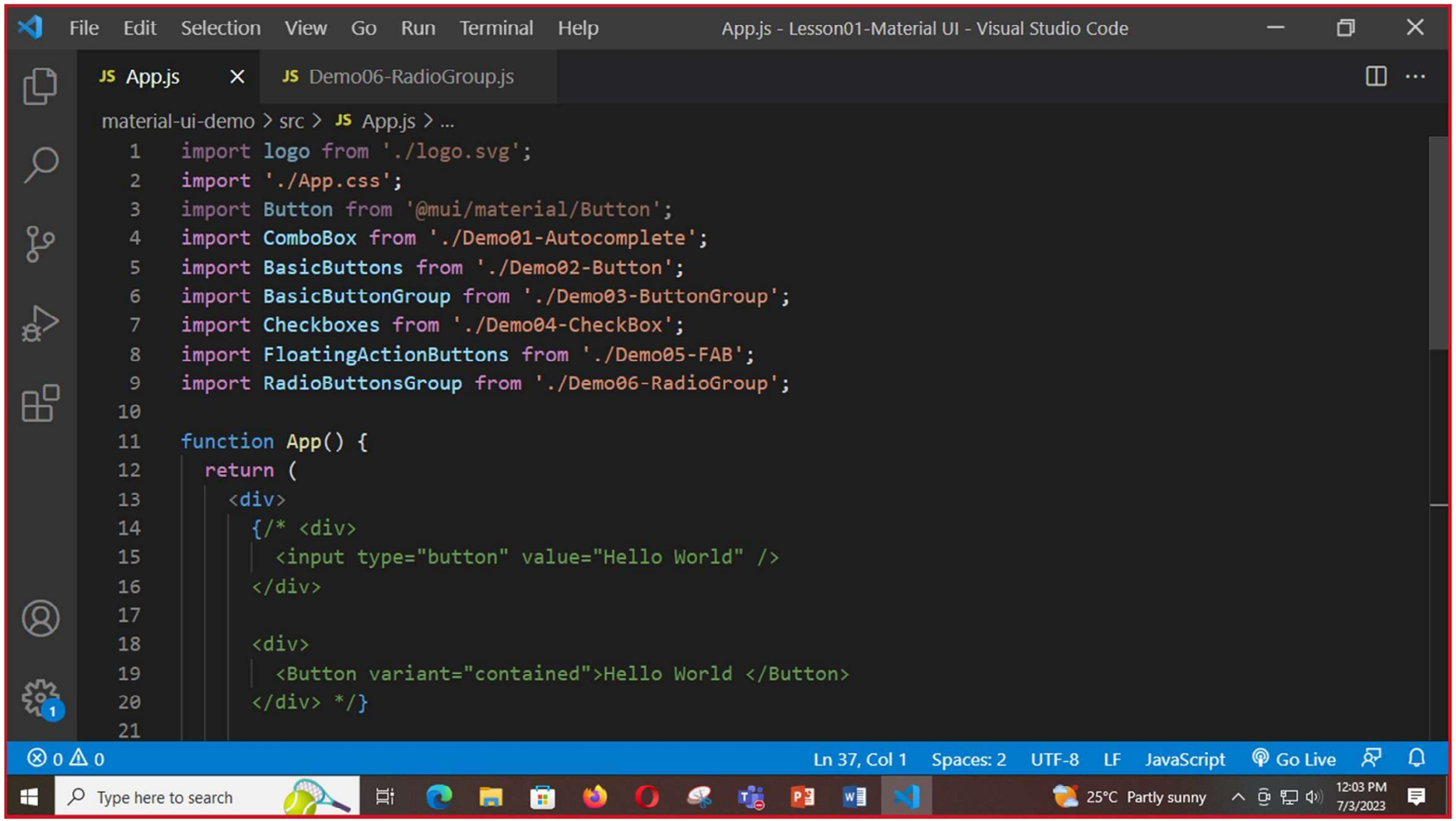
The UMD links use the `latest` tag to point to the latest version of the library. This pointer is *unstable* and subject to change as we release new versions. You should consider pointing to a specific version, such as v5.0.0.

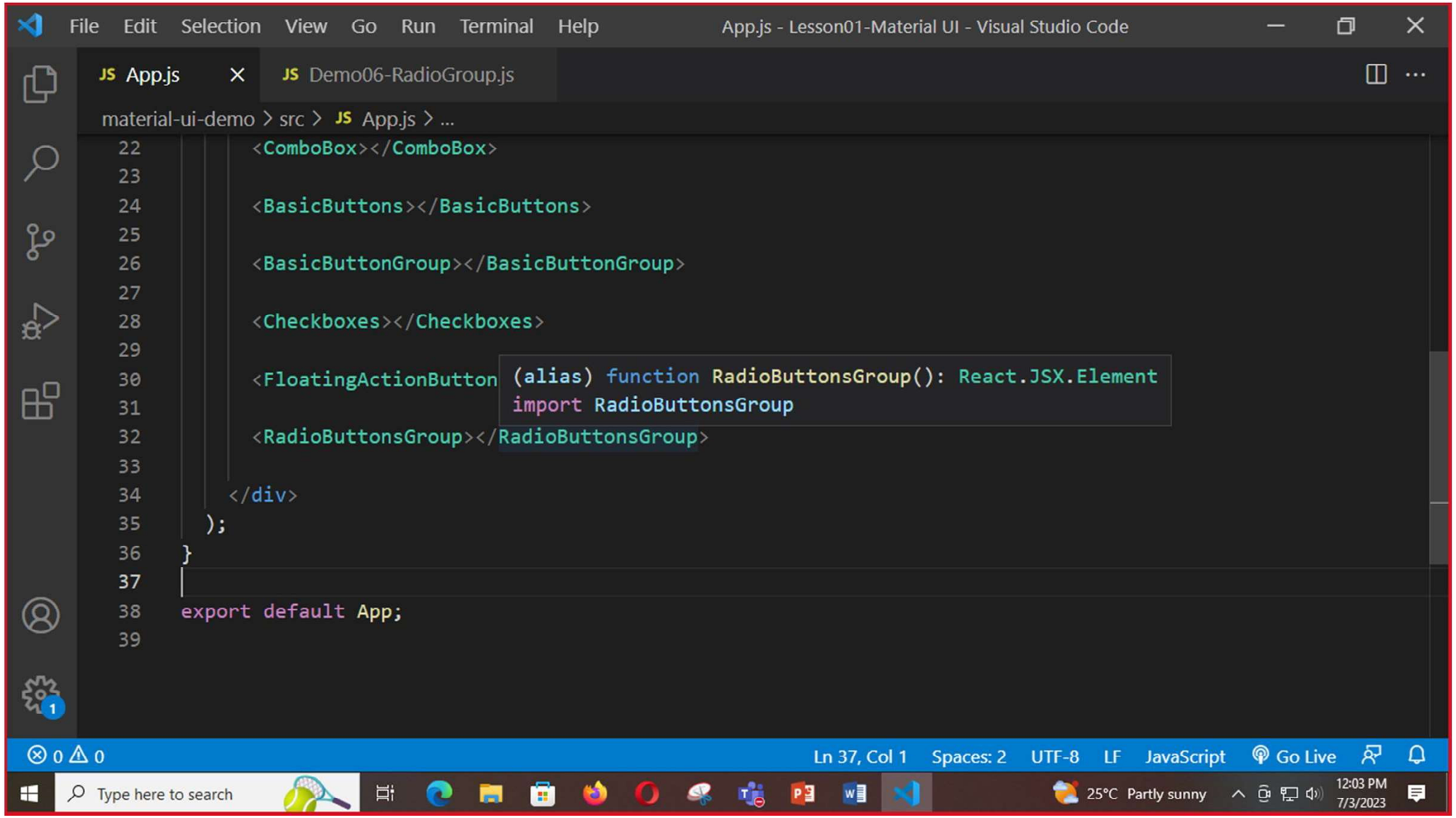
Lets See How to Add Material to React Application





Input Material UI Comps


File Edit Selection View Go Run Terminal Help
App.js - Lesson01-Material UI - Visual Studio Code
App.js
Demo06-RadioGroup.js
material-ui-demo > src > 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
14 {/*
15 <input type="button" value="Hello World" />
16
17
18
19 <Button variant="contained">Hello World </Button>
20 */}
21
⊗ 0 ⚠ 0
Ln 37, Col 1 Spaces: 2 UTF-8 LF JavaScript Go Live
Type here to search
25°C Partly sunny
12:03 PM
7/3/2023

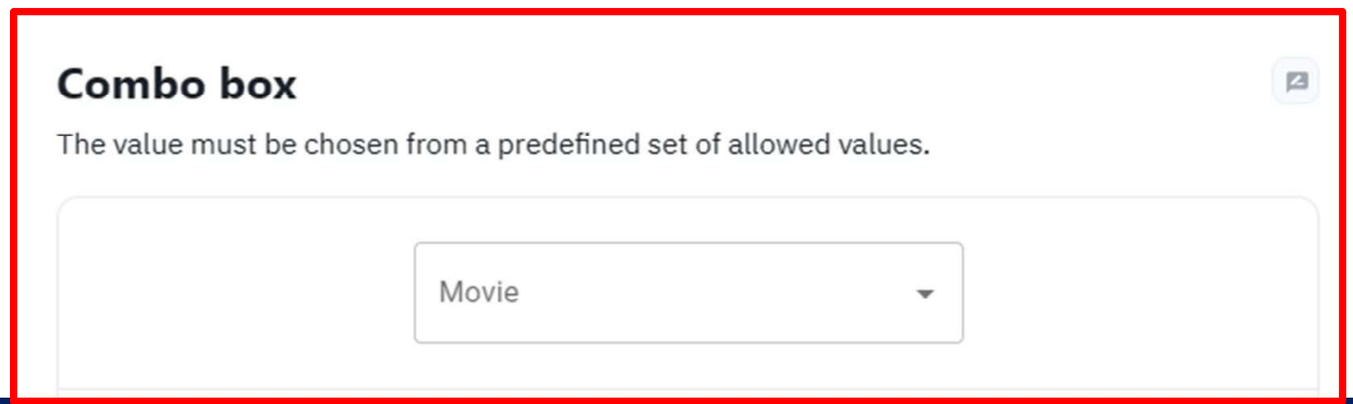

File Edit Selection View Go Run Terminal Help
App.js - Lesson01-Material UI - Visual Studio Code
App.js
Demo06-RadioGroup.js
material-ui-demo > src > App.js > ...
<ComboBox></ComboBox>
<BasicButtons></BasicButtons>
<BasicButtonGroup></BasicButtonGroup>
<Checkboxes></Checkboxes>
<FloatingActionButton
(alias) function RadioButtonsGroup(): React.JSX.Element
import RadioButtonsGroup
<RadioButtonsGroup></RadioButtonsGroup>

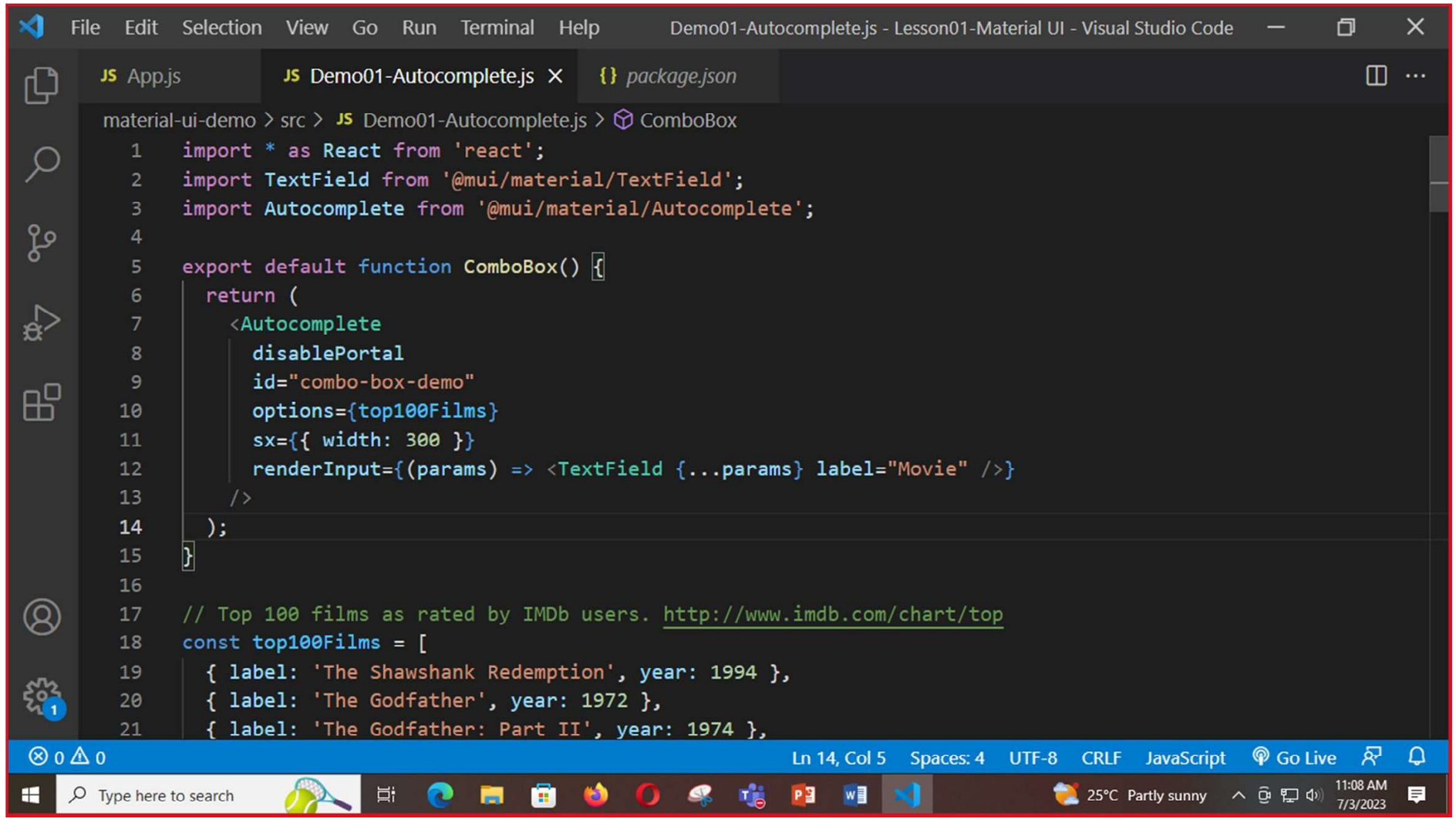
);
}
export default App;
Ln 37, Col 1 Spaces: 2 UTF-8 LF JavaScript Go Live
Type here to search
25°C Partly sunny
12:03 PM
7/3/2023

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.

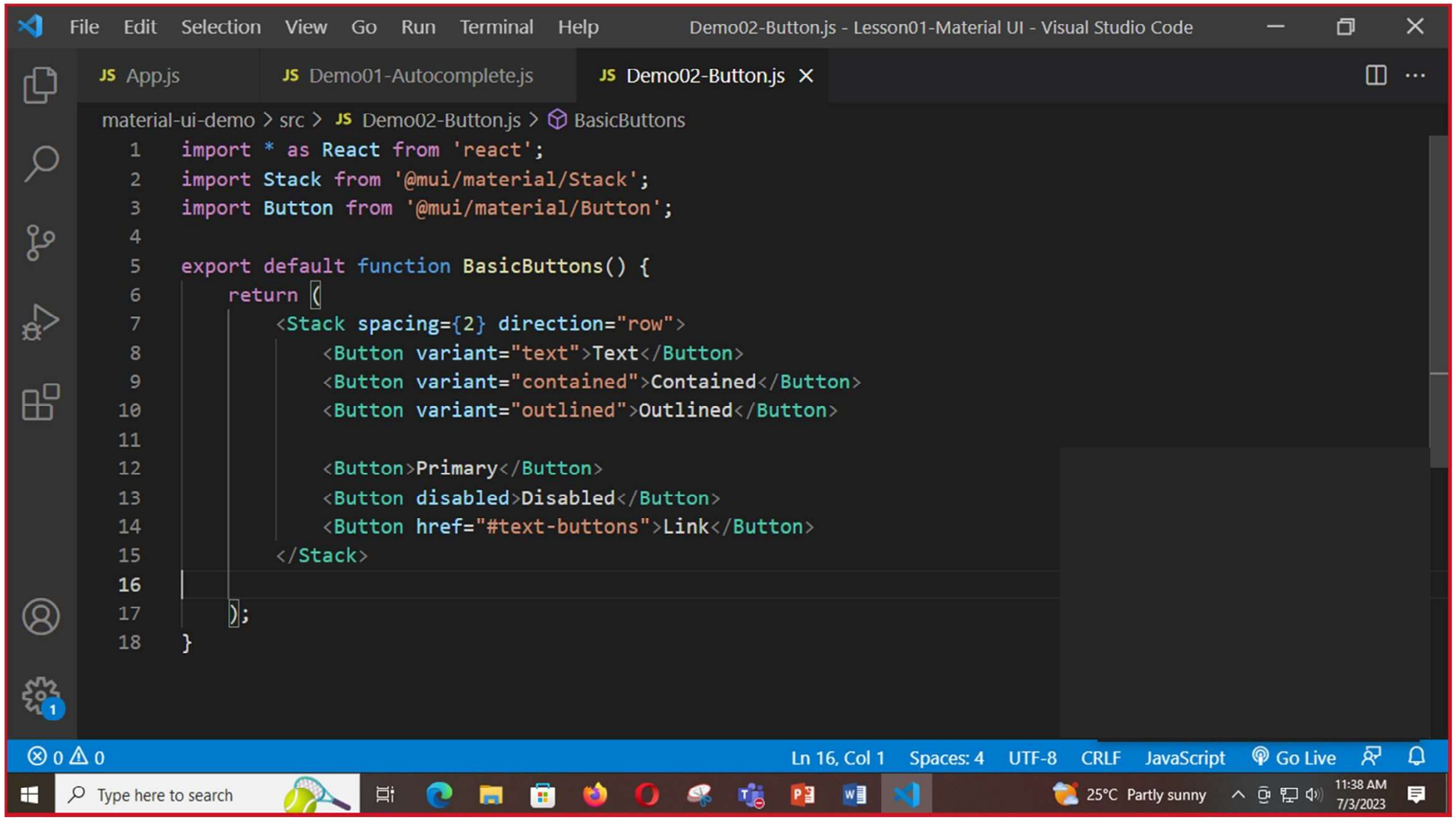



File Edit Selection View Go Run Terminal Help
Demo01-Autocomplete.js - Lesson01-Material UI - Visual Studio Code
App.js
Demo01-Autocomplete.js
package.json
material-ui-demo > src > Demo01-Autocomplete.js > ComboBox
1 import * as React from 'react';
2 import TextField from '@mui/material/TextField';
3 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 },
0 0
Ln 14, Col 5 Spaces: 4 UTF-8 CRLF JavaScript Go Live
Type here to search
25°C Partly sunny
11:08 AM
7/3/2023

Button

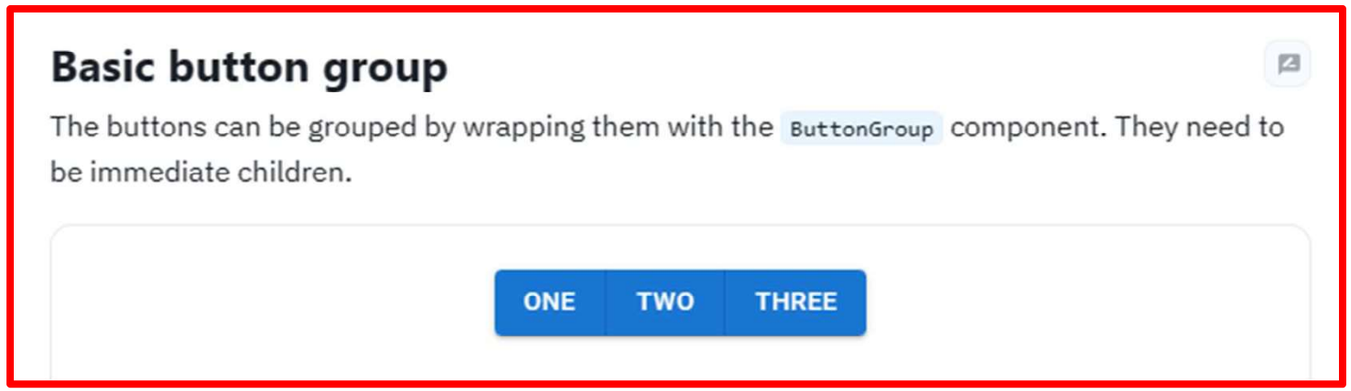
- Buttons allow users to take actions, and make choices, with a single tap.

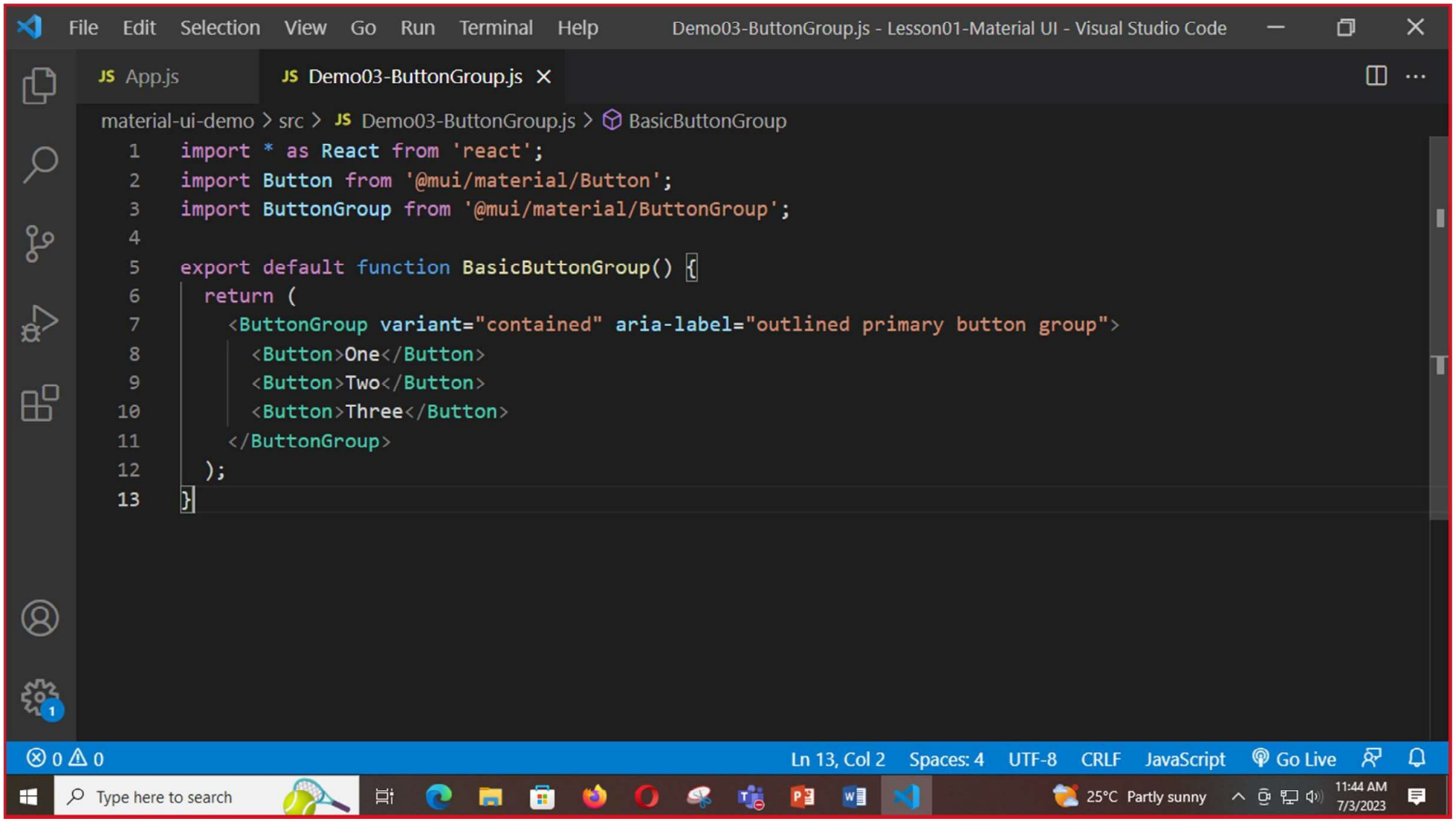
- Buttons communicate actions that users can take. They are typically placed throughout your UI, in places like:
  - Modal windows
  - Forms
  - Cards
  - Toolbars


File Edit Selection View Go Run Terminal Help
Demo02-Button.js - Lesson01-Material UI - Visual Studio Code
App.js
Demo01-Autocomplete.js
Demo02-Button.js
material-ui-demo > src > Demo02-Button.js > BasicButtons
import * as React from 'react';
import Stack from '@mui/material/Stack';
import Button from '@mui/material/Button';
export default function BasicButtons() {
return (
<Stack spacing={2} direction="row">
<Button variant="text">Text</Button>
<Button variant="contained">Contained</Button>
<Button variant="outlined">Outlined</Button>
<Button>Primary</Button>
<Button disabled>Disabled</Button>
<Button href="#text-buttons">Link</Button>
</Stack>
);
}
Ln 16, Col 1 Spaces: 4 UTF-8 CRLF JavaScript Go Live
Type here to search
25°C Partly sunny
11:38 AM
7/3/2023

ButtonGroup

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

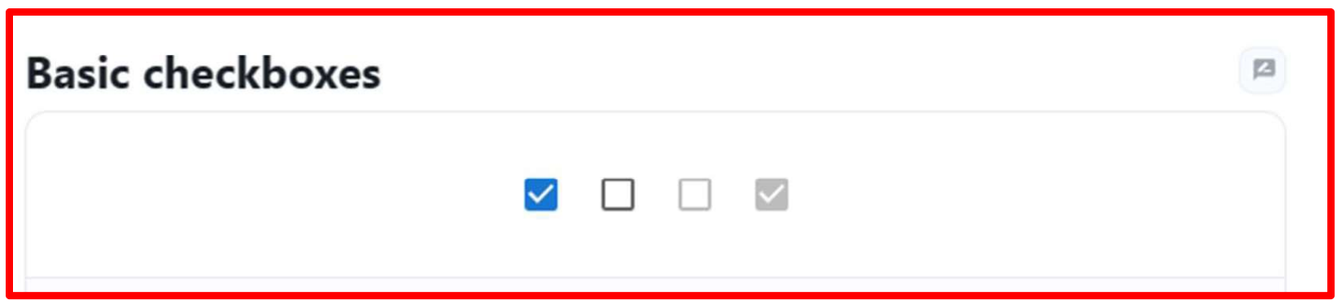


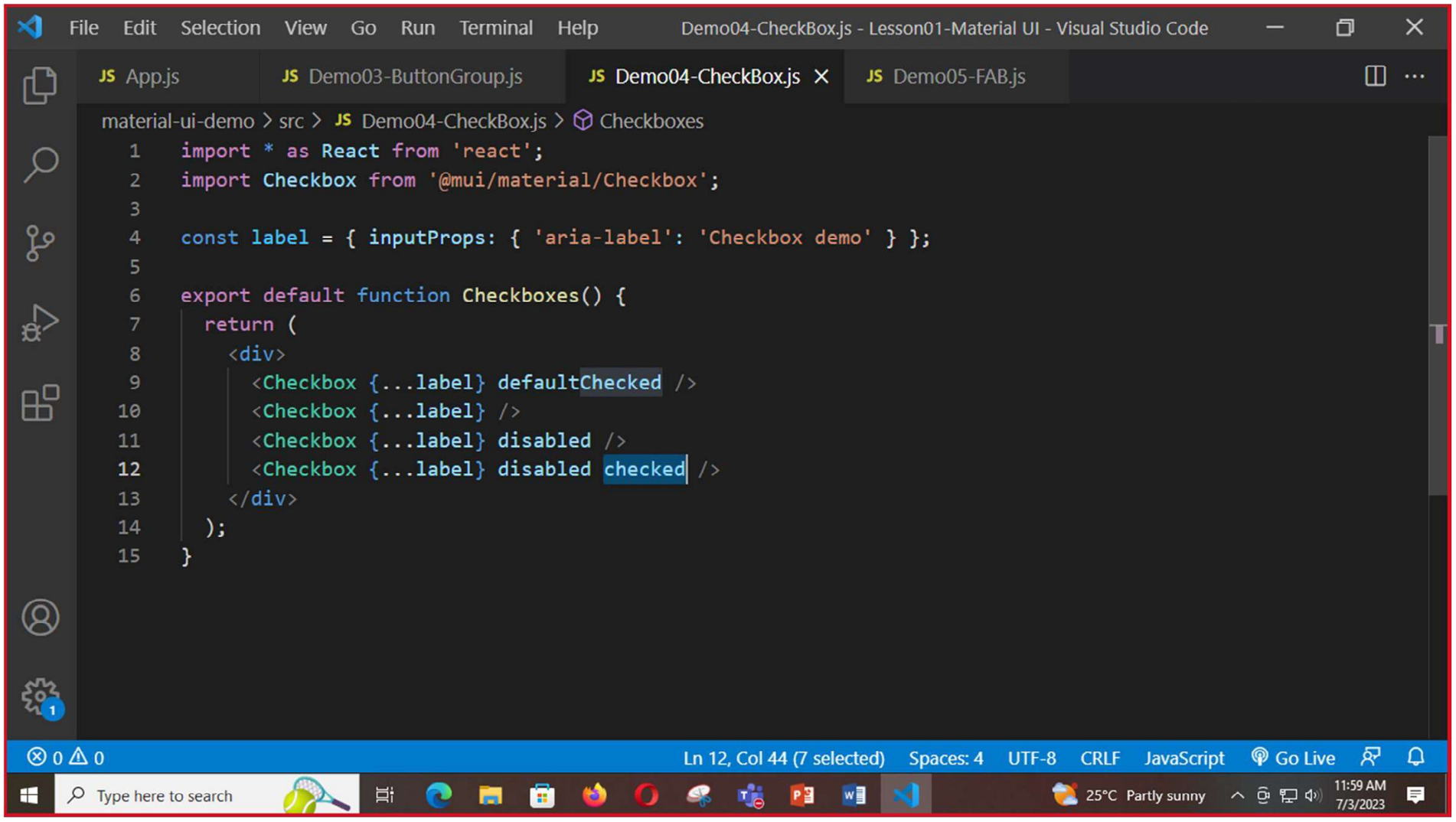

File Edit Selection View Go Run Terminal Help
Demo03-ButtonGroup.js - Lesson01-Material UI - Visual Studio Code
App.js
Demo03-ButtonGroup.js
material-ui-demo > src > Demo03-ButtonGroup.js > BasicButtonGroup
1 import * as React from 'react';
2 import Button from '@mui/material/Button';
3 import ButtonGroup 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 }
0 0
Ln 13, Col 2
Spaces: 4
UTF-8
CRLF
JavaScript
Go Live
Type here to search
25°C Partly sunny
11:44 AM
7/3/2023

Checkbox

- Checkboxes allow the user to select one or more items from a set.

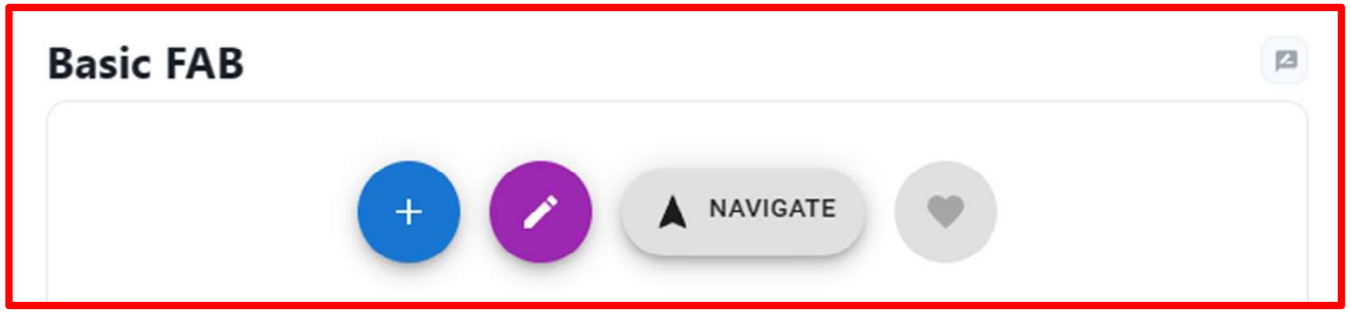
- Checkboxes can be used to turn an option on or off.
- If you have multiple options appearing in a list, you can preserve space by using checkboxes instead of on/off switches. If you have a single option, avoid using a checkbox and use an on/off switch instead.

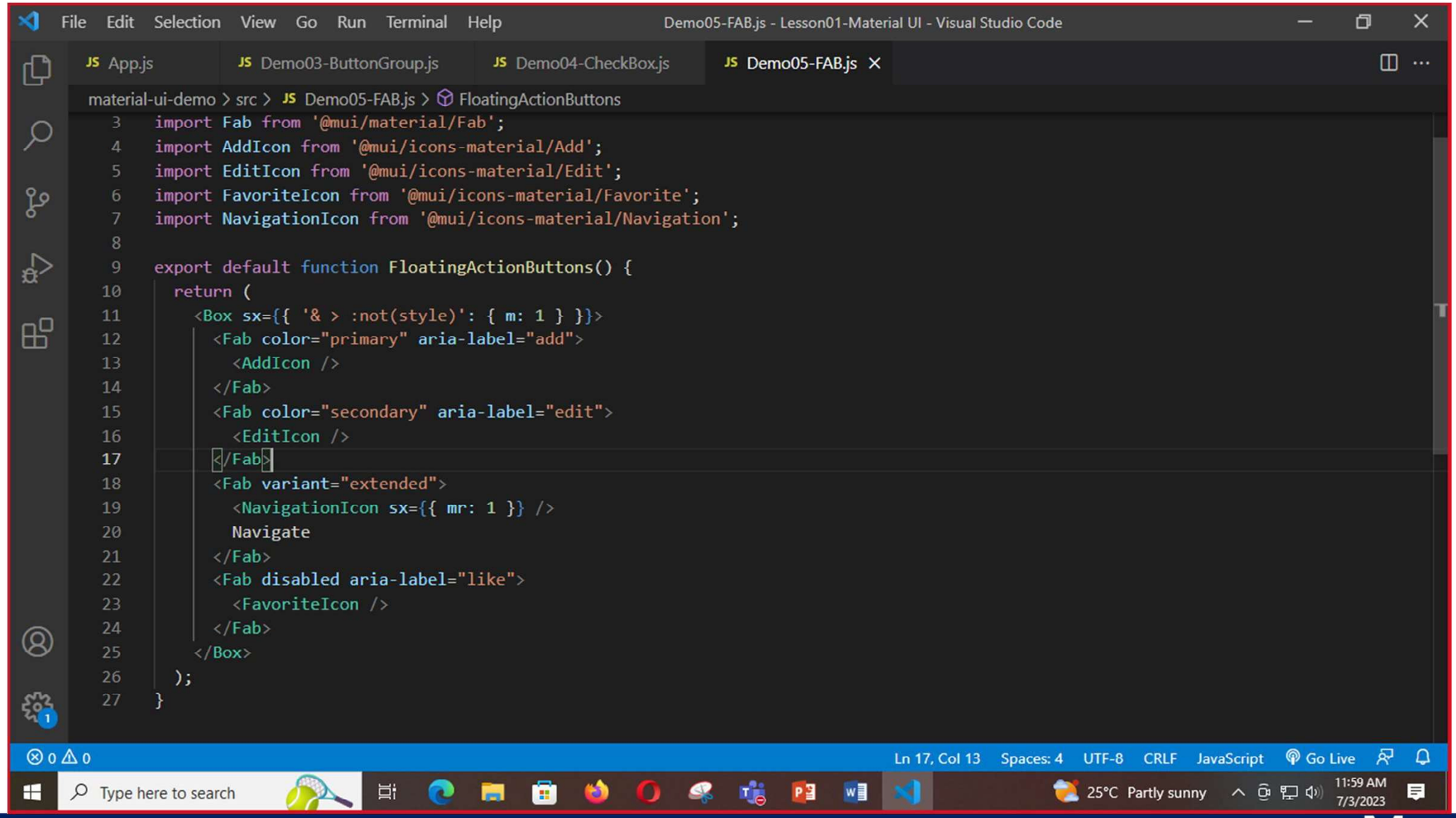



File Edit Selection View Go Run Terminal Help
Demo04-CheckBox.js - Lesson01-Material UI - Visual Studio Code
App.js
Demo03-ButtonGroup.js
Demo04-CheckBox.js
Demo05-FAB.js
material-ui-demo > src > Demo04-CheckBox.js > Checkboxes
1 import * as React from 'react';
2 import Checkbox from '@mui/material/Checkbox';
3
4 const label = { inputProps: { 'aria-label': 'Checkbox demo' } };
5
6 export default function Checkboxes() {
7 return (
8
9 <Checkbox {...label} defaultChecked />
10 <Checkbox {...label} />
11 <Checkbox {...label} disabled />
12 <Checkbox {...label} disabled checked />
13
14);
15 }
0 0
Ln 12, Col 44 (7 selected)
Spaces: 4
UTF-8
CRLF
JavaScript
Go Live
Type here to search
25°C Partly sunny
11:59 AM
7/3/2023

Floating Action Button

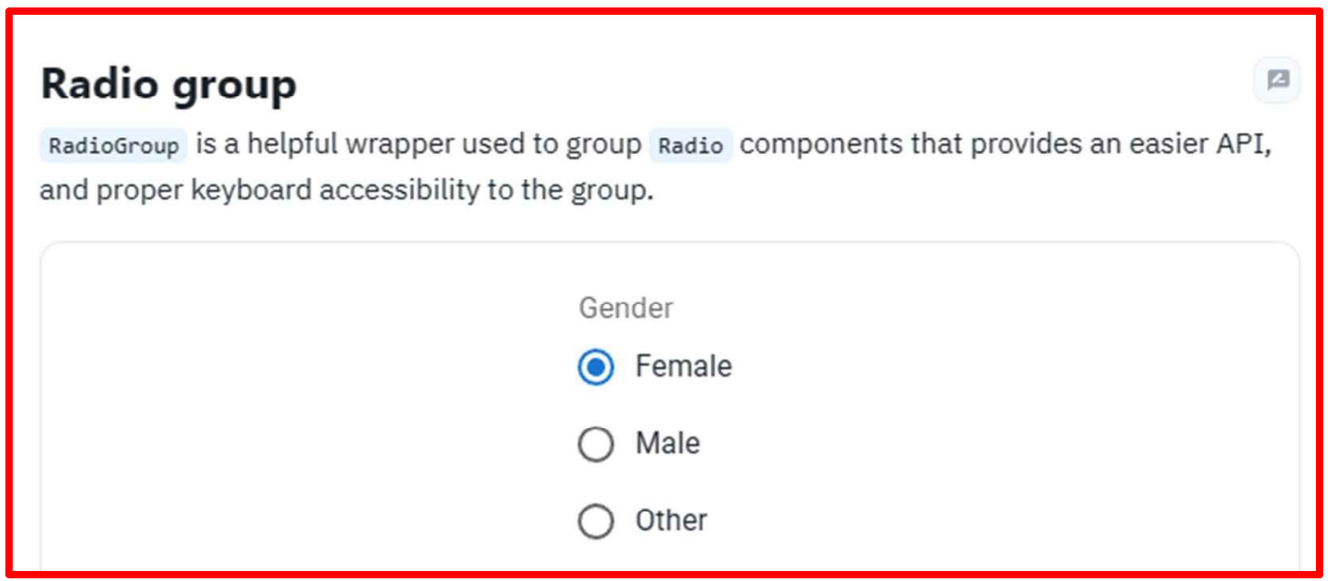
- A Floating Action Button (FAB) performs the primary, or most common, action on a screen.
- A floating action button appears in front of all screen content, typically as a circular shape with an icon in its center. FABs come in two types: regular, and extended.
- Only use a FAB if it is the most suitable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

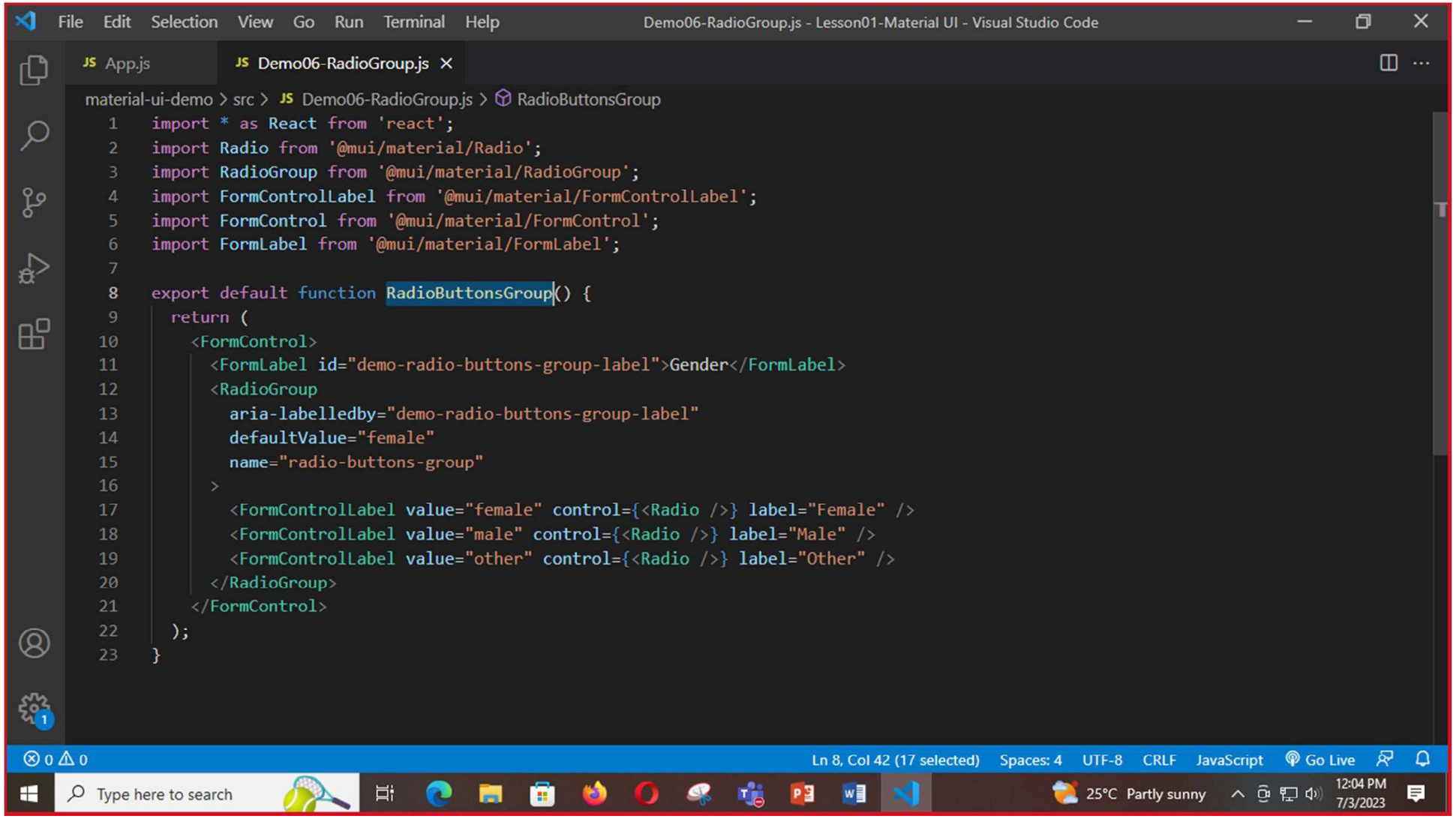



File Edit Selection View Go Run Terminal Help
Demo05-FAB.js - Lesson01-Material UI - Visual Studio Code
App.js
Demo03-ButtonGroup.js
Demo04-CheckBox.js
Demo05-FAB.js
material-ui-demo > src > Demo05-FAB.js > FloatingActionButtons
import Fab from '@mui/material/Fab';
import AddIcon from '@mui/icons-material/Add';
import EditIcon from '@mui/icons-material/Edit';
import FavoriteIcon from '@mui/icons-material/Favorite';
import NavigationIcon from '@mui/icons-material/Navigation';
export default function FloatingActionButtons() {
return (
<Box sx={{ '& > :not(style)': { m: 1 } }}>
<Fab color="primary" aria-label="add">
<AddIcon />
</Fab>
<Fab color="secondary" aria-label="edit">
<EditIcon />
</Fab>
<Fab variant="extended">
<NavigationIcon sx={{ mr: 1 }} />
Navigate
</Fab>
<Fab disabled aria-label="like">
<FavoriteIcon />
</Fab>
</Box>
);
}
Ln 17, Col 13
Spaces: 4
UTF-8
CRLF
JavaScript
Go Live
Type here to search
25°C Partly sunny
11:59 AM
7/3/2023

Radio Group

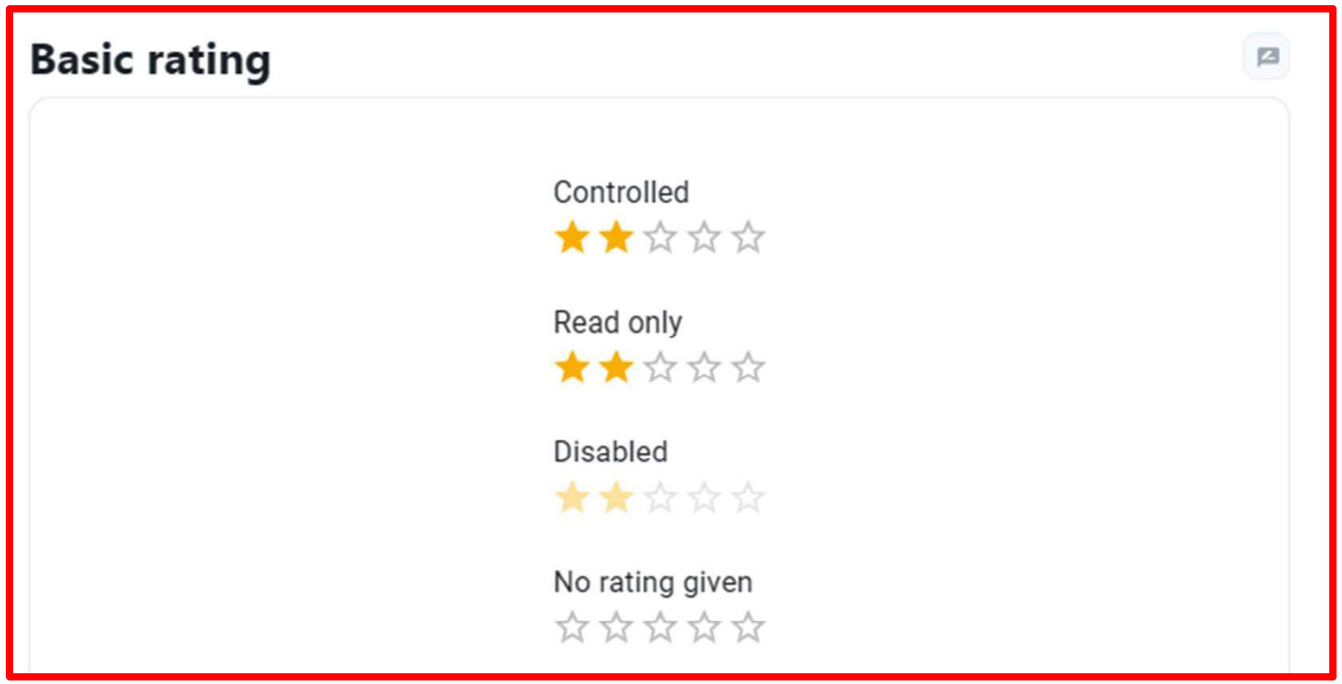
- The Radio Group allows the user to select one option from a set.
- Use radio buttons when the user needs to see all available options. If available options can be collapsed, consider using a Select component because it uses less space.
- Radio buttons should have the most commonly used option selected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

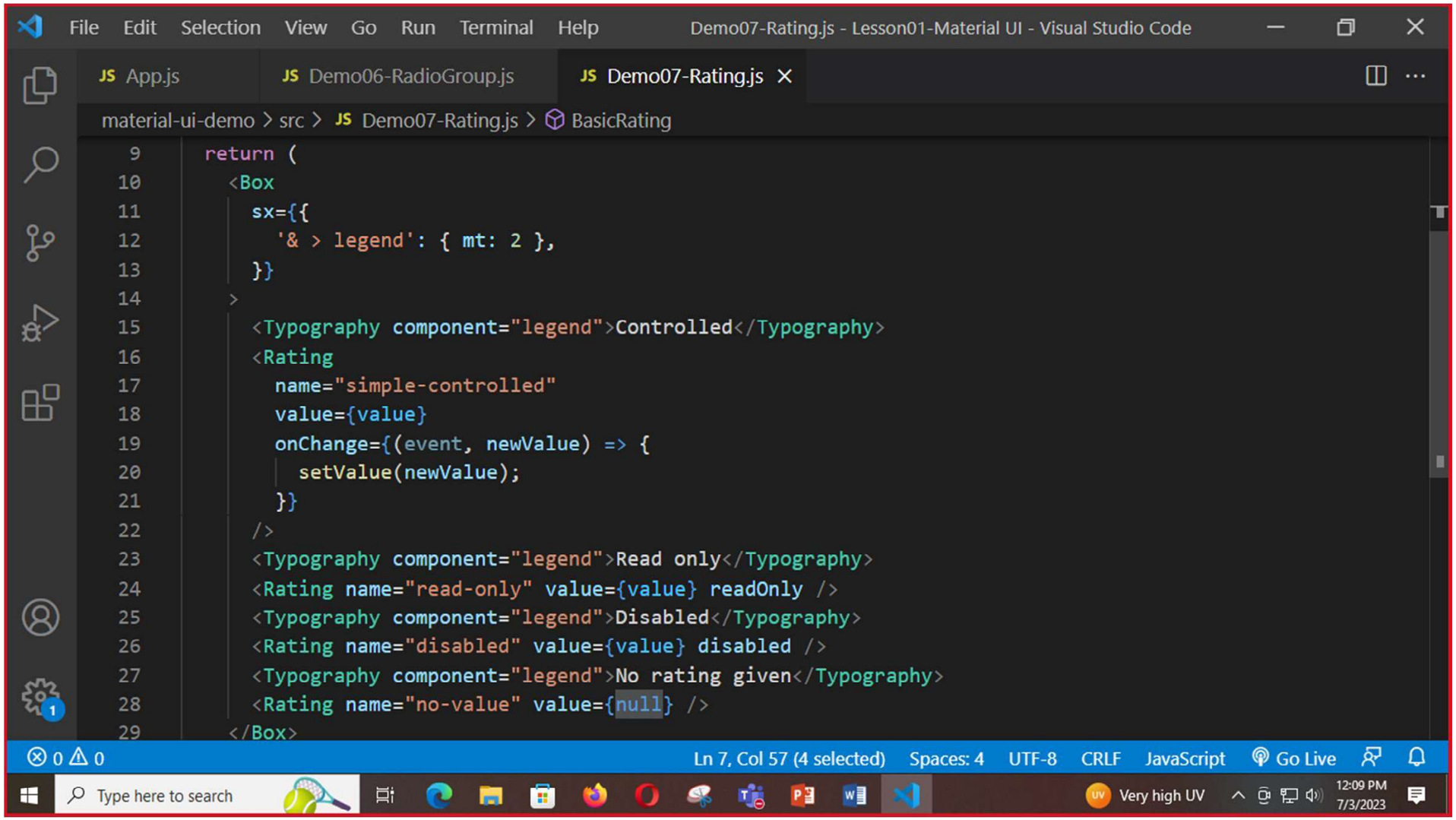



File Edit Selection View Go Run Terminal Help
Demo06-RadioGroup.js - Lesson01-Material UI - Visual Studio Code
App.js
Demo06-RadioGroup.js
material-ui-demo > src > Demo06-RadioGroup.js > RadioButtonsGroup
1 import * as React from 'react';
2 import Radio from '@mui/material/Radio';
3 import RadioGroup 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 }
Ln 8, Col 42 (17 selected) Spaces: 4 UTF-8 CRLF JavaScript Go Live
Type here to search
25°C Partly sunny
12:04 PM
7/3/2023

Rating

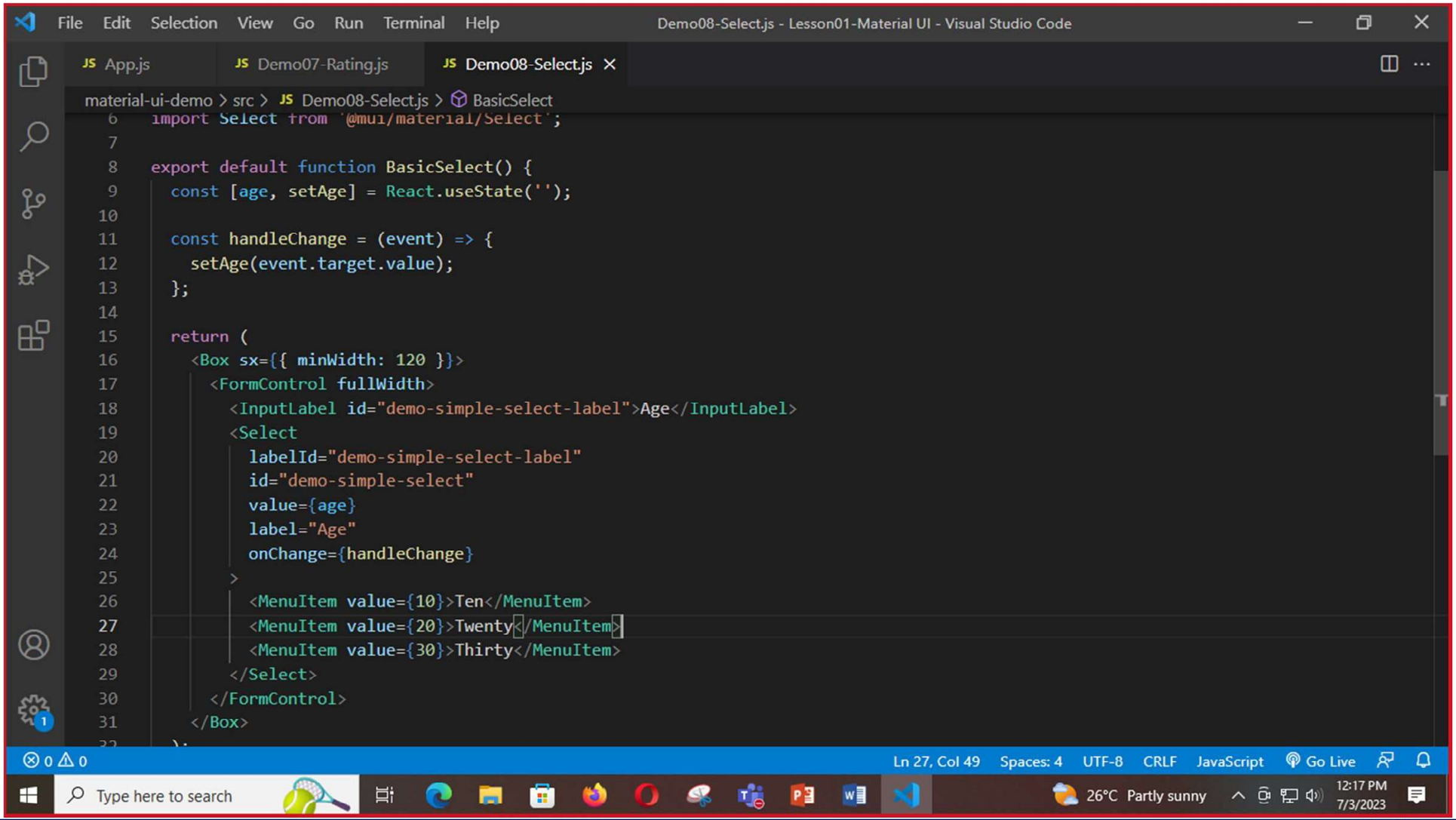
- Ratings provide insight regarding others' opinions and experiences, and can allow the user to submit a rating of their own.




File Edit Selection View Go Run Terminal Help
Demo07-Rating.js - Lesson01-Material UI - Visual Studio Code
App.js
Demo06-RadioGroup.js
Demo07-Rating.js
material-ui-demo > src > Demo07-Rating.js > BasicRating
return (
<Box
sx={{
'& > legend': { mt: 2 },
}}
>
<Typography component="legend">Controlled</Typography>
<Rating
name="simple-controlled"
value={value}
onChange={(event, newValue) => {
setValue(newValue);
}}
/>
<Typography component="legend">Read only</Typography>
<Rating name="read-only" value={value} readOnly />
<Typography component="legend">Disabled</Typography>
<Rating name="disabled" value={value} disabled />
<Typography component="legend">No rating given</Typography>
<Rating name="no-value" value={null} />
</Box>
Ln 7, Col 57 (4 selected) Spaces: 4 UTF-8 CRLF JavaScript Go Live
Type here to search
Very high UV
12:09 PM
7/3/2023

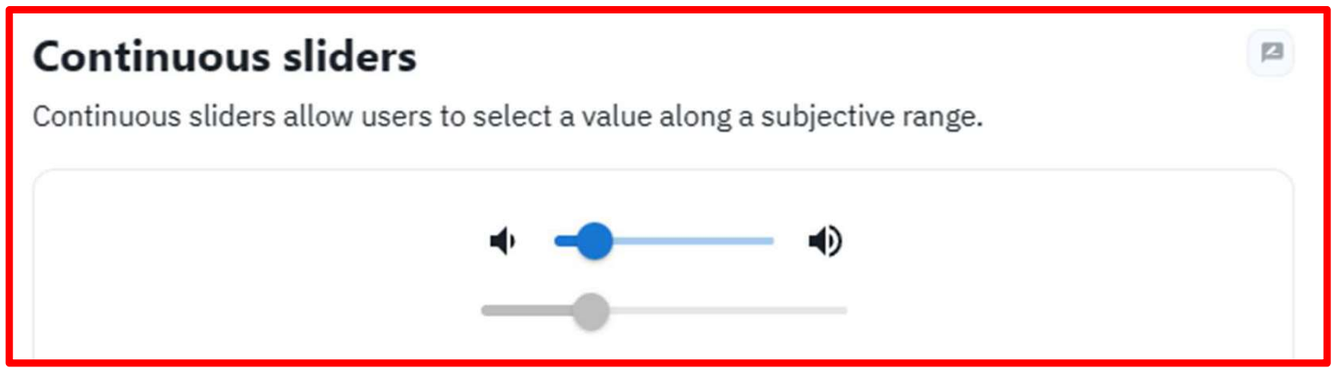
Select

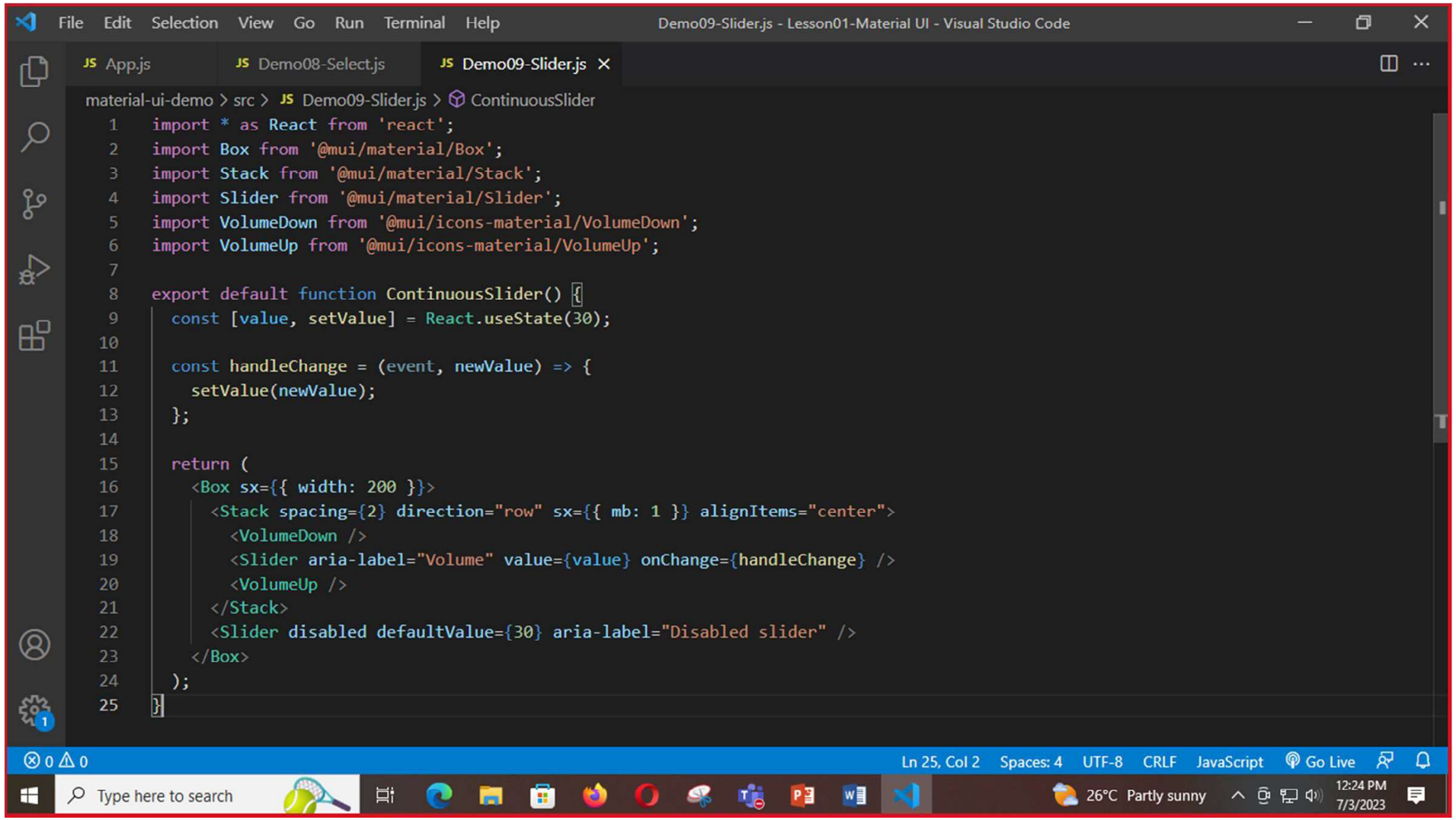
- Select components are used for collecting user provided information from a list of options.


File Edit Selection View Go Run Terminal Help
Demo08-Select.js - Lesson01-Material UI - Visual Studio Code
App.js
Demo07-Rating.js
Demo08-Select.js
material-ui-demo > src > Demo08-Select.js > BasicSelect
import Select from '@mui/material/Select';
export default function BasicSelect() {
const [age, setAge] = React.useState('');
const handleChange = (event) => {
setAge(event.target.value);
};
return (
<Box sx={{ minWidth: 120 }}>
<FormControl fullWidth>
<InputLabel id="demo-simple-select-label">Age</InputLabel>
<Select
labelId="demo-simple-select-label"
id="demo-simple-select"
value={age}
label="Age"
onChange={handleChange}
>
<MenuItem value={10}>Ten</MenuItem>
<MenuItem value={20}>Twenty</MenuItem>
<MenuItem value={30}>Thirty</MenuItem>
</Select>
</FormControl>
</Box>
Ln 27, Col 49 Spaces: 4 UTF-8 CRLF JavaScript Go Live
Type here to search
26°C Partly sunny
12:17 PM
7/3/2023

Slider

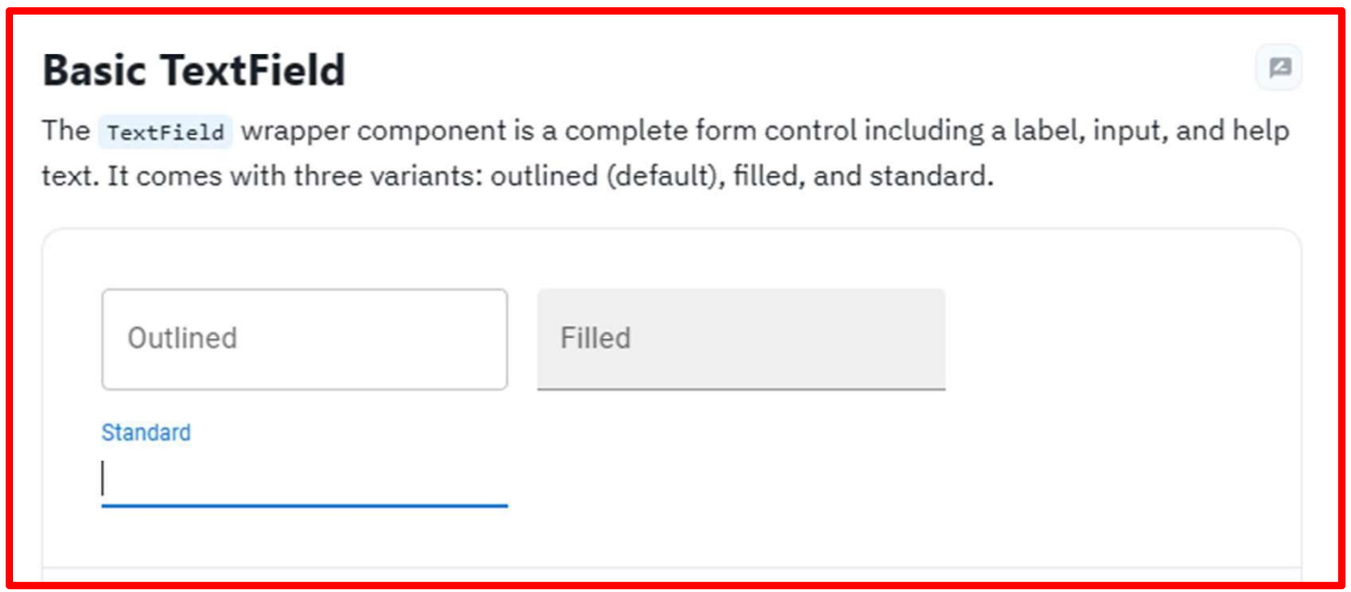
- Sliders allow users to make selections from a range of values.
- Sliders reflect a range of values along a bar, from which users may select a single value. They are ideal for adjusting settings such as volume, brightness, or applying image filters.

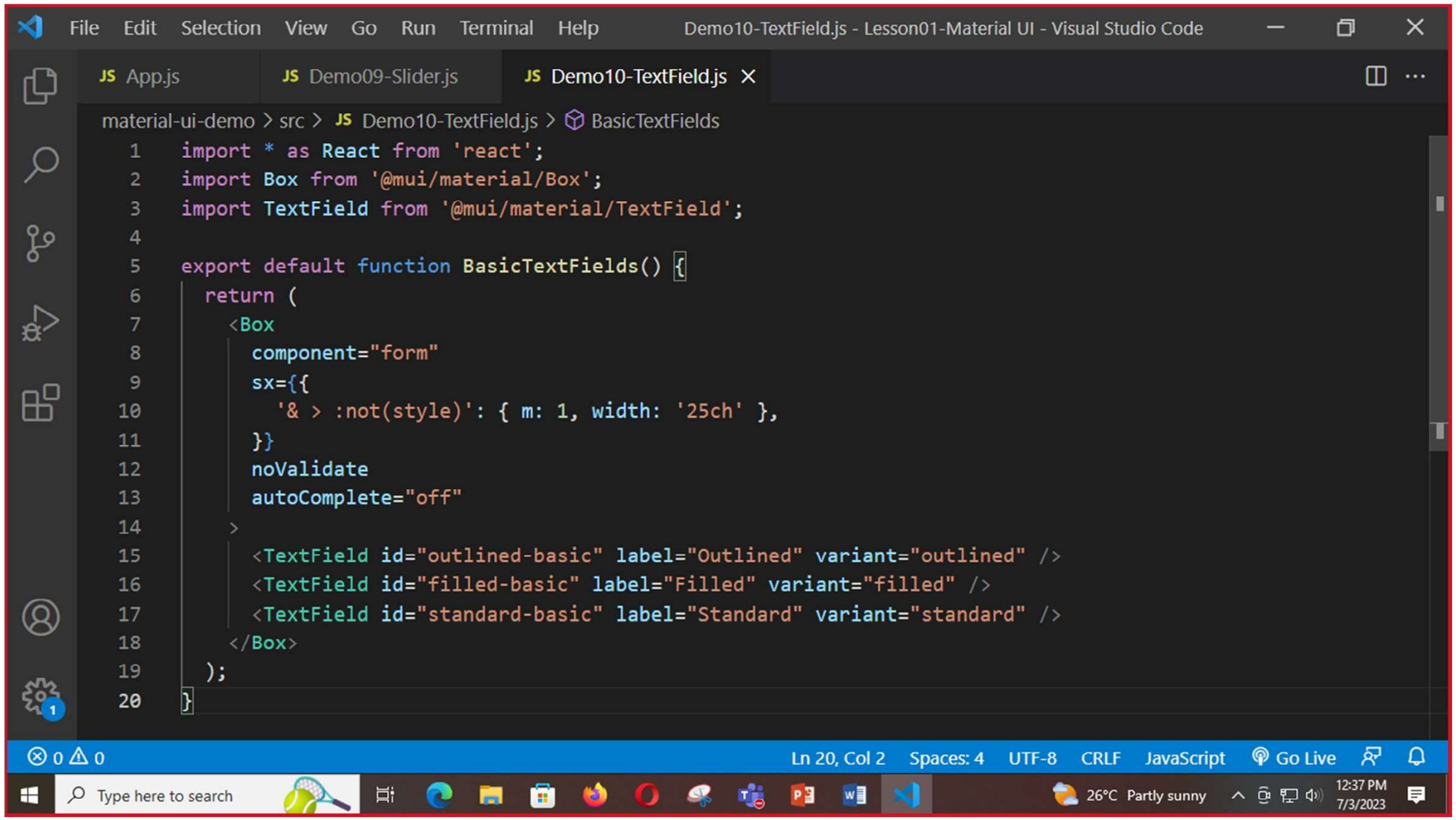



File Edit Selection View Go Run Terminal Help
Demo09-Slider.js - Lesson01-Material UI - Visual Studio Code
App.js
Demo08-Select.js
Demo09-Slider.js
material-ui-demo > src > Demo09-Slider.js > ContinuousSlider
1 import * as React from 'react';
2 import Box from '@mui/material/Box';
3 import Stack from '@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
Type here to search
26°C Partly sunny
12:24 PM
7/3/2023

TextField

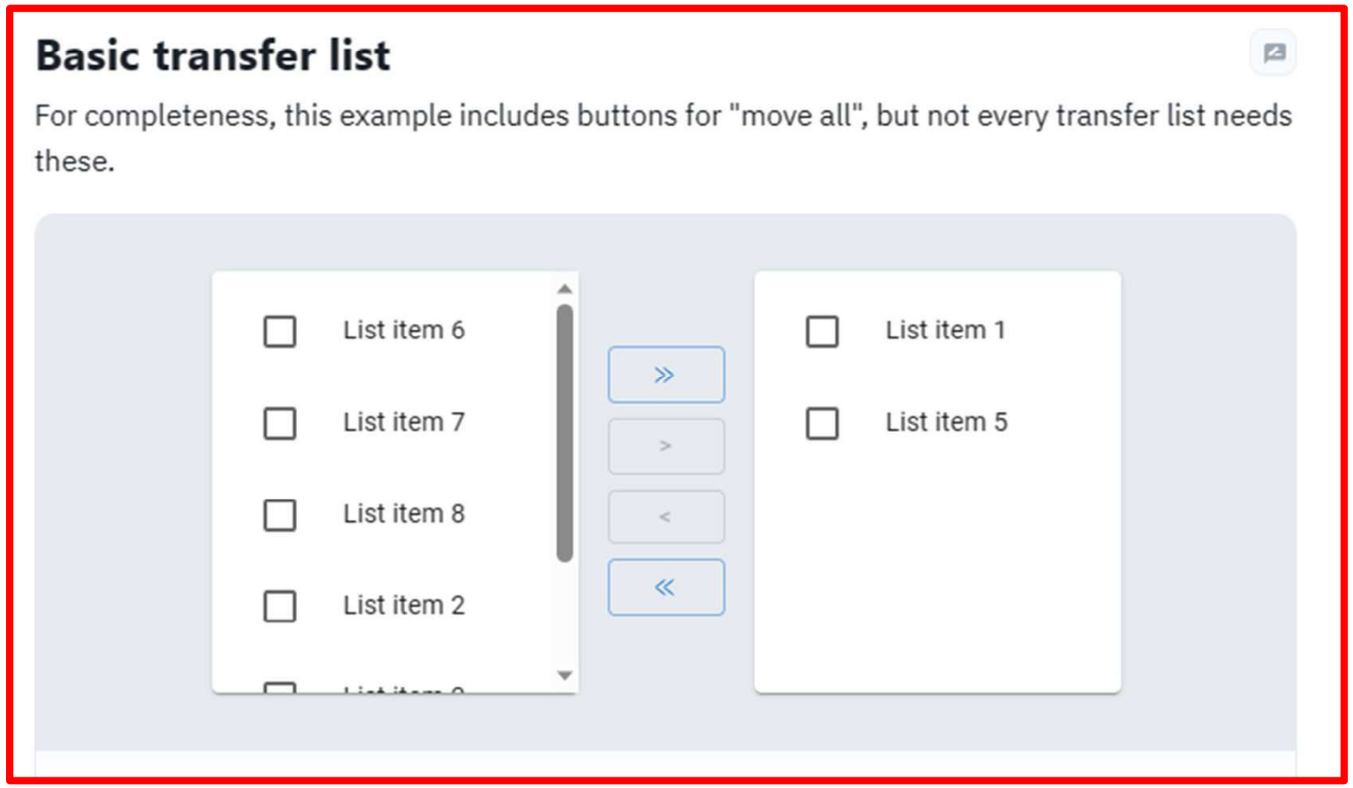
- Text Fields let users enter and edit text.
- Text fields allow users to enter text into a UI. They typically appear in forms and dialogs.

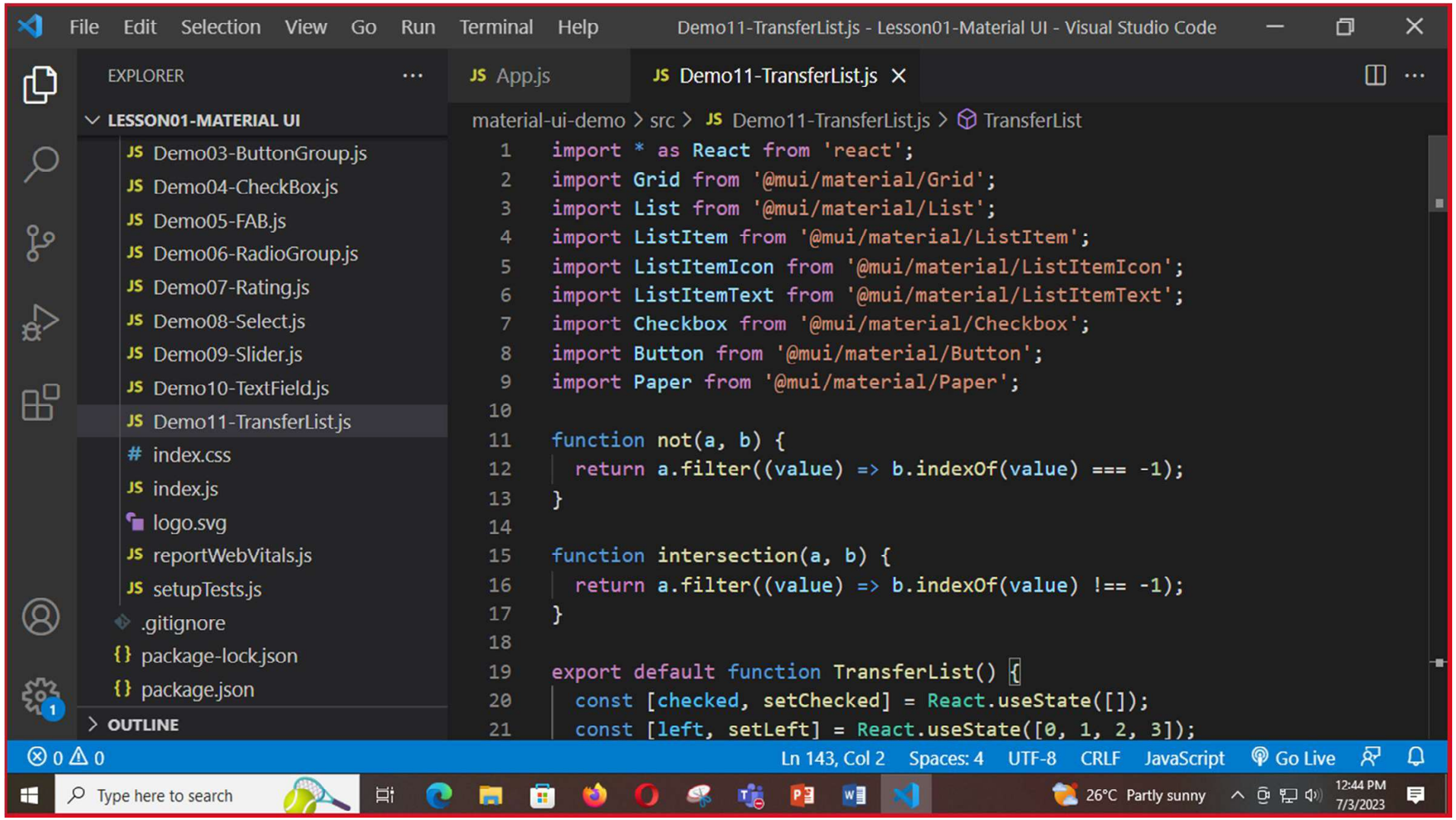


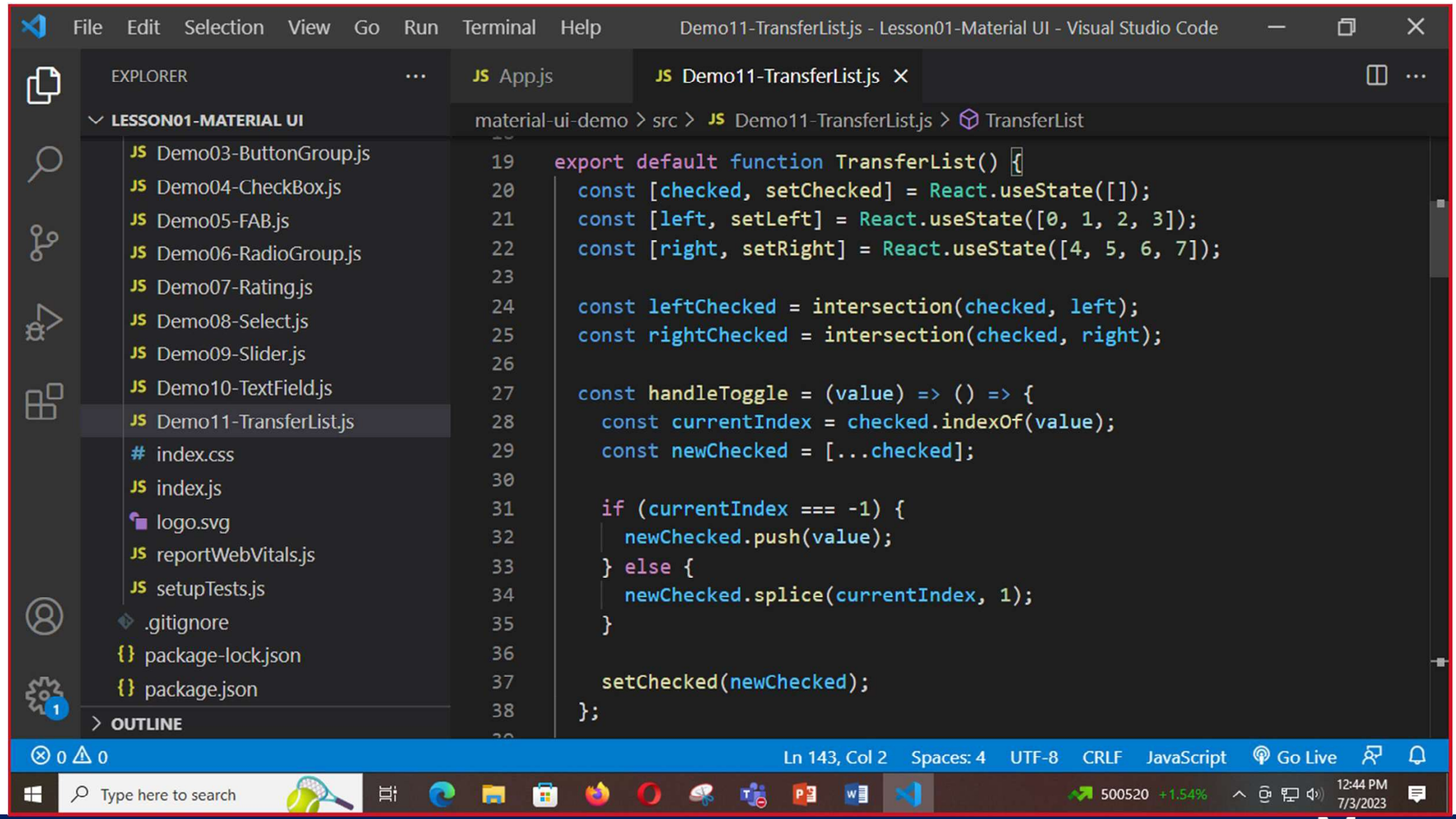

File Edit Selection View Go Run Terminal Help
Demo10-TextField.js - Lesson01-Material UI - Visual Studio Code
App.js
Demo09-Slider.js
Demo10-TextField.js
material-ui-demo > src > Demo10-TextField.js > BasicTextFields
1 import * as React from 'react';
2 import Box from '@mui/material/Box';
3 import TextField 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 }
0 0
Ln 20, Col 2 Spaces: 4 UTF-8 CRLF JavaScript Go Live
Type here to search
26°C Partly sunny
12:37 PM
7/3/2023

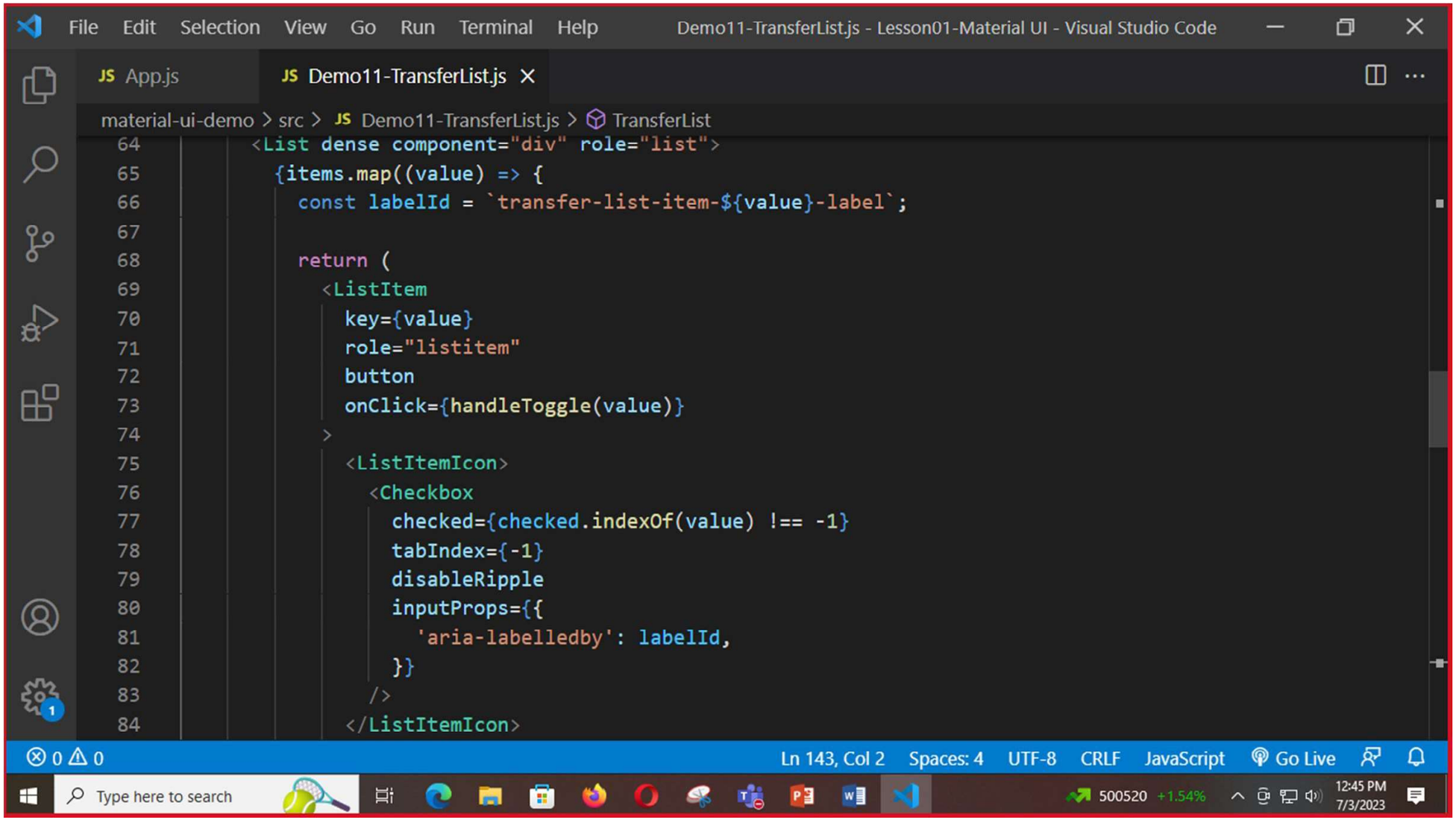
Transfer List / Shuttle

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



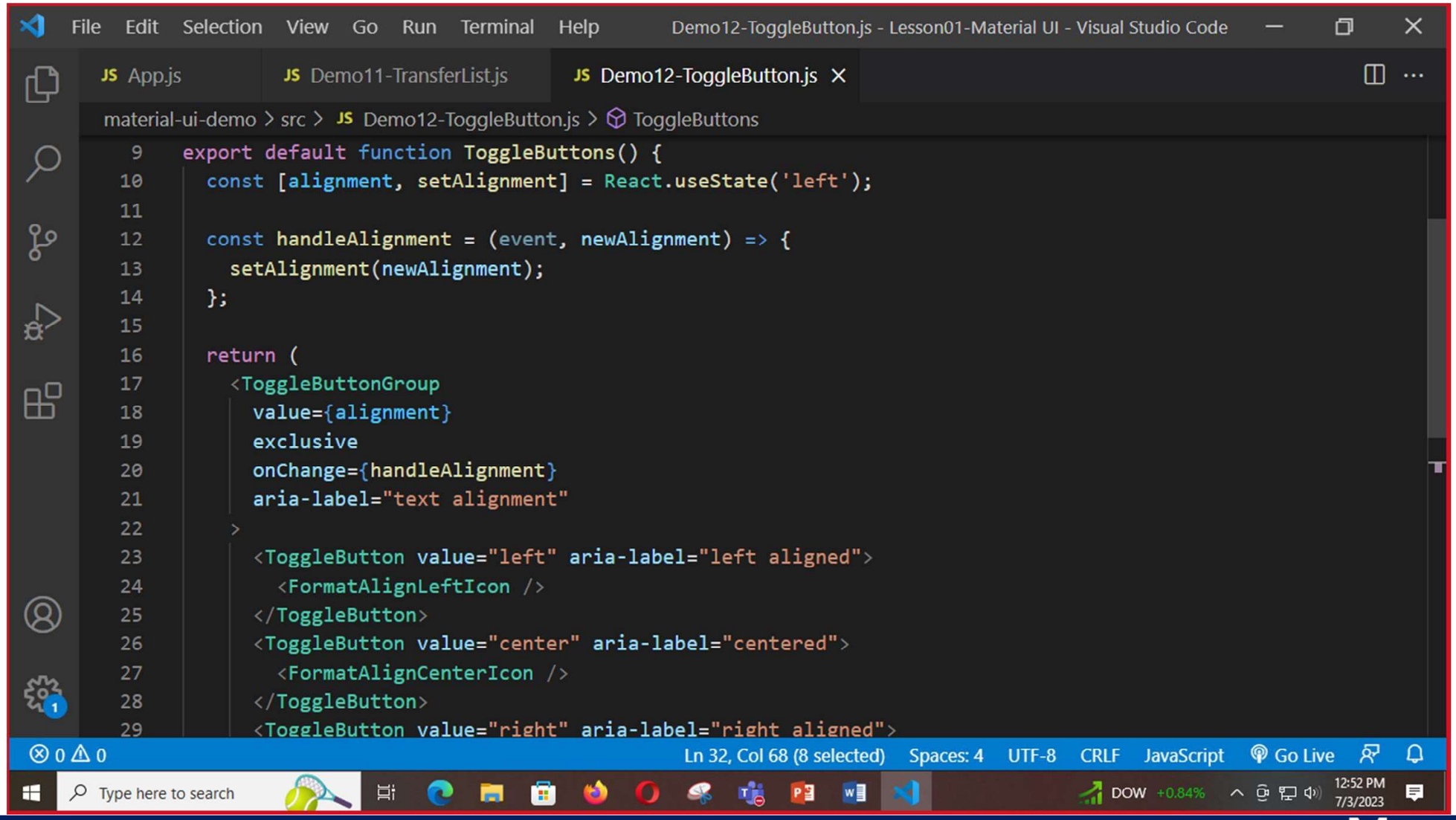

File Edit Selection View Go Run Terminal Help
Demo11-TransferList.js - Lesson01-Material UI - Visual Studio Code
EXPLORER
LESSON01-MATERIAL UI
Demo03-ButtonGroup.js
Demo04-CheckBox.js
Demo05-FAB.js
Demo06-RadioGroup.js
Demo07-Rating.js
Demo08-Select.js
Demo09-Slider.js
Demo10-TextField.js
Demo11-TransferList.js
index.css
index.js
logo.svg
reportWebVitals.js
setupTests.js
.gitignore
package-lock.json
package.json
OUTLINE
App.js
Demo11-TransferList.js
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 import ListItemText 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
11 function not(a, b) {
12 return a.filter((value) => b.indexOf(value) === -1);
13 }
14
15 function intersection(a, b) {
16 return a.filter((value) => b.indexOf(value) !== -1);
17 }
18
19 export default function TransferList() {
20 const [checked, setChecked] = React.useState([]);
21 const [left, setLeft] = React.useState([0, 1, 2, 3]);
0 0
Ln 143, Col 2 Spaces: 4 UTF-8 CRLF JavaScript Go Live
Type here to search
26°C Partly sunny
12:44 PM
7/3/2023

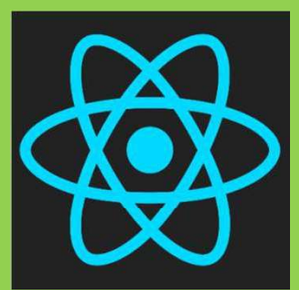

File Edit Selection View Go Run Terminal Help
Demo11-TransferList.js - Lesson01-Material UI - Visual Studio Code
EXPLORER
App.js
Demo11-TransferList.js
LESSON01-MATERIAL UI
material-ui-demo > src > Demo11-TransferList.js > TransferList
Demo03-ButtonGroup.js
Demo04-CheckBox.js
Demo05-FAB.js
Demo06-RadioGroup.js
Demo07-Rating.js
Demo08-Select.js
Demo09-Slider.js
Demo10-TextField.js
Demo11-TransferList.js
index.css
index.js
logo.svg
reportWebVitals.js
setupTests.js
.gitignore
package-lock.json
package.json
OUTLINE
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);
};
0 0
Ln 143, Col 2 Spaces: 4 UTF-8 CRLF JavaScript Go Live
Type here to search
500520 +1.54%
12:44 PM
7/3/2023


File Edit Selection View Go Run Terminal Help
Demo11-TransferList.js - Lesson01-Material UI - Visual Studio Code
App.js
Demo11-TransferList.js
material-ui-demo > src > Demo11-TransferList.js > TransferList
<List dense component="div" role="list">
{items.map((value) => {
const labelId = `transfer-list-item-${value}-label`;
return (
<ListItem
key={value}
role="listitem"
button
onClick={handleToggle(value)}
>
<ListItemIcon>
<Checkbox
checked={checked.indexOf(value) !== -1}
tabIndex={-1}
disableRipple
inputProps={{
'aria-labelledby': labelId,
}}
/>
</ListItemIcon>
Ln 143, Col 2
Spaces: 4
UTF-8
CRLF
JavaScript
Go Live
Type here to search
500520 +1.54%
12:45 PM
7/3/2023

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.

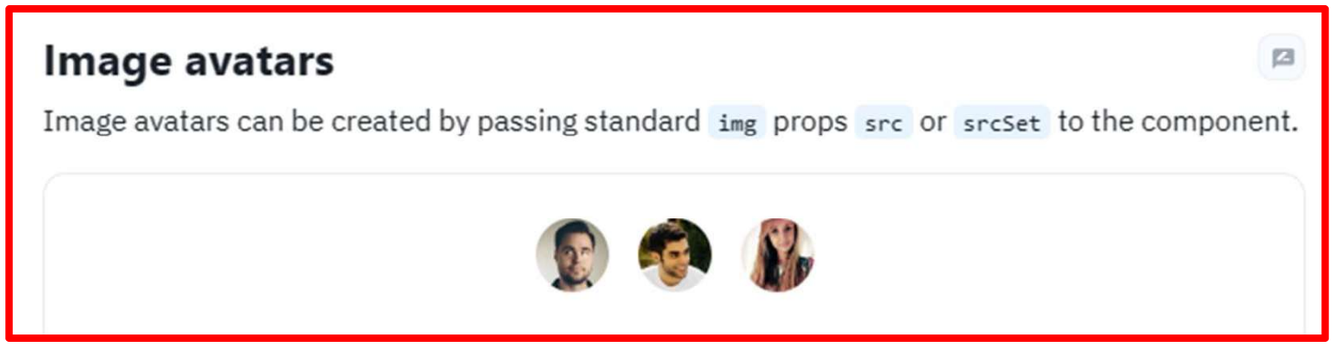

File Edit Selection View Go Run Terminal Help
Demo12-ToggleButton.js - Lesson01-Material UI - Visual Studio Code
App.js
Demo11-TransferList.js
Demo12-ToggleButton.js
material-ui-demo > src > Demo12-ToggleButton.js > ToggleButtons
export default function ToggleButtons() {
const [alignment, setAlignment] = React.useState('left');
const handleAlignment = (event, newAlignment) => {
setAlignment(newAlignment);
};
return (
<ToggleButtonGroup
value={alignment}
exclusive
onChange={handleAlignment}
aria-label="text alignment"
>
<ToggleButton value="left" aria-label="left aligned">
<FormatAlignLeftIcon />
</ToggleButton>
<ToggleButton value="center" aria-label="centered">
<FormatAlignCenterIcon />
</ToggleButton>
<ToggleButton value="right" aria-label="right aligned">
0 0
Ln 32, Col 68 (8 selected)
Spaces: 4
UTF-8
CRLF
JavaScript
Go Live
Type here to search
DOW +0.84%
12:52 PM
7/3/2023

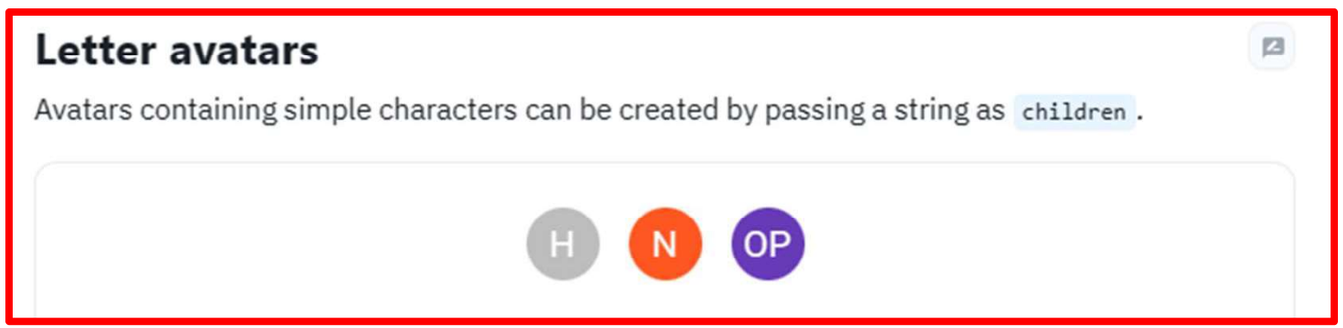


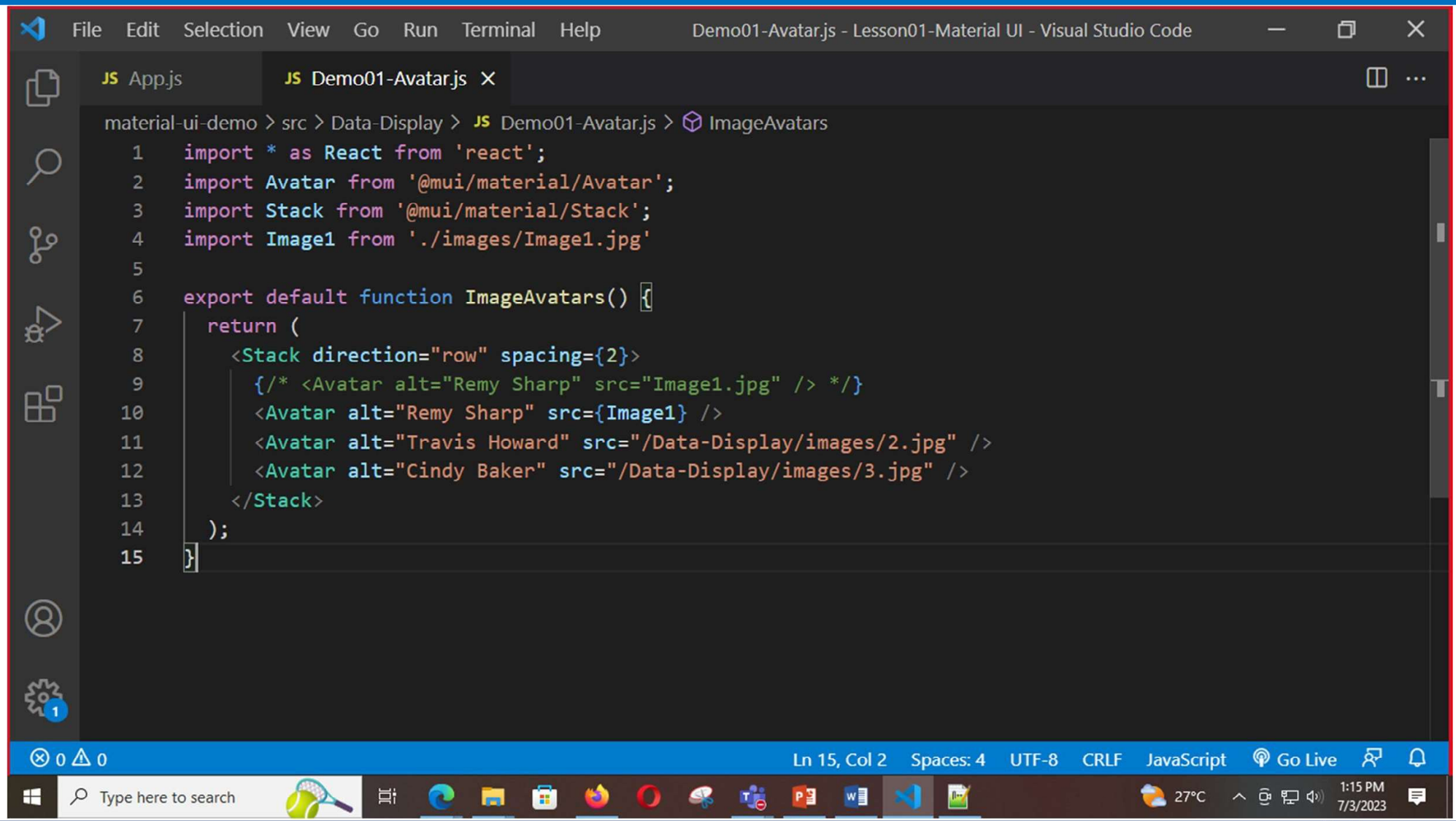
Data Display

Avatar

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

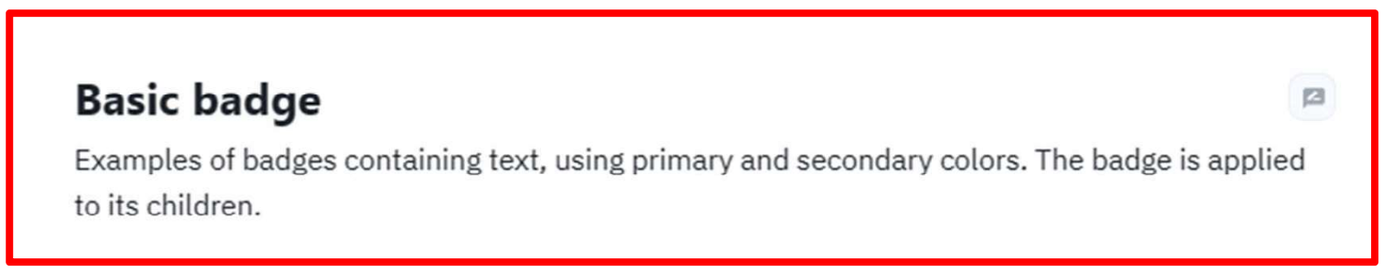


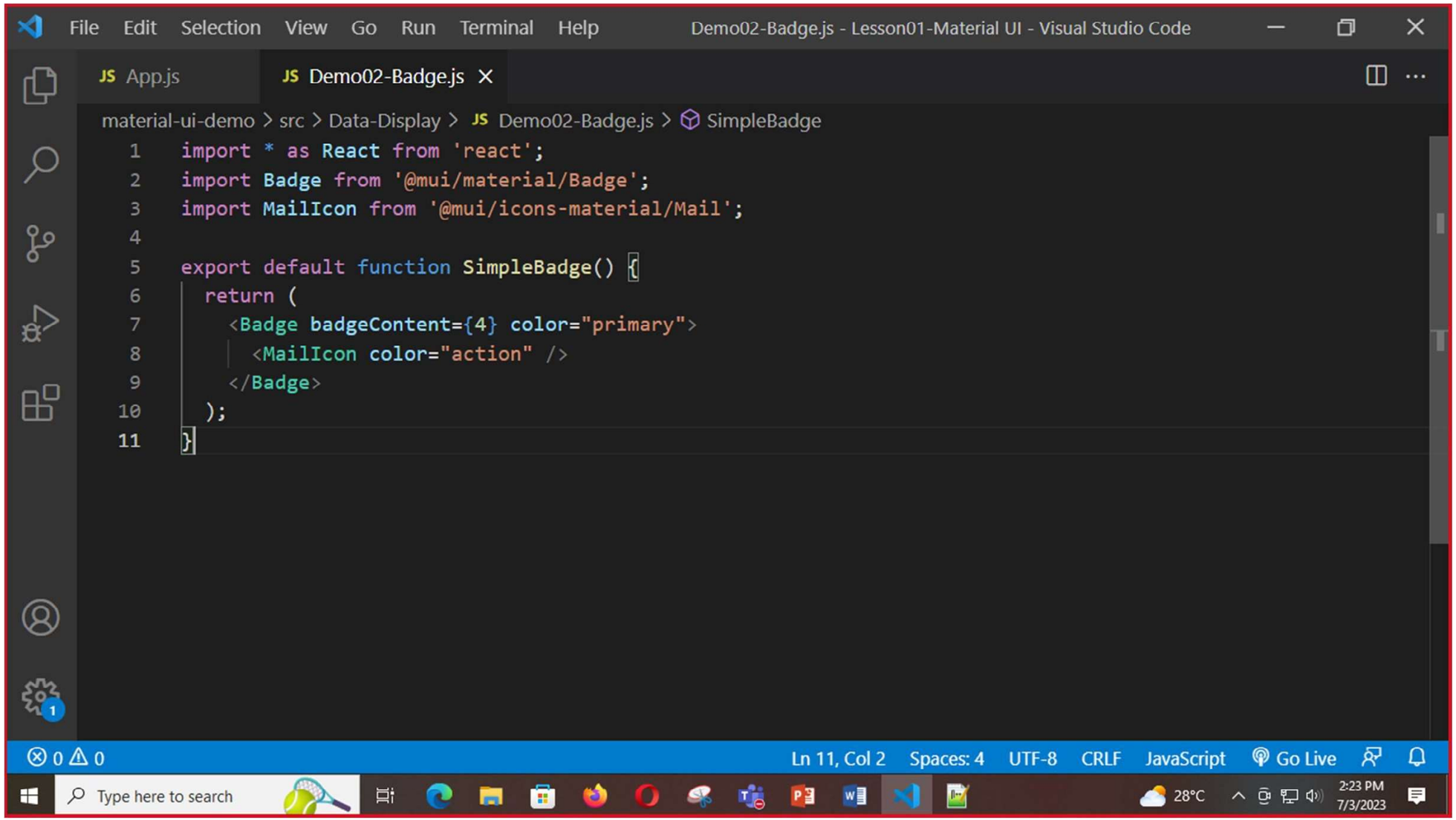



File Edit Selection View Go Run Terminal Help
Demo01-Avatar.js - Lesson01-Material UI - Visual Studio Code
App.js
Demo01-Avatar.js
material-ui-demo > src > Data-Display > Demo01-Avatar.js > ImageAvatars
1 import * as React from 'react';
2 import Avatar from '@mui/material/Avatar';
3 import Stack from '@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 }
0 0
Ln 15, Col 2 Spaces: 4 UTF-8 CRLF JavaScript Go Live
Type here to search
27°C
1:15 PM
7/3/2023

Badge

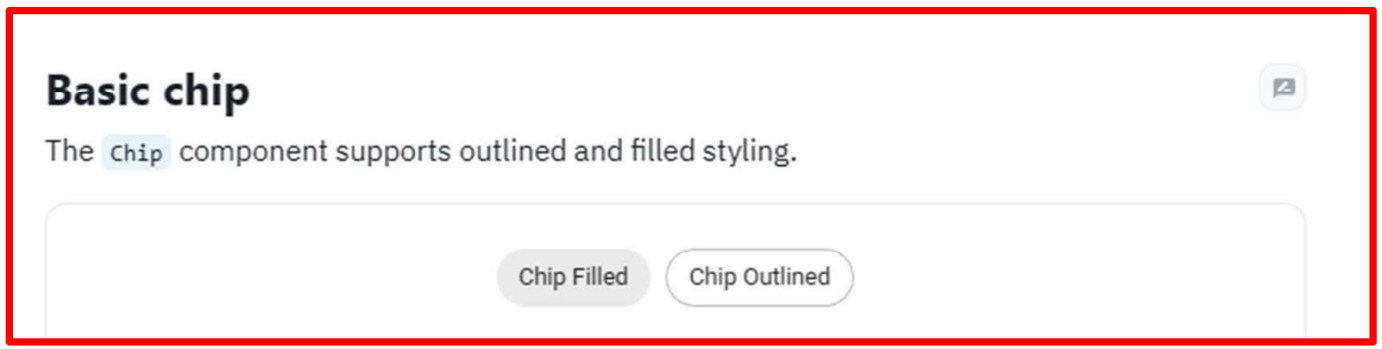
- Badge generates a small badge to the top-right of its child(ren).

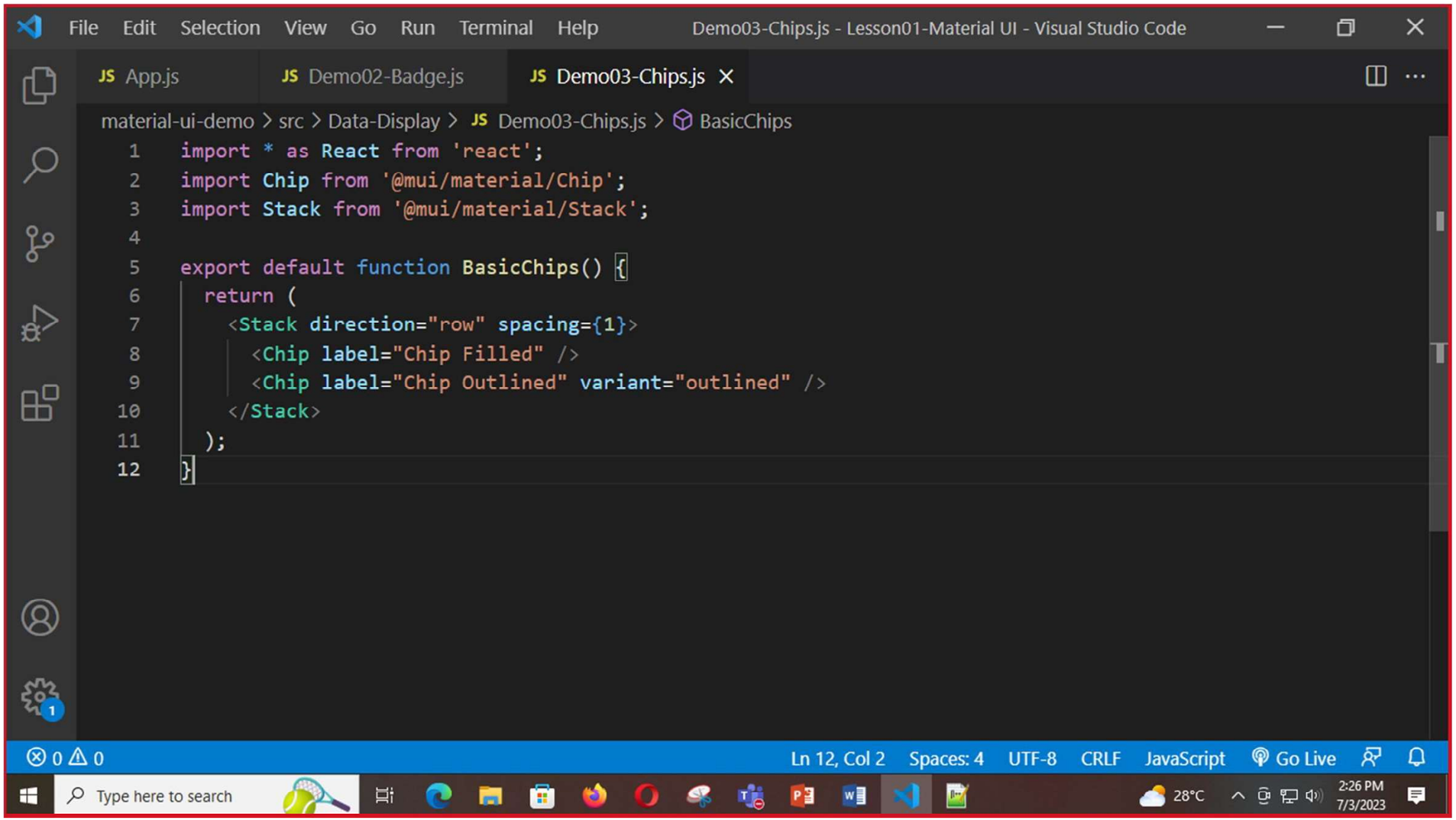



File Edit Selection View Go Run Terminal Help
Demo02-Badge.js - Lesson01-Material UI - Visual Studio Code
App.js
Demo02-Badge.js
material-ui-demo > src > Data-Display > Demo02-Badge.js > SimpleBadge
1 import * as React from 'react';
2 import Badge from '@mui/material/Badge';
3 import MailIcon from '@mui/icons-material/Mail';
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
28°C
2:23 PM
7/3/2023

Chips

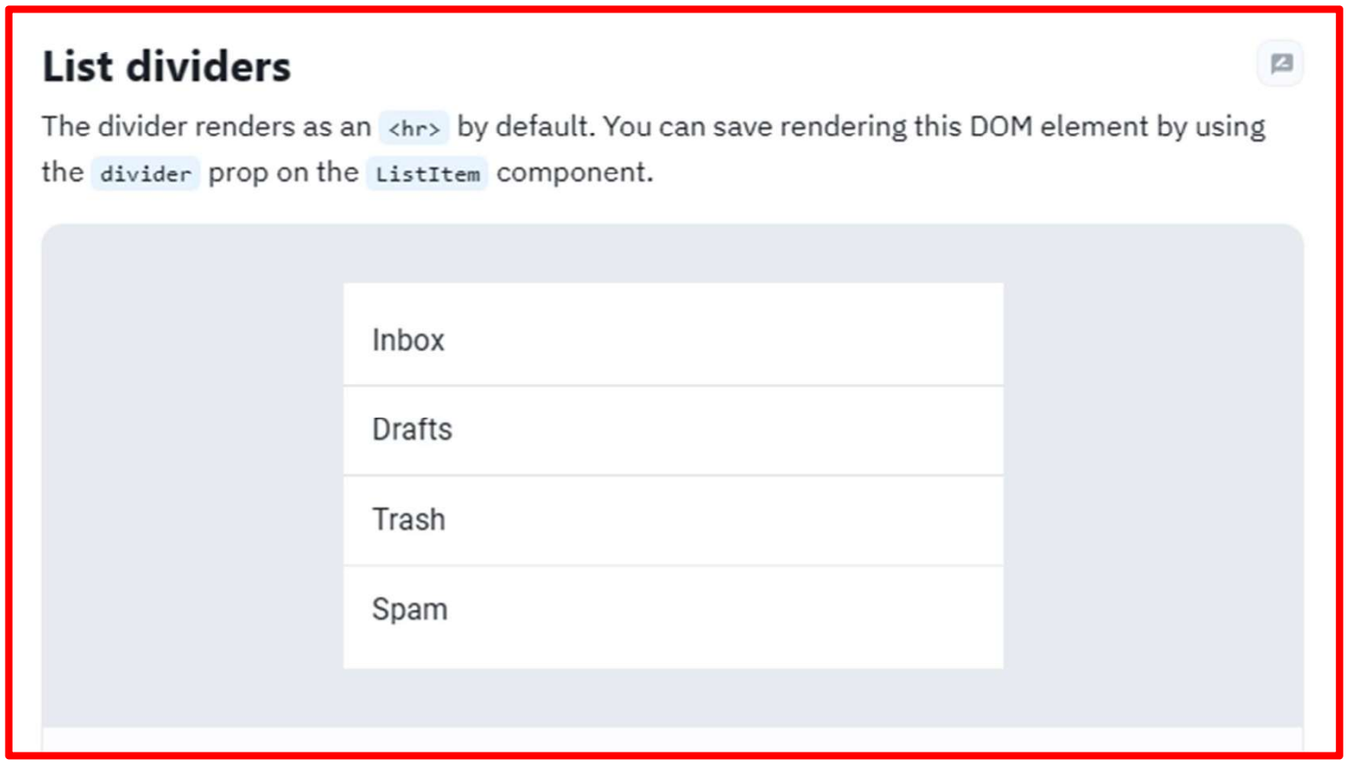
- Chips are compact elements that represent an input, attribute, or action.
- Chips allow users to enter information, make selections, filter content, or trigger actions.
- While included here as a standalone component, the most common use will be in some form of input, so some of the behavior demonstrated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

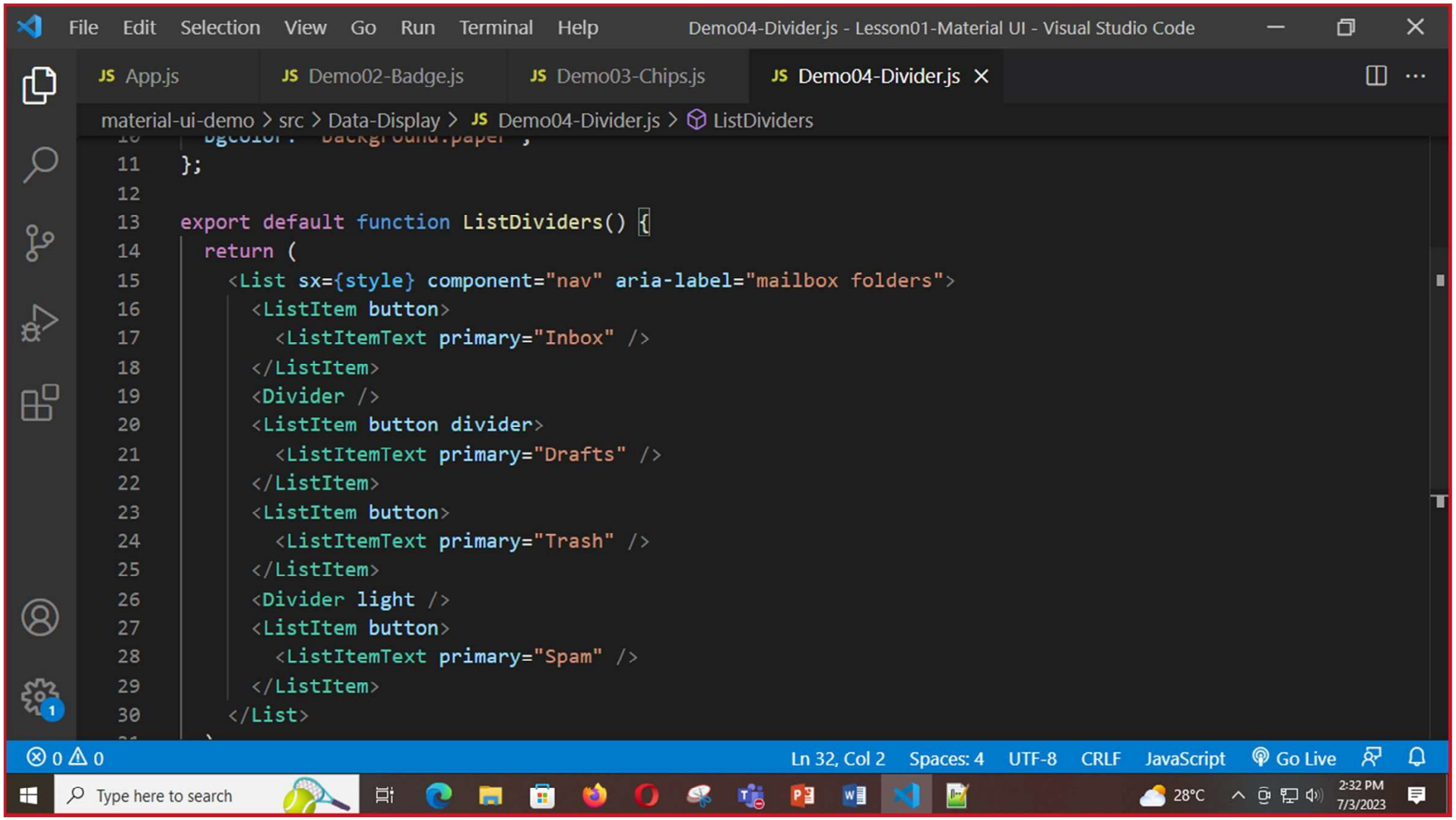



File Edit Selection View Go Run Terminal Help
Demo03-Chips.js - Lesson01-Material UI - Visual Studio Code
App.js
Demo02-Badge.js
Demo03-Chips.js
material-ui-demo > src > Data-Display > Demo03-Chips.js > BasicChips
1 import * as React from 'react';
2 import Chip from '@mui/material/Chip';
3 import Stack 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 }
0 0
Ln 12, Col 2
Spaces: 4
UTF-8
CRLF
JavaScript
Go Live
Type here to search
28°C
2:26 PM
7/3/2023

Divider

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

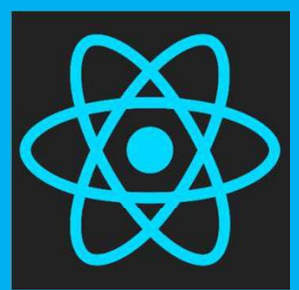



File Edit Selection View Go Run Terminal Help
Demo04-Divider.js - Lesson01-Material UI - Visual Studio Code
App.js
Demo02-Badge.js
Demo03-Chips.js
Demo04-Divider.js
material-ui-demo > src > Data-Display > Demo04-Divider.js > ListDividers
};
export default function ListDividers() {
return (
<List sx={style} component="nav" aria-label="mailbox folders">
<ListItem button>
<ListItemText primary="Inbox" />
</ListItem>
<Divider />
<ListItem button divider>
<ListItemText primary="Drafts" />
</ListItem>
<ListItem button>
<ListItemText primary="Trash" />
</ListItem>
<Divider light />
<ListItem button>
<ListItemText primary="Spam" />
</ListItem>
</List>
Ln 32, Col 2
Spaces: 4
UTF-8
CRLF
JavaScript
Go Live
Type here to search
28°C
2:32 PM
7/3/2023

Icons

- Guidance and suggestions for using icons with Material UI.

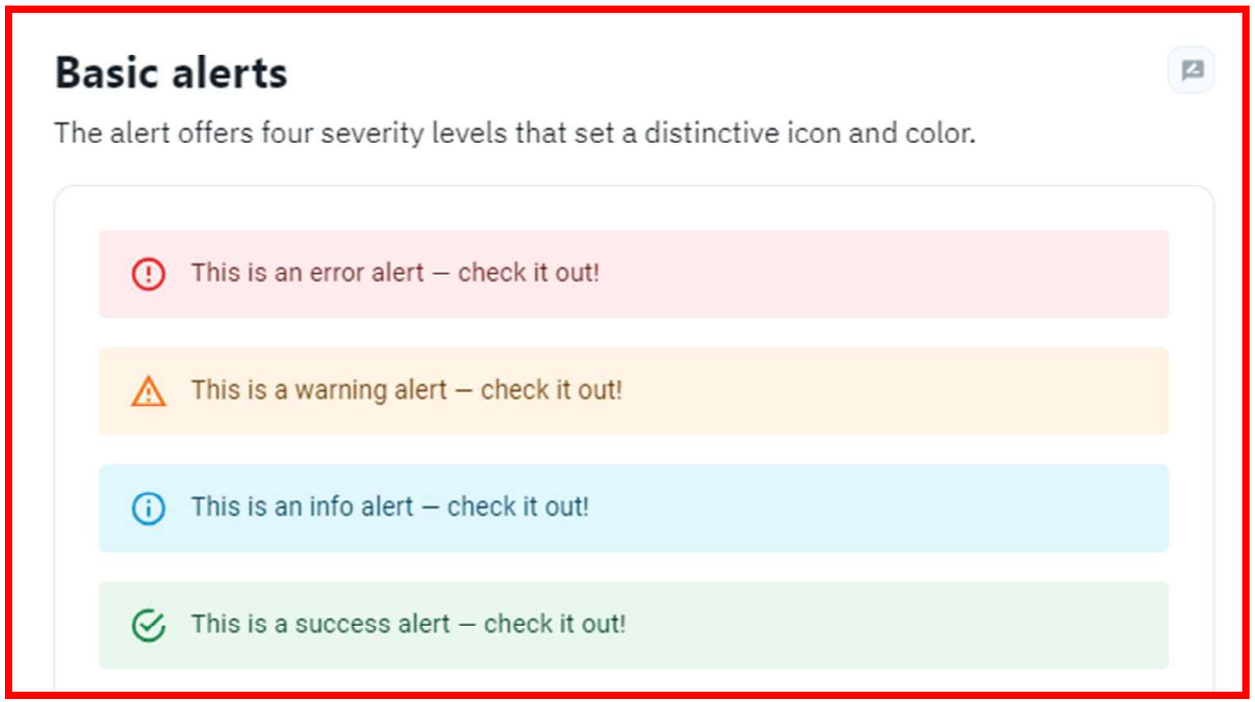
- Material UI provides icon support in three ways:
  1. Standardized Material Icons exported as React components (SVG icons).
  2. With the SvgIcon component, a React wrapper for custom SVG icons.
  3. With the Icon component, a React wrapper for custom font icons.

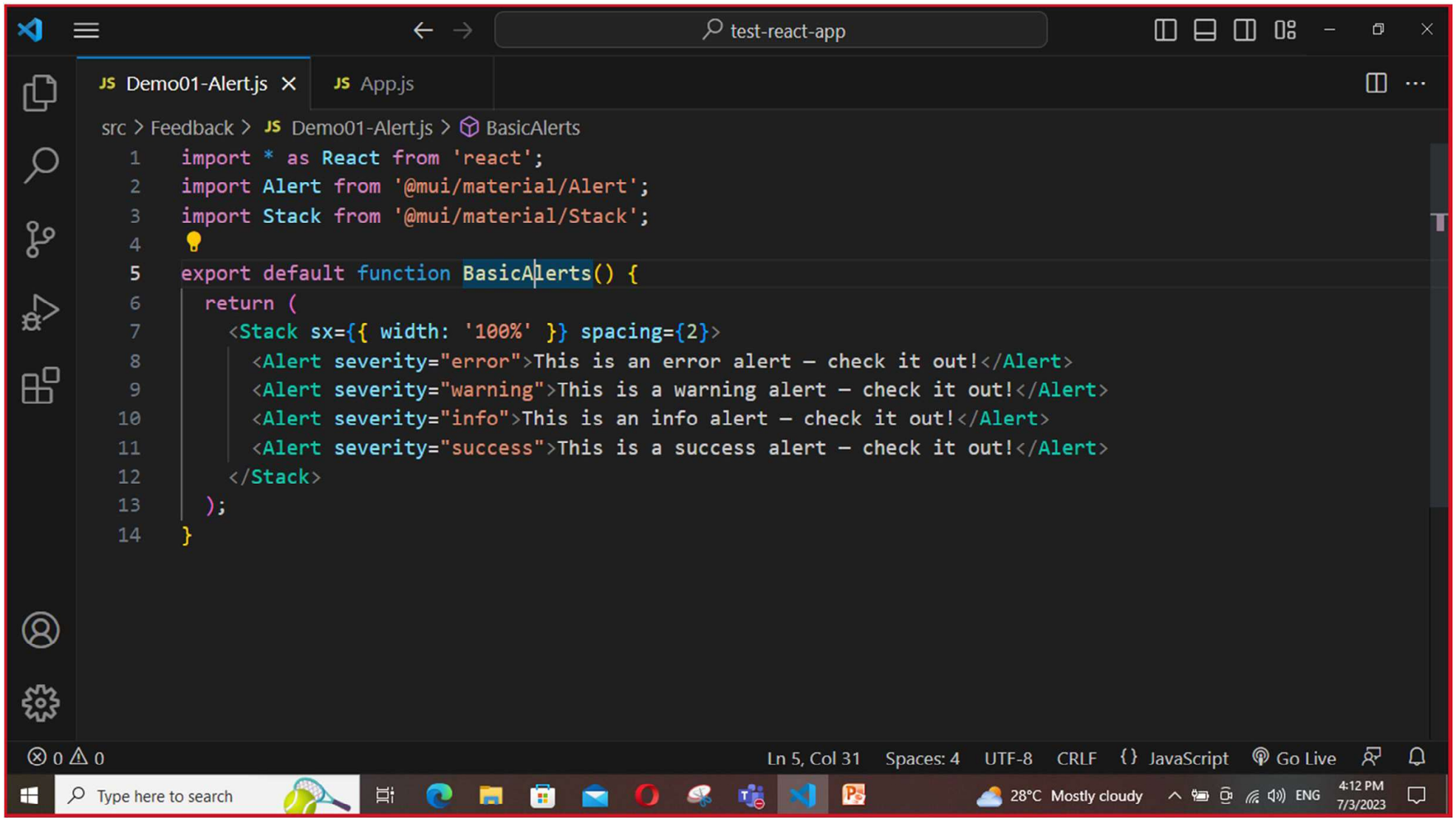


Feedback

Alert

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

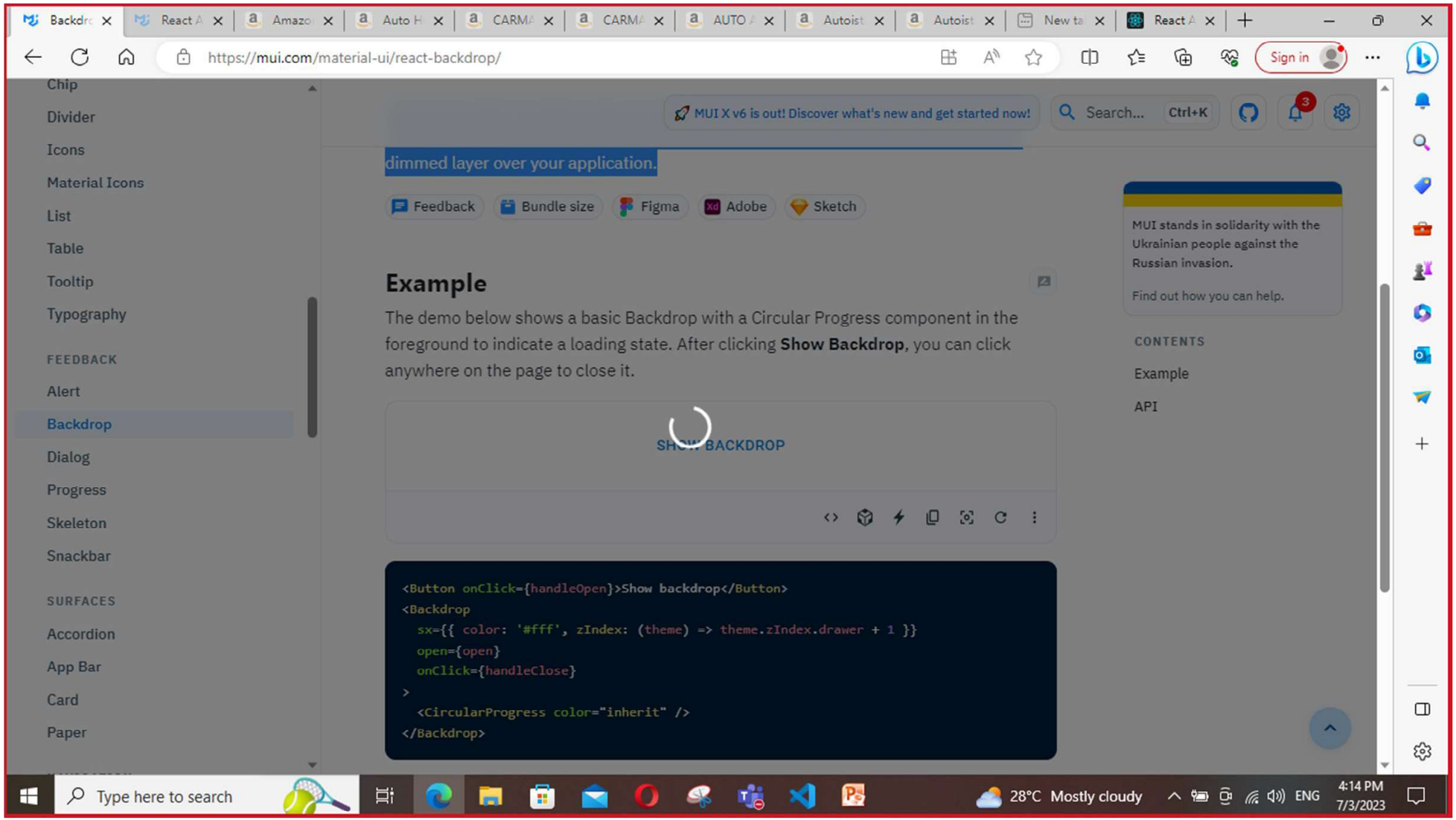



test-react-app
Demo01-Alert.js
App.js
src > Feedback > Demo01-Alert.js > BasicAlerts
1 import * as React from 'react';
2 import Alert from '@mui/material/Alert';
3 import Stack from '@mui/material/Stack';
4
5 export default function BasicAlerts() {
6 return (
7 <Stack sx={{ width: '100%' }} spacing={2}>
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 }
Ln 5, Col 31
Spaces: 4
UTF-8
CRLF
JavaScript
Go Live
Type here to search
28°C Mostly cloudy
ENG
4:12 PM
7/3/2023

Backdrop

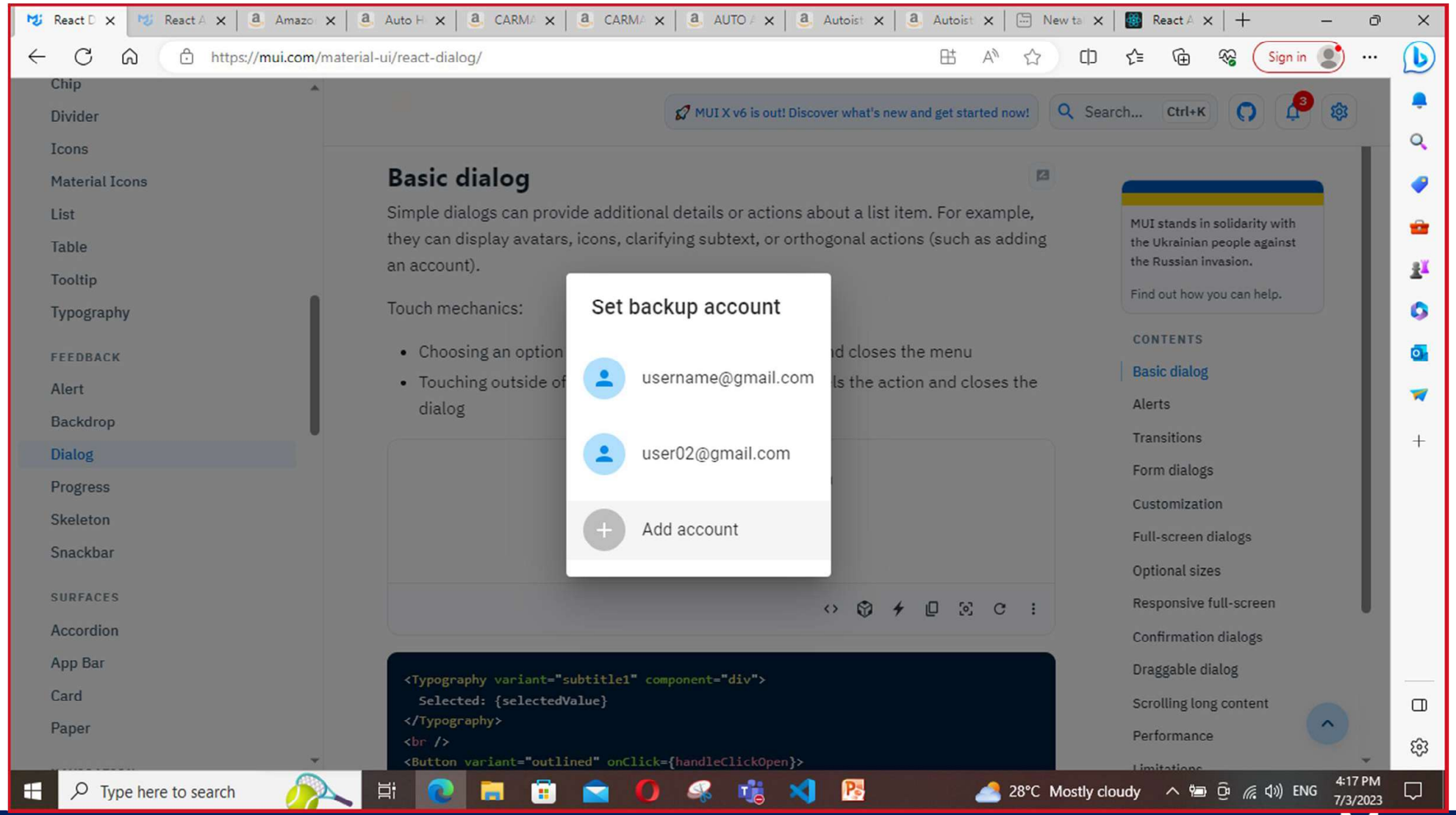
- The Backdrop component narrows the user's focus to a particular element on the screen.

- 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.

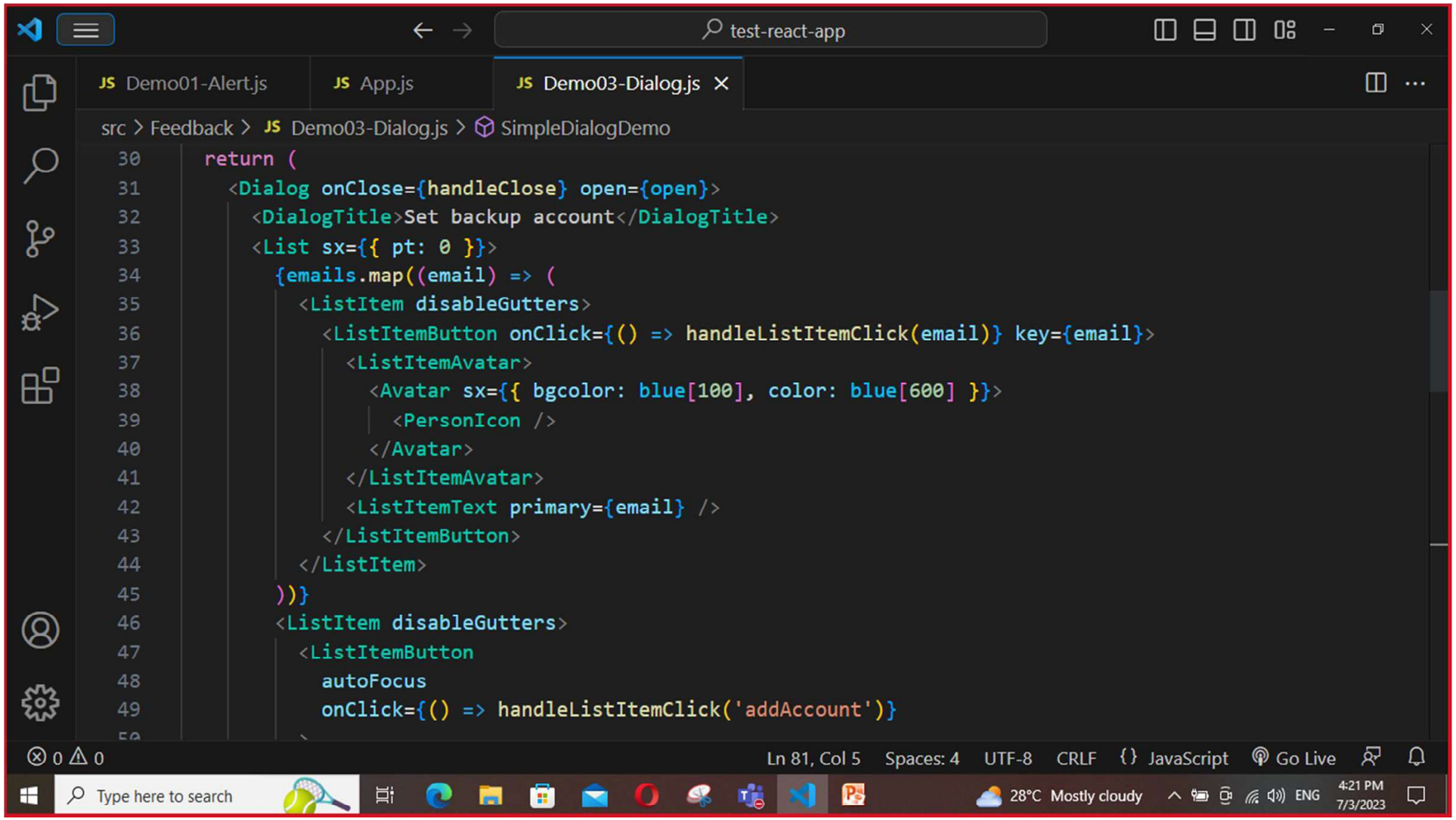

https://mui.com/material-ui/react-backdrop/
Sign in
Chip
Divider
Icons
Material Icons
List
Table
Tooltip
Typography
FEEDBACK
Alert
Backdrop
Dialog
Progress
Skeleton
Snackbar
SURFACES
Accordion
App Bar
Card
Paper
MUI X v6 is out! Discover what's new and get started now!
Search... Ctrl+K
dimmed layer over your application.
Feedback
Bundle size
Figma
Adobe
Sketch
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
Type here to search
28°C Mostly cloudy
ENG
4:14 PM
7/3/2023

Dialog

- 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.


https://mui.com/material-ui/react-dialog/
Sign in
MUI X v6 is out! Discover what's new and get started now!
Search... Ctrl+K
Chip
Divider
Icons
Material Icons
List
Table
Tooltip
Typography
FEEDBACK
Alert
Backdrop
Dialog
Progress
Skeleton
Snackbar
SURFACES
Accordion
App Bar
Card
Paper
Basic dialog
Simple dialogs can provide additional details or actions about a list item. For example, they can display avatars, icons, clarifying subtext, or orthogonal actions (such as adding an account).
Touch mechanics:
Choosing an option
Touching outside of
dialog
closes the menu
the action and closes the
Set backup account
username@gmail.com
user02@gmail.com
Add account
MUI stands in solidarity with the Ukrainian people against the Russian invasion.
Find out how you can help.
CONTENTS
Basic dialog
Alerts
Transitions
Form dialogs
Customization
Full-screen dialogs
Optional sizes
Responsive full-screen
Confirmation dialogs
Draggable dialog
Scrolling long content
Performance
<Typography variant="subtitle1" component="div">
Selected: {selectedValue}
</Typography>

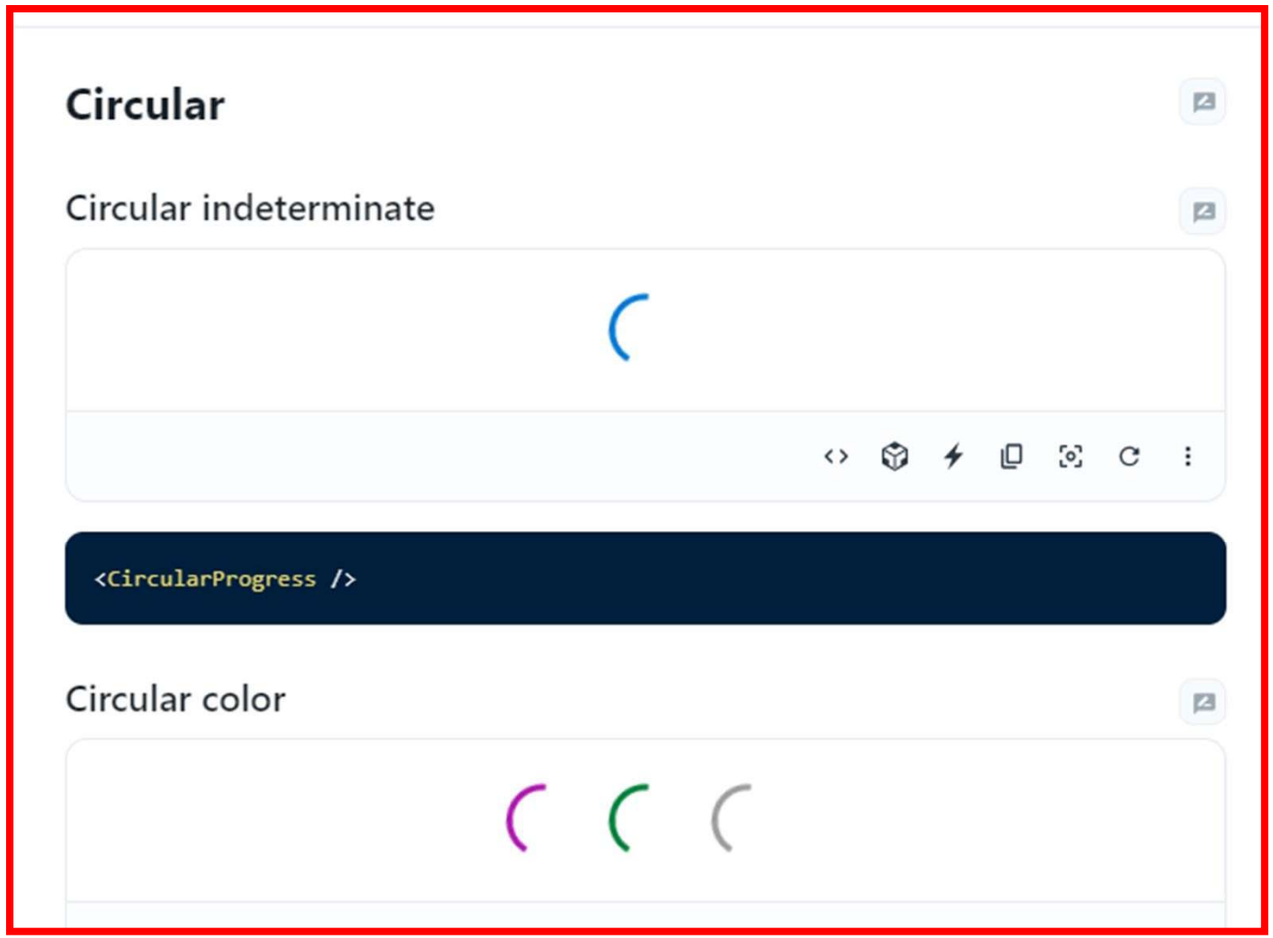
<Button variant="outlined" onClick={handleClickOpen}>
Type here to search
28°C Mostly cloudy
ENG
4:17 PM
7/3/2023


test-react-app
Demo01-Alert.js
App.js
Demo03-Dialog.js
src > Feedback > 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 <ListItemButton onClick={() => handleListItemClick(email)} key={email}>
37 <ListItemAvatar>
38 <Avatar sx={{ bgcolor: blue[100], color: blue[600] }}>
39 <PersonIcon />
40 </Avatar>
41 </ListItemAvatar>
42 <ListItemText primary={email} />
43 </ListItemButton>
44 </ListItem>
45))}
46 <ListItem disableGutters>
47 <ListItemButton
48 autoFocus
49 onClick={() => handleListItemClick('addAccount')}
0 0
Ln 81, Col 5
Spaces: 4
UTF-8
CRLF
JavaScript
Go Live
Type here to search
28°C Mostly cloudy
ENG
4:21 PM
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.


Circular
Circular indeterminate
<CircularProgress />
Circular color

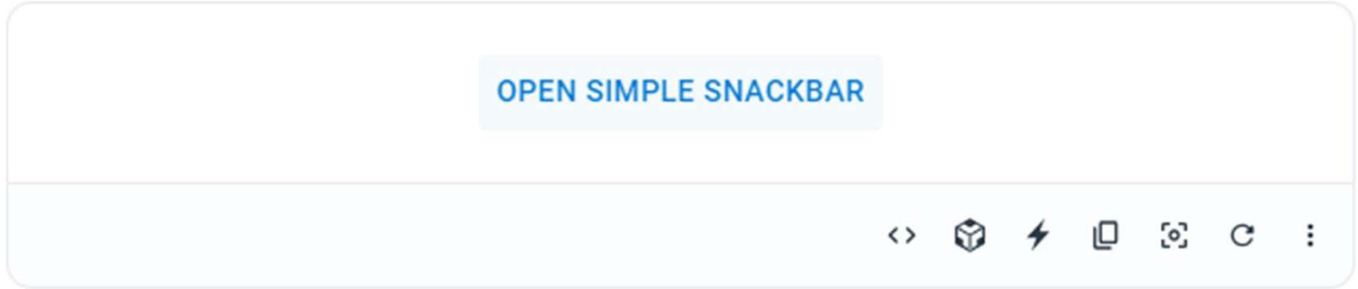
Snackbar

- Snackbars provide brief notifications. The component is also known as a toast.

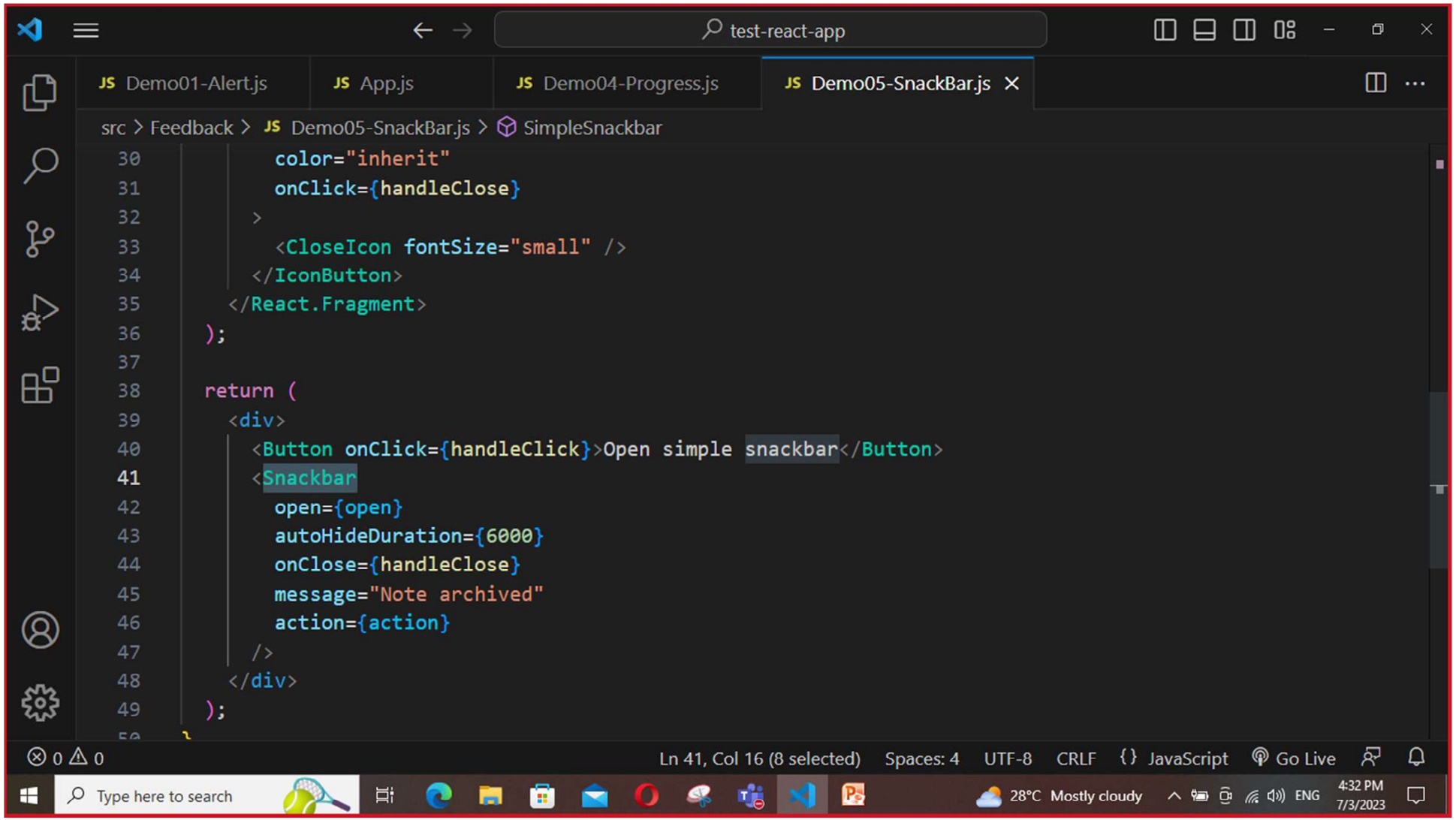
- Snackbars inform users of a process that an app has performed or will perform. They appear temporarily, towards the bottom of the screen. They shouldn't interrupt the user experience, and they don't require user input to disappear.
- Snackbars contain a single line of text directly related to the operation 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.

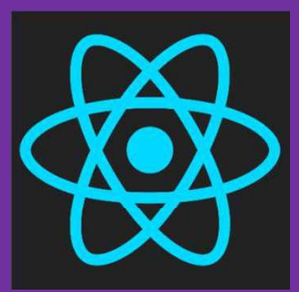



```
<Button onClick={handleClick}>Open simple snackbar</Button>
<Snackbar
  open={open}
  autoHideDuration={6000}
  onClose={handleClose}
  message="Note archived"
  action={action}
/>
```


test-react-app
Demo01-Alert.js
App.js
Demo04-Progress.js
Demo05-SnackBar.js
src > Feedback > Demo05-SnackBar.js > SimpleSnackbar
color="inherit"
onClick={handleClose}
<CloseIcon fontSize="small" />
</IconButton>
</React.Fragment>
);
return (

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

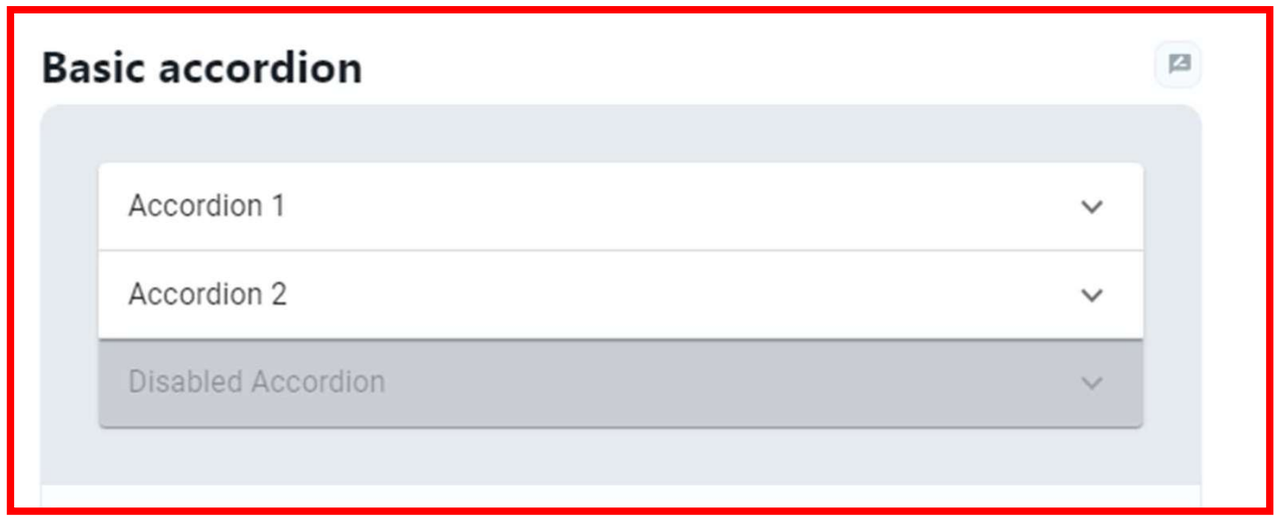
);
Ln 41, Col 16 (8 selected)
Spaces: 4
UTF-8
CRLF
JavaScript
Go Live
Type here to search
28°C Mostly cloudy
ENG
4:32 PM
7/3/2023

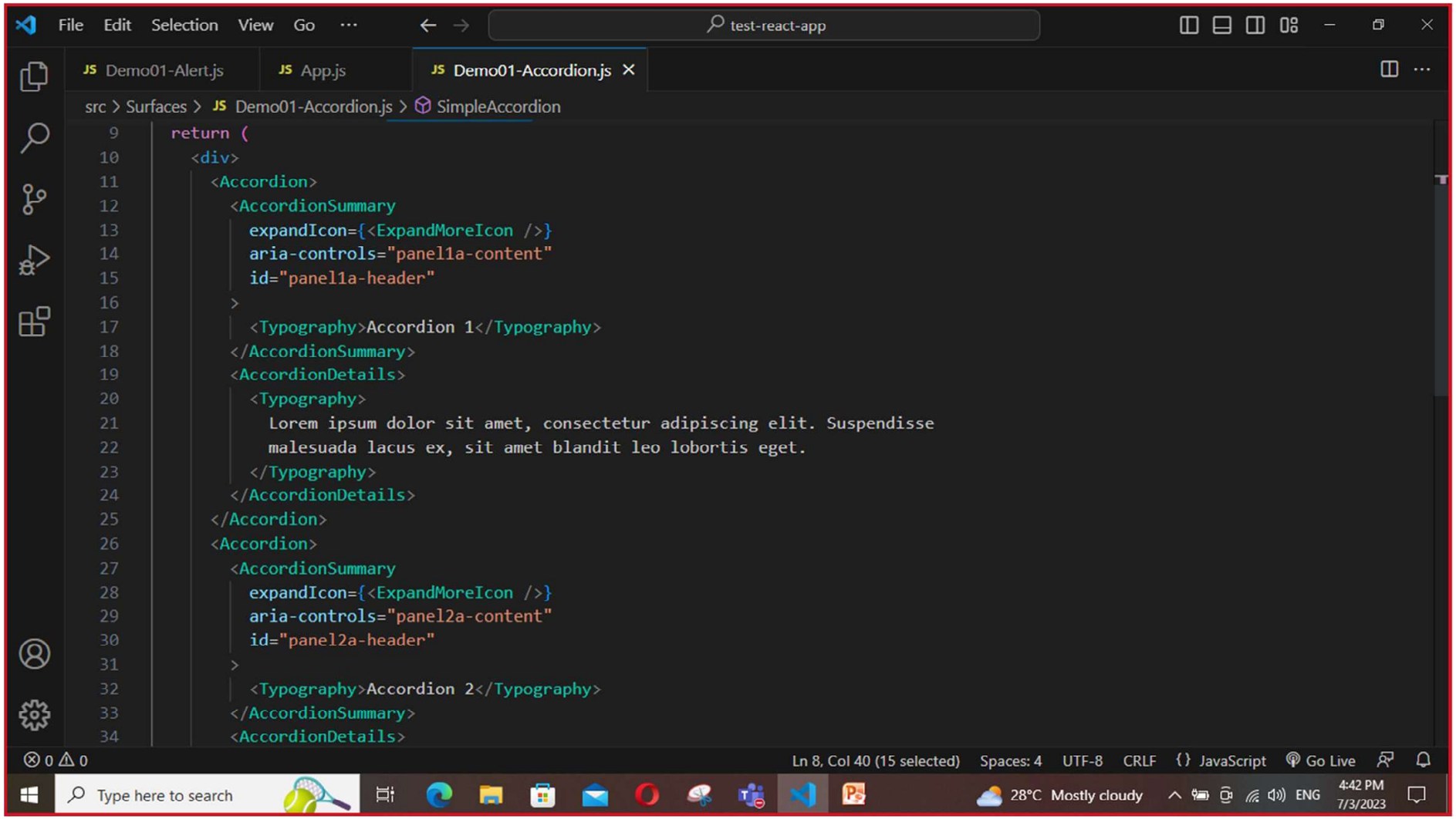


Surfaces

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.

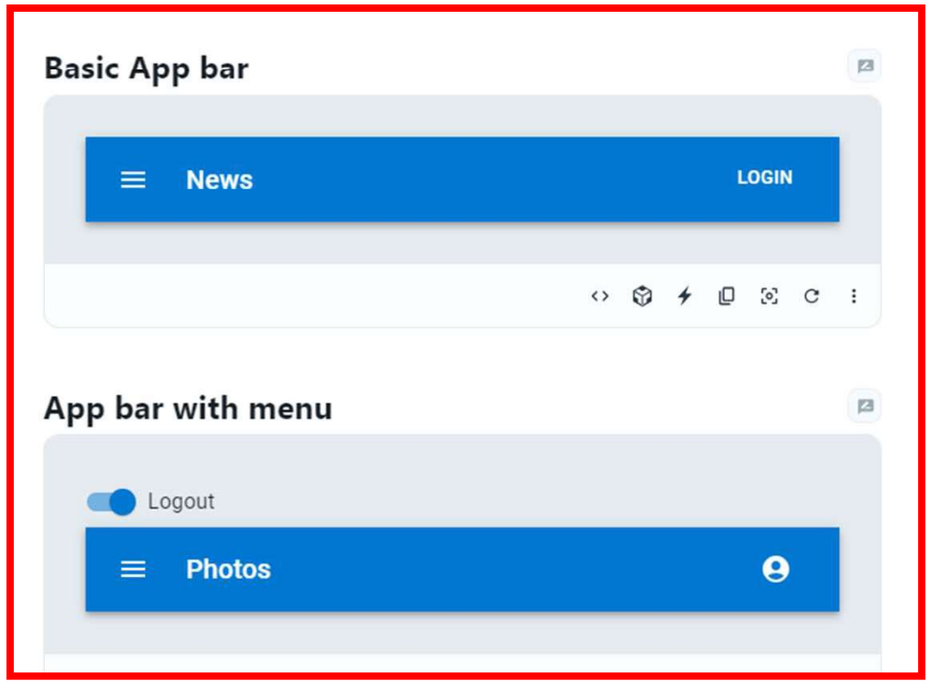


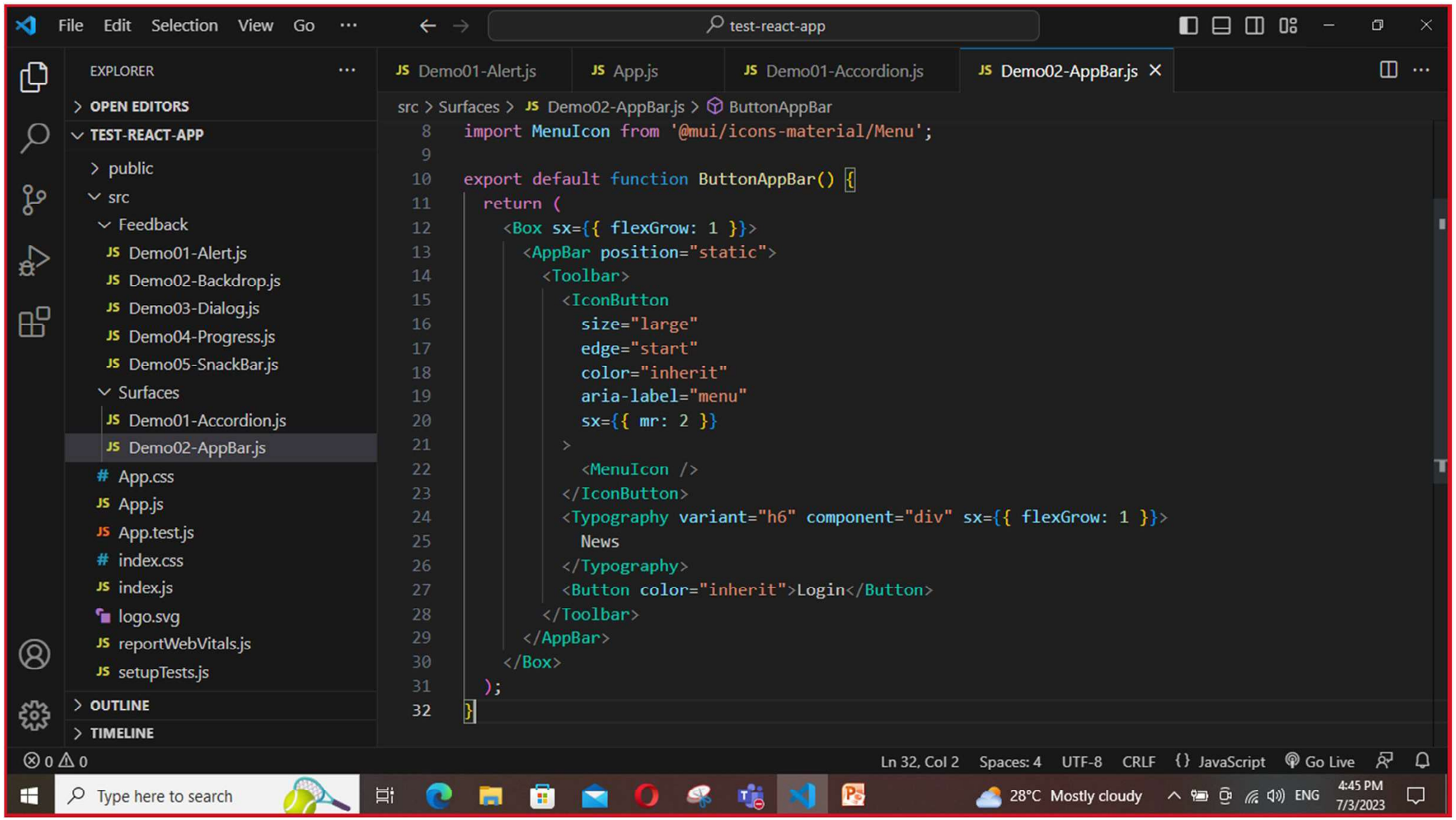

File Edit Selection View Go ···
test-react-app
Demo01-Alert.js
App.js
Demo01-Accordion.js
src > Surfaces > Demo01-Accordion.js > SimpleAccordion
return (

<Accordion>
<AccordionSummary
expandIcon={<ExpandMoreIcon />}
aria-controls="panel1a-content"
id="panel1a-header"
>
<Typography>Accordion 1</Typography>
</AccordionSummary>
<AccordionDetails>
<Typography>
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse
malesuada lacus ex, sit amet blandit leo lobortis eget.
</Typography>
</AccordionDetails>
</Accordion>
<Accordion>
<AccordionSummary
expandIcon={<ExpandMoreIcon />}
aria-controls="panel2a-content"
id="panel2a-header"
>
<Typography>Accordion 2</Typography>
</AccordionSummary>
<AccordionDetails>
0 0
Ln 8, Col 40 (15 selected)
Spaces: 4
UTF-8
CRLF
JavaScript
Go Live
Type here to search
28°C Mostly cloudy
ENG
4:42 PM
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
test-react-app
EXPLORER
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
Demo01-Alert.js
App.js
Demo01-Accordion.js
Demo02-AppBar.js
src > Surfaces > Demo02-AppBar.js > ButtonAppBar
import MenuIcon from '@mui/icons-material/Menu';
export default function ButtonAppBar() {
return (
<Box sx={{ flexGrow: 1 }}>
<AppBar position="static">
<Toolbar>
<IconButton
size="large"
edge="start"
color="inherit"
aria-label="menu"
sx={{ mr: 2 }}
>
<MenuIcon />
</IconButton>
<Typography variant="h6" component="div" sx={{ flexGrow: 1 }}>
News
</Typography>
<Button color="inherit">Login</Button>
</Toolbar>
</AppBar>
</Box>
);
}
Ln 32, Col 2
Spaces: 4
UTF-8
CRLF
JavaScript
Go Live
Type here to search
28°C Mostly cloudy
ENG
4:45 PM
7/3/2023

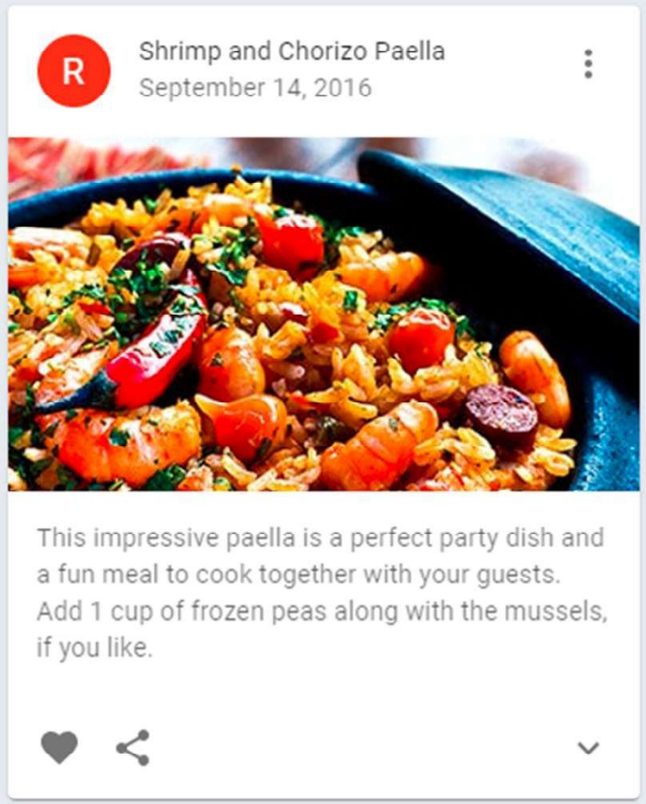
Card

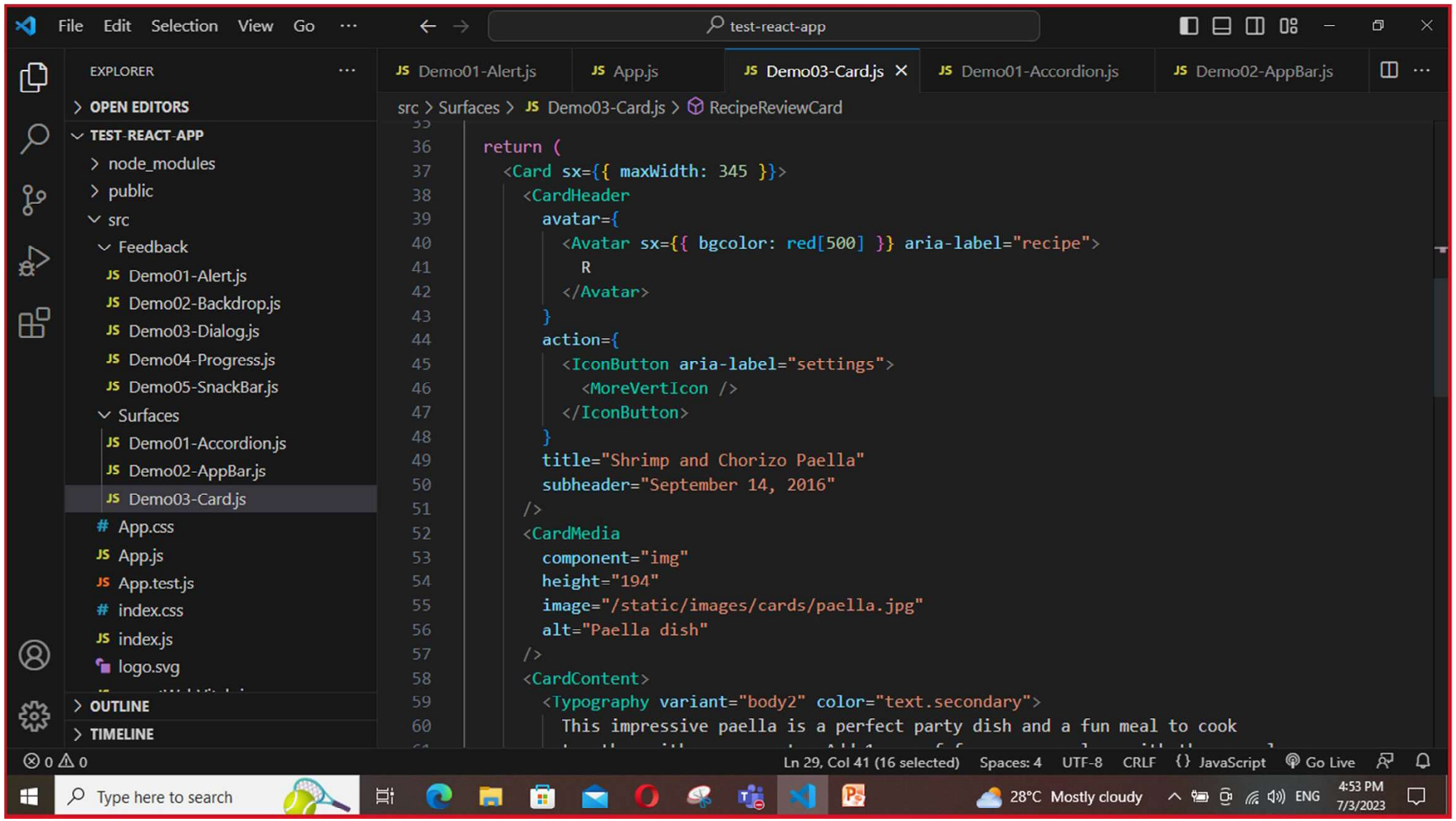
- Cards contain content and actions about a single subject.

- 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.)




File Edit Selection View Go
test-react-app
EXPLORER
OPEN EDITORS
TEST-REACT-APP
node_modules
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
Demo03-Card.js
App.css
App.js
App.test.js
index.css
index.js
logo.svg
OUTLINE
TIMELINE
Demo01-Alert.js
App.js
Demo03-Card.js
Demo01-Accordion.js
Demo02-AppBar.js
src > Surfaces > Demo03-Card.js > RecipeReviewCard
return (
<Card sx={{ maxWidth: 345 }}>
<CardHeader
avatar={
<Avatar sx={{ bgcolor: red[500] }} aria-label="recipe">
R
</Avatar>
}
action={
<IconButton aria-label="settings">
<MoreVertIcon />
</IconButton>
}
title="Shrimp and Chorizo Paella"
subheader="September 14, 2016"
/>
<CardMedia
component="img"
height="194"
image="/static/images/cards/paella.jpg"
alt="Paella dish"
/>
<CardContent>
<Typography variant="body2" color="text.secondary">
This impressive paella is a perfect party dish and a fun meal to cook
Ln 29, Col 41 (16 selected)
Spaces: 4
UTF-8
CRLF
JavaScript
Go Live
Type here to search
28°C Mostly cloudy
ENG
4:53 PM
7/3/2023