



React components for faster & easier web development.  
Highly customizable.

1) Installation (See docs of MUI)  
npm install @material-ui/core

also it uses Roboto font

So the link shown in doc should be included in index.html.

For icons have to separately include them

npm install @material-ui/icons.

Note :-

Material UI follows principles of Material. Material is a design system created by google to help teams build high-quality digital experience for Android, iOS, flutter & web.

2) Typography

Too many type sizes & styles at once can spoil any layout. A typographic scale has a limited set of type sizes that work well together.

2 ways to import → import Typography from '@material-ui/core/Typography';  
import {Typography} from '@material-ui/core';

# in place of h1, h2, p, span etc use Typography.

<Typography variant="h1" color='textSecondary' gutterBottom>

Hello world

</Typography>

• see below in docs API  
(open to see all Props available)

### 3) Button

search on top button

1) default button have no border.

2) contained button have border.

```
<Button variant = "contained" color = "primary">  
  Submit  
</Button>
```

3) Can add a link button also  
Can add our styles too or even a ClassName.

```
<Button style = {{ "color": "blue" }}  
  variant = "outlined"  
  href = "#contained-buttons">  
  Link </Button>
```

4) to wrap some buttons together

import ButtonGroup from "@material-ui/core/ButtonGroup"

```
Now <ButtonGroup>  
  <Button> one </Button>  
  <Button> two </Button>  
</ButtonGroup>
```

### 4) Container

Acts like div to wrap elements.

by default add some padding & margin

## 2) Icons

npm install @material-ui/icons

> Search material icons

> Select type of icons → filled, outlined, ---  
click on icon to use

Copy link comes above.

eg import AddOutlinedIcon from '@material-ui/icons/AddOutlined';  
<AddOutlinedIcon />

→ can add prop

<AddOutlinedIcon color="Secondary" fontSize="large" />

eg add in buttons

<Button  
  type="Submit" color="Secondary"  
  startIcon={ <SendIcon /> }  
  endIcon={ <KeyboardArrowRightIcon /> }  
>  
  Submit  
</Button>

## 6) Custom Styles

only like this. (since is for not component)

import { makeStyles } from '@material-ui/core';

const useStyles = makeStyles({

  btn: {

    backgroundColor: 'violet',

    '&:hover': {

      backgroundColor: 'blue'

    }

  },

  title: {

    marginBottom: 20

  }

});

→ inside class, function  
const classes = useStyles();

<Button

  className={ classes.btn }

> Submit </Button>

### # Note

Can even add normal react className & inline styles.

## 7) Custom Theme

to see default theme  
search default theme.

- to add custom theme

import { createMuiTheme, ThemeProvider } from '@material-ui/core';

```
const theme = createMuiTheme({  
  palette: {  
    primary: {  
      main: '#fefefe'  
    },  
  },  
});
```

in → function APP () {

return (

<ThemeProvider theme={theme}>

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

</ThemeProvider>

}

- search color

show different colors provided by MUI

import { purple } from '@material-ui/colors';

and in → secondary: purple;

default add purple variants into 'main', light, dark etc.

## 3) FORM & Textarea (TextField)

Can either take material UI form or make our form & use their Textarea.

- import { makeStyles } from '@material-ui/core';  
import TextField from '@material-ui/core/TextField';

```
const useStyles = makeStyles({
```

```
  field: {
```

```
    marginTop: 20,
```

```
    marginBottom: 20,
```

```
    display: 'block'
```

```
  },
```

```
});
```

```
export default function Create() {
```

```
  const classes = useStyles();
```

```
  const [title, setTitle] = useState('');
```

```
  const [details, setDetails] = useState('');
```

```
  const [titleError, setTitleError] = useState(false)
```

```
  const [detailsError, setDetailsError] = useState(false)
```

```
  const handleSubmit = (e) => {
```

```
    e.preventDefault();
```

```
    setTitleError(false);
```

```
    setDetailsError(false);
```

```
    if (title === '') {
```

```
      setTitleError(true);
```

```
    }
```

```
    if (details === '') {
```

```
      setDetailsError(true);
```

```
    }
```

```
    if (title && details) {
```

```
      console.log(title, details)
```

```
    }
```

```
  }
```



```
<form novalidate autoComplete="off" onSubmit={handleSubmit}>
```

```
< TextField
```

```
  onChange = { (e) => setTitle (e.target.value) }  
  className = { classes.field }      label = "Note Title"  
  color = "primary"      variant = "outlined"  
  fullWidth  
  required  
  error = { titleError }  
/>
```

```
< TextField
```

```
  onChange = { (e) => setDetails (e.target.value) }  
  className = { classes.field }  
  label = "Details"  
  color = "primary"  
  variant = "outlined"  
  multiline  
  rows = { 4 }  
  fullWidth  
  required  
  error = { detailsError }  
/>
```

# if ~~error~~ == true so error inbuilt style is applied on TextField.

```
< Button
```

```
  type = "submit"  
  variant = "contained"  
  color = "secondary">  
  Submit </Button>
```

```
</form>
```

## ⑨ Radio Button

Importing  $\rightarrow$  import Radio from '@material-ui/core/Radio'  
import RadioGroup from '@material-ui/core/RadioGroup'  
import { FormControlLabel, makeStyles } from '@material-ui/core'  
import FormControl from '@material-ui/core/FormControl'  
import FormLabel from '@material-ui/core/FormLabel'

### way 1

```
<RadioGroup>  
  <Radio value = "hello" />  
  <Radio value = "goodbye" />  
</RadioGroup>
```

### way 2 (here more functionality) [can give label]

1st make state

```
const [category, setCategory] = useState('todos')
```

```
<FormControl className = {classes.field}> # binds up all form components  
  <FormLabel> Note Category </FormLabel> # gives Labels to Form  
  <RadioGroup value = {category} onChange = {(e) => setCategory(e.target.value)}>  
    <FormControlLabel value = "todos" control = {{<Radio/>}} label = "Todos" />  
    <FormControlLabel value = "money" control = {{<Radio/>}} label = "Money" />  
    <FormControlLabel value = "reminders" control = {{<Radio/>}} label = "Reminders" />  
    <FormControlLabel value = "work" control = {{<Radio/>}} label = "work" />  
  </RadioGroup> # Binds up Radio Buttons  
</FormControl>
```

# FormControlLabel gives label to form components

## 10) Grid System

- Grid is a 12 column system.
- Sizes xs, md, sm, lg.

eg  
import Grid from '@material-ui/core/Grid';  
import Paper from '@material-ui/core/Paper';

<Grid container>

# grid container

<Grid item xs={12} sm={6} md={3}>

# grid item

<Paper> 1 </Paper>

</Grid>

# this props are optional

<Grid item xs={12} sm={6} md={3}>

<Paper> 1 </Paper>

# paper is basic div with some box shadow.

</Grid>

</Grid>

### • in code

<Container>

<Grid container>

{notes.map(note => (

<Grid item key={note.id} xs={12} md={6} lg={4}>

<Paper> {note.title} </Paper>

</Grid>

))}

</Grid>

</Container>



## 11) Card & delete from database

in Note.js (where Grid was defined)

```
const handleDelete = async (id) => {  
  await fetch('http://localhost:8080/notes/' + id, {  
    method: 'DELETE' # deletes from database  
  })  
  const newNotes = notes.filter(note => note.id !== id) # delete from current state  
  setNotes(newNotes); # also whenever state changes  
  # and refreshes the component.  
}
```

⇒ we will make card component separately.

```
{ notes.map(note => (  
  <Grid item key={note.id} xs={12} lg={4} md={6}>  
    <NoteCard note={note} handleDelete={handleDelete} />  
  </Grid>  
))}
```

⇒ in Note Card

```
export default function NoteCard({ note, handleDelete }) {  
  return (  
    <div>  
      <Card elevation={1}> # 1 means base value 8  
      2 = 8 x 2  
      n = 8 x n  
        <Card Header  
          action={(  
            <IconButton onClick={() => handleDelete(note.id)} />  
            <Delete Outlined />  
          )}>  
        </Card Header>  
        <div>  
          title = {note.title}  
          subtitle = {note.category}  
        </div>  
        <Card Content>  
          <Typography variant="body2" color="textSecondary" component="p">  
            {note.details}  
          </Typography>  
        </Card Content>  
      </div>  
    </div>  
  )}
```

## 12) layout

Will wrap for all components  
eg sidebar, footers, headers

make layout.js

- in app.jsx

import Layout & inside Routes do

```
<Router>
  <Layout>
    <Switch>
      <Route exact path="/" > <Notes /> </Route>
      <Route exact path="/create" > <Create /> </Route>
    </Switch>
  </Layout>
</Router>
```

- in Layout.jsx

export default function Layout({children}){

const classes = useStyles();

return (  
 <div className = {classes.root} > display: flex

{ /\* sidebar \*/ }

<Drawer className = {classes.drawer} variant = "permanent"  
 anchor = "left" classes = { } paper = classes.drawerPaper { } >

# overwrite paper default class

<div >

<Typography variant = "h5">

Palash Bajpai

</Typography>

</div>

</Drawer>

<div className = {classes.page} >

{children}

</div>

</div>

}

## • List

### List of Links in sidebar

// import all things

import { useHistory, useLocation } from 'react-router-dom';

inside <Drawer>

{/\* List Links \*/}

<List>

{ menuItems.map (item => {

<ListItem key = {item.tent}>

button

onClick = { () => history.push (item.path) }

className = { location.pathname === item.path ? classes.active : null }

↑  
background: '#f4f4f4'

>

<ListItemIcon> {item.icon} </ListItemIcon>

<ListItemText primary = {item.tent} />

</ListItem>

)) }

</List>

### inside function Layout

const history = useHistory ();

to route betw pages

const location = useLocation ();

to get current location

const classes = useStyles ();

# made useStyles using makeStyles

const MenuItems = [

{

tent : "My Notes",

icon : <SubjectOutlined color = "secondary" />,

path : '/'

},

{

tent : "Create Note",

icon : <AddCircleOutlineOutlined color = "secondary" />,

path : '/create'

}

}

# ★ React Masonry

initial design is



want



- Search `react-masonry-css`  
`npm i react-masonry-css`

→ in `Notes.jsx` (where cards are rendered)

import Masonry from 'react-masonry-css';

`<Container>`

`<Masonry`

`breakpointCols = { breakpoints }`

`className = "my-masonry-grid"`

`columnClassName = "my-masonry-grid-column"`

`>`

`{ notes.map (note => (`

`<div key={note.id} style={{marginBottom: "30px"}}>`

`<NoteCard note={note} handleDelete={handleDelete} />`

`</div>`

`))`

`</Masonry>`

`</Container>`

```
const breakpoints = {  
  default: 3,  
  1100: 2,  
  700: 1,  
}
```

# Also add css given in npm site

Extra

## • Running a json server

add notes as json file in data  
↳ db.json

eg {  
 "notes": [  
 {  
 "title": "Happy Birthday",  
 "details": "Lorem ipsum",  
 "id": 1  
 },  
 --  
 ]  
}

- npm install -g json-server
- json-server --watch data/db.json --port 8000.  
(loads on localhost:8000/notes)

## • Form submit

to update db.json

eg 3 variables title, detail & category needs to be submitted  
id is self deprecated.

```
• if (title && details) {  
  fetch("http://localhost:8000/notes", {  
    method: "POST",  
    headers: { "Content-type": "application/json" },  
    body: JSON.stringify({ title, details, category })  
  }).then(() => history.push('/'))  
}
```

```
# import { useHistory } from 'react-router-dom'  
const history = useHistory(); # inside function  
                                used to redirect
```



## ◦ Fetching from json

```
const [notes, setnotes] = useState([]);
```

```
useEffect () => {
```

```
  fetch ('http://localhost:3004/notes')
```

```
    .then (res => res.json())
```

```
    .then (data => setNotes(data))
```

```
}. [])
```