



...



# Material UI (MUI) for React: Basic to Advanced

Gnanavel Durairaj

# ■ Components & layouts

Component	Usage Example
AppBar	Top navigation bar
Drawer	Side navigation drawer
Grid / Box	Responsive layouts
Snackbar	Temporary messages/toasts
Tabs	Tabbed navigation

# ■ Agenda

- MUI overview
- Installation & setup
- Components & layouts
- Theming & customization
- Data display & interactivity
- Advanced MUI features

# ■ MUI overview

## What is Material UI (MUI)?

- MUI is a comprehensive React UI library following Google's Material Design.
- Used by top companies for consistent, modern UIs in React apps.
- Provides prebuilt, customizable components.

## Why Use MUI?

- Ready-to-use, accessible, themeable widgets
- Fast development; consistent UX
- Active community, robust documentation

# ■ Installation & Setup

## Installing MUI

### Required:

Node & npm

### Install core package and dependencies:

- `npm install @mui/material @emotion/react @emotion/styled6`

# ■ Components & layouts

## MUI Project Structure

```
my-mui-app/
  ├── public/
  |   └── index.html
  ├── src/
  |   ├── assets/          # Static files (images, fonts, etc.)
  |   ├── components/      # Reusable MUI-based components (e.g., ButtonGroup, Navbar, Card)
  |   ├── layouts/          # Page layout components using MUI's Grid, Box, etc.
  |   ├── pages/            # Application views (e.g., Home, About, Dashboard)
  |   ├── theme/            # MUI custom theme configuration
  |   |   └── theme.js
  |   ├── App.js            # Root component
  |   ├── index.js          # App entry point
  |   └── utils/             # Helper functions or context providers
  └── .gitignore
  └── package.json
  └── README.md
  └── yarn.lock / package-lock.json
```

# ■ Components & layouts

## Using MUI Components

### Example:

```
import Button from '@mui/material/Button';  
<Button variant="contained">Click Me</Button>
```

Component	Usage Example
Button	Submit, Reset, Click actions
TextField	Forms and search inputs
Card	Display info blocks (image, text, actions)
Dialog	Popup/modal windows

# ■ Components & layouts

## Layout components

**Container**: center content

**Box**: utility wrapper

**Grid**: responsive layouts

**Stack**: vertical/horizontal stacking

### Example : Container

```
import { Container } from '@mui/material';
<Container maxWidth="md">
  Centered content with padding
</Container>
```

# ■ Components & layouts

## Example: Box

```
import { Box } from '@mui/material';

<Box sx={{ p: 2, bgcolor: 'lightblue' }}>
  Content inside a Box
</Box>
```

## Example: Stack

```
import { Stack, Button } from '@mui/material';
<Stack direction="row" spacing={2}>
  <Button variant="contained">One</Button>
  <Button variant="outlined">Two</Button>
</Stack>
```

# ■ Components & layouts

## Example : Grid

```
import { Grid } from '@mui/material';

<Grid container spacing={2}>
  <Grid item xs={6}>
    Left
  </Grid>
  <Grid item xs={6}>
    Right
  </Grid>
</Grid>
```

# ■ Theming & customization

## Typography in MUI (Material UI)

Typography in MUI refers to the text styles used in your application—such as headings, body text, captions, etc.—based on Google's Material Design guidelines.

### Example:

```
import { Typography } from '@mui/material';
```

```
<Typography variant="h4" component="h1" gutterBottom>
  Welcome to MUI
</Typography>
```

# ■ Theming & customization

## Buttons in MUI

The `<Button />` component in MUI is a core interactive element used to trigger actions like submit, navigate, or perform logic.

- Button variants: contained, outlined, text
- Colors: primary, secondary, etc.
- Icon buttons

## Button Variants

```
<Button variant="contained">Contained</Button>
<Button variant="outlined">Outlined</Button>
<Button variant="text">Text</Button>
```

# ■ Theming & customization

## Button Colors, Size and Icon Buttons

### Color

```
<Button color="secondary">Secondary</Button>  
<Button color="error">Delete</Button>
```

### Size

```
<Button size="small">Small</Button>  
<Button size="medium">Medium</Button>  
<Button size="large">Large</Button>
```

### Icon Buttons

```
import { Button } from '@mui/material';  
import SaveIcon from '@mui/icons-material/Save';
```

```
<Button variant="contained" startIcon={<SaveIcon />}>  
  Save  
</Button>
```

# ■ Theming & customization

## Drawer (Sidebar Menu)

- Drawer for navigation
- SwipeableDrawer for mobile
- Persistent and temporary drawer options

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

### Example

```
import { Drawer, Button } from '@mui/material';
const [open, setOpen] = useState(false);
<Button onClick={() => setOpen(true)}>Open</Button>
<Drawer open={open} onClose={() => setOpen(false)}>
  <div style={{ width: 200, padding: 20 }}>Sidebar Menu</div>
</Drawer>
```

# ■ Theming & customization

## Cards

- Card, CardHeader,CardContent,CardActions
- Useful for displaying structured content, e.g., user profiles

Example:

```
import { Card, CardHeader,CardContent,CardActions,Button,Avatar,Typography } from '@mui/material';
<Card sx={{ maxWidth: 300 }}>
  <CardHeader avatar={<Avatar>P</Avatar>} title="Pugazh" subheader="Fullstack
Developer" />
  <CardContent><Typography variant="body2">Skilled in React & Node
js</Typography></CardContent>
  <CardActions><Button size="small">View</Button><Button
size="small">Message</Button></CardActions>
</Card>
```

# ■ Theming & customization

## Lists

- List & ListItem for menus, settings, or messages
- Demo: vertical or nested lists

### Example:

```
import { List, ListItem, ListItemText } from '@mui/material';
<List>
  <ListItem><ListItemText primary="Home" /></ListItem>
  <ListItem><ListItemText primary="Settings" /></ListItem>
  <ListItem><ListItemText primary="Logout" /></ListItem>
</List>
```

# ■ Theming & customization

## Tables

- Table, TableRow, TableCell for data grids
- DataGrid component (adv) for sorting/filtering/pagination

## Example:

```
import { Table, TableRow, TableCell, TableBody, TableHead } from
'@mui/material';
<Table>
  <TableHead><TableRow><TableCell>Name</TableCell>
    <TableCell>Role</TableCell></TableRow></TableHead>
    <TableBody><TableRow><TableCell>Pugazh</TableCell>
      <TableCell>Developer</TableCell></TableRow></TableBody>
</Table>
```

# ■ Theming & customization

## Forms & Inputs

- TextField, Select, Checkbox, Radio, Switch
- InputAdornment for icons
- FormGroup to group fields

## Examples

```
import { TextField, Select, MenuItem, Checkbox, FormGroup, FormControlLabel, Radio, RadioGroup } from '@mui/material';
<FormGroup>
  <TextField label="Name" />
  <Select defaultValue=""><MenuItem value="dev">Developer</MenuItem></Select>
  <FormControlLabel control={<Checkbox />} label="Subscribe" />
  <RadioGroup row><FormControlLabel control={<Radio />} value="male" label="Male" />
</RadioGroup>
</FormGroup>
```

# ■ Theming & customization

## Dialogs & Modals

Dialogs and Modals are used to grab user attention for important tasks like confirmations, alerts, or input forms.

- Dialog for pop-up forms/alerts
- Modal for custom overlays

### Example:

```
import { Dialog, DialogTitle,DialogContent, DialogActions, Button } from  
'@mui/material';  
<Dialog open={open} onClose={() => setOpen(false)}>  
  <DialogTitle>Alert</DialogTitle>  
  <DialogContent>This is a simple dialog.</DialogContent>  
  <DialogActions><Button onClick={() => setOpen(false)}>Close</Button>  
  </DialogActions>  
</Dialog>
```

# ■ Theming & customization

## Snackbars & Alerts

Snackbar for temporary messages  
Alert for feedback/status

### Examples:

```
import { Snackbar, Alert, Button } from '@mui/material';
<Snackbar open={open} autoHideDuration={3000} onClose={() =>
  setOpen(false)}>
  <Alert severity="success" onClose={() => setOpen(false)}>Saved successfully!
  </Alert>
</Snackbar>
<Button onClick={() => setOpen(true)}>Show Snackbar</Button>
```

# ■ Data display & interactivity

## ➤ Icons in MUI

### **Install:**

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

### **Example:**

```
import DeleteIcon from '@mui/icons-material/Delete';
<IconButton><DeleteIcon /></IconButton>
```

## ➤ Theming Basics

Customize colors, spacing, typography

### **Example:**

```
import { createTheme, ThemeProvider } from '@mui/material/styles';
const theme = createTheme({ palette: { primary: main: '#0055ff' } });
```

# ■ Data display & interactivity

## ➤ Customizing Components

- sx prop for quick styling
- styled() API for custom styles

### Example:

```
import { Button, styled } from '@mui/material';
const CustomBtn = styled(Button)({ backgroundColor: 'purple', color: 'white' });
<Button sx={{ m: 1, bgcolor: 'green', color: 'white' }}>SX Button</Button>
<CustomBtn>Styled Button</CustomBtn>
```

# ■ Advanced MUI features.

## Dark & Light Mode

### Why Dark/Light Mode?

It improves user experience by matching the app's appearance with user/system preferences (e.g., dark at night, light in daytime).

#### Example:

```
const darkMode = true; // or useState toggle
const theme = createTheme({
  palette: { mode: darkMode ? 'dark' : 'light' },
});
<ThemeProvider theme={theme}>
  <YourApp />
</ThemeProvider>
```

# ■ Advanced MUI features.

## Why Responsiveness & Breakpoints?

To make your UI adapt beautifully across mobile, tablet, and desktop screen sizes.

### Example:

```
import { Grid, Paper } from '@mui/material';
<Grid container spacing={2}>
  <Grid item xs={12} sm={6}><Paper>Left</Paper></Grid>
  <Grid item xs={12} sm={6}><Paper>Right</Paper></Grid>
</Grid>
```

# ■ Advanced MUI features.

## MUI Best Practices

- **Use ThemeProvider at the root**
  - Centralizes theming, colors, typography, dark/light mode, etc.
- **Stay updated with the latest MUI version**
  - To access new features, security patches, and performance improvements.
- **Follow Material Design Guidelines**
  - Ensures consistent UI/UX aligned with Google's design system.
- **Use sx for quick styles, styled() for reusable components**
  - Keeps styling clean and maintainable.
- **Leverage built-in components instead of custom CSS**
  - MUI handles responsiveness, accessibility, and design best practices.

# ■ Advanced MUI features.

## Resources & Next Steps

### Official Docs:

[mui.com](https://mui.com) – Complete documentation, examples, and guides.

### Example Repos:

Search “MUI Templates” on GitHub for real-world UIs and dashboards.

### Community Support:

- [Stack Overflow](https://stackoverflow.com/questions/tagged/material-ui) (material-ui tag)
- MUI Discord
- GitHub Issues for bug reports & contributions

### Next Steps:

- Start a small project using MUI
- Explore advanced features like theming, responsive layout, and accessibility