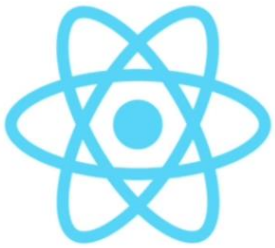
React JS

WHAT IS REACT?

A JavaScript library / framework created and maintained by Facebook that is used for building user interfaces

React gives us a way to build websites & UIs with organized and reusable components



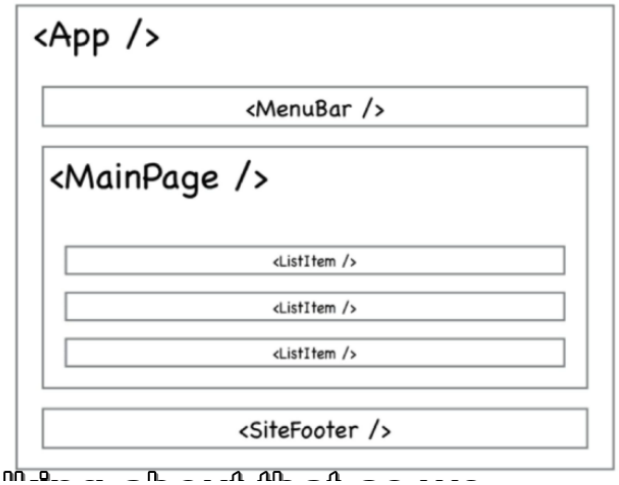
UI COMPONENTS

OUTPUT / HTML (JSX)

LOGIC (JS)

STYLE (CSS)

Optional: Can also be separate



React JS

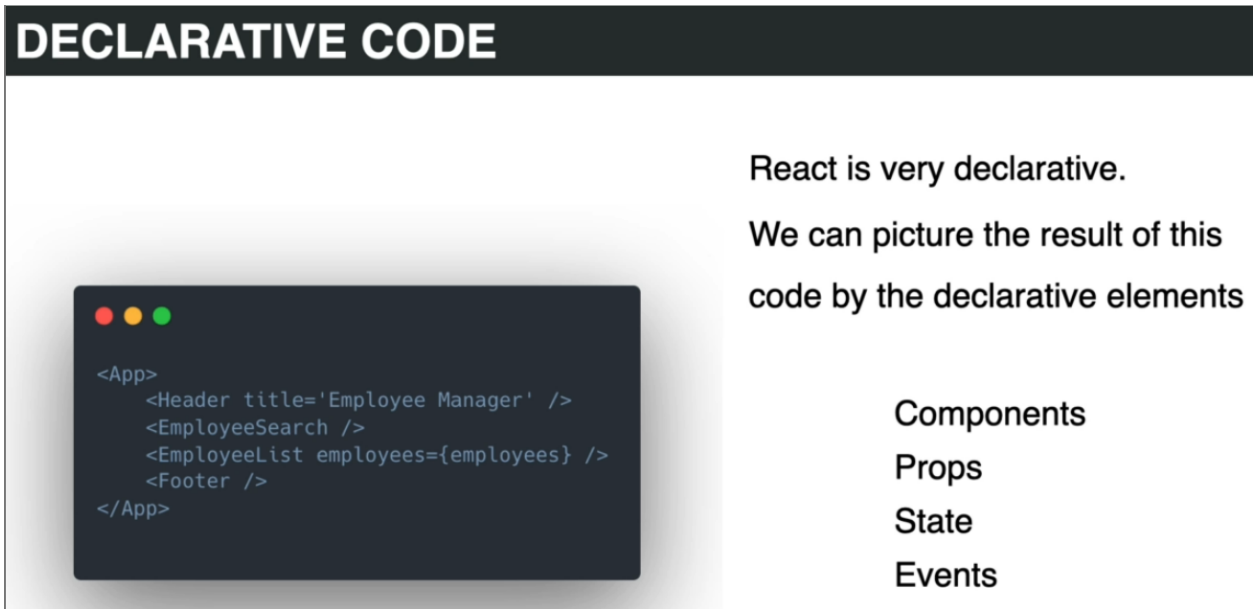
WHAT YOU SHOULD KNOW

- JavaScript Basics - Data structures, loops, functions, etc
- The DOM - Document Object Model
- High Order Array Methods - forEach, map, filter, reduce
- Arrow Functions
- Async Programming & Fetch API
- NPM (Node Package Manager)

WHY LEARN REACT?

- Organization
- Reusable
- Flexibility
- Popularity & Support
- Performance

React JS



3. Environment Setup

4. Code Repos

React JS

2. React Basics & JSX

1. Feedback Project Intro

2. Create React App

npx create-react-app Keep-Note --use-npm

```
C:\Users\Dell\Desktop\Sudip>npx create-react-app keep-note --use-npm

Creating a new React app in C:\Users\Dell\Desktop\Sudip\keep-note.

Installing packages. This might take a couple of minutes.
Installing react, react-dom, and react-scripts with cra-template...


added 1394 packages in 6m

210 packages are looking for funding
  run `npm fund` for details

Initialized a git repository.

Installing template dependencies using npm...

added 56 packages in 31s

210 packages are looking for funding
  run `npm fund` for details
Removing template package using npm...


removed 1 package, and audited 1450 packages in 10s

210 packages are looking for funding
  run `npm fund` for details

6 high severity vulnerabilities

To address all issues (including breaking changes), run:
  npm audit fix --force

Run `npm audit` for details.

Created git commit.

Success! Created keep-note at C:\Users\Dell\Desktop\Sudip\keep-note
```

React JS

```
Success! Created keep-note at C:\Users\Dell\Desktop\Sudip\keep-note
Inside that directory, you can run several commands:

  npm start
    Starts the development server.

  npm run build
    Bundles the app into static files for production.

  npm test
    Starts the test runner.

  npm run eject
    Removes this tool and copies build dependencies, configuration files
    and scripts into the app directory. If you do this, you can't go back!

We suggest that you begin by typing:

  cd keep-note
  npm start

Happy hacking!
```

```
C:\Users\Dell\Desktop\Sudip\keep-note>code .

C:\Users\Dell\Desktop\Sudip\keep-note>
```

```
You can now view keep-note in the browser.

  Local:            http://localhost:3000
  On Your Network:  http://192.168.1.68:3000

Note that the development build is not optimized.
To create a production build, use npm run build.

webpack compiled successfully
```

React JS

3. Initializing React

index.js M | App.js U | index.css M

src > index.js

```
1
2  import React from 'react'
3  import  ReactDOM  from 'react-dom'
4  import './index.css'
5  import App from './App'
6
7  ReactDOM.render(
8  <React.StrictMode>
9  <App />
10 </React.StrictMode>
11 , document.getElementById('root'))
```

src > App.js > default

```
1  function App(){
2      return <h1> Hello form the app component</h1>
3
4  }
5
6  export default App
```

Hello form the app component

4. Intro To JSX

React JS

src > JS App.js > ...

```
function App(){
    return (
        <div className="container">
        <h1> My App</h1>

        </div >
 )
}

export default App



// import React from 'react'

// function App(){
//     return React.createElement(
//         'div',
//          {className: 'container'},
//     React.createElement('h1', {}, 'My App')
//     )
// }
```

JS index.js

```
import React from 'react'
import  ReactDOM  from 'react-dom'
import './index.css'
import App from './App'

ReactDOM.render(
<React.StrictMode>
<App />
</React.StrictMode>
, document.getElementById('root'))
```

React JS

5. Dynamic Values & LIsts in JSX

App.js U index.css M

src > App.js > App > comments.map() callback

```
13  function App(){
14      const title = 'Blog Post'
15      const body = 'This is my blog post'
16      const comments = [
17          {id:1, text: 'Comment one'},
18          {id:2, text: 'Comment two'},
19          {id:3, text: 'Comment three'},
20      ]
21      return (
22          <div className="container">
23          <h1> {title.toUpperCase()}</h1>
24          <p> {body}</p>
25
26          <div className="comments">
27              <h3> Comments ({comments.length})</h3>
28              <ul>
29                  {comments.map((comment, index) => (
30                      <li key={index}>{comment.text}</li>
31                  ))}
32              </ul>
33          </div>
34          </div >
35  )
36  }
37
38  export default App
```

BLOG POST

This is my blog post

Comments (3)

- Comment one
- Comment two
- Comment three

React JS

6. Conditionals in JSX

src > JS App.js > App > commentBlock

```
22      const  loading = false
23      const showComments = true
24
25      if(loading) return <h1> Loading...</h1>
26
27      const commentBlock = (
28          <div className="comments">
29              <h3> Comments ({comments.length})</h3>
30              <ul>
31                  {comments.map((comment, index) => (
32                      <li key={index}>{comment.text}</li>
33                  ))}
34              </ul>
35          </div>
36      )
37      return (
38          <div className="container">
39          <h1> {title.toUpperCase()}</h1>
40          <p> {body}</p>
41
42          {showComments && commentBlock}
43
44
45          </div >
46  )
47  }
48
49  export default App
```

BLOG POST

This is my blog post

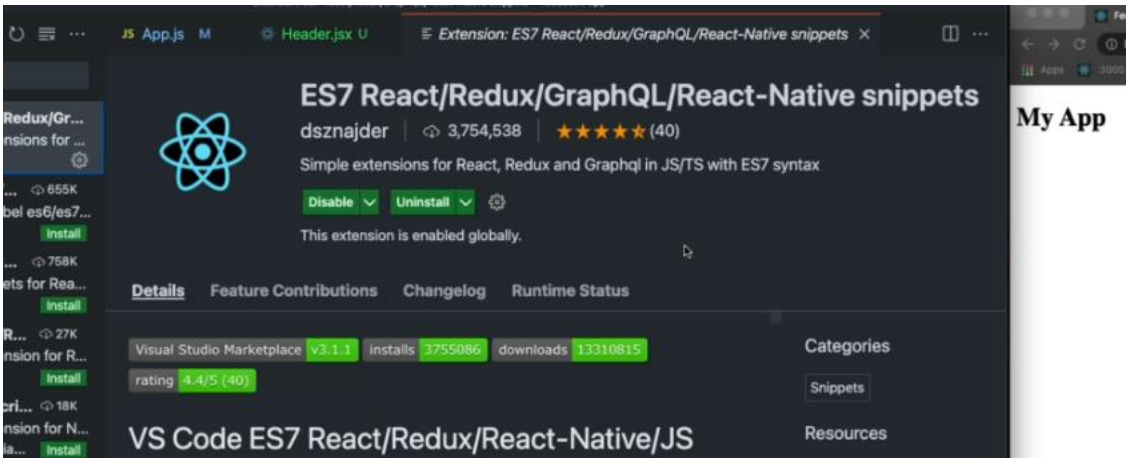
Comments (3)

- Comment one
- Comment two
- Comment three

React JS

3. Components, Props & State

1. Creating Your First Component & Props



rfce

React JS

App.js U Header.jsx U

src > App.js > default

```
56  import Header from "./components/Header"
57
58  function App(){
59      return (
60          <>
61          <Header/>
62          </>
63    )
64  }
65  export default App
66
```

App.js U Header.jsx U

src > components > Header.jsx > default

```
1   import PropTypes from 'prop-types'
2
3   function header({ text}) {
4     return (
5       <header>
6           <div className="container">
7           <h2>{text}</h2>
8
9           </div>
10
11      </header>
12    )
13  }
14
15  header.defaultProps = {
16      text:'Feedback UI',
17  }
18
19  Headers.propTypes = {
20      text:PropTypes.string
21  }
22
23  export default header
```

React JS

2. Adding Styles To A Component

src > App.js > App

```
58  function App(){
59      return (
60          <>
61          <Header bgColor = 'red'  textColor='blue'/>
62          <div className="container">
63              <h1>Keep Notes</h1>
64          </div>
65          </>
66      )
67  }
68  export default App
```

App.js U | Header.jsx U

src > components > Header.jsx > header > constructor > headerStyles

```
1   import PropTypes from 'prop-types'
2
3   function header({ text, bgColor, textColor}) {
4
5       const headerStyles ={
6           backgroundColor: bgColor,
7             color: textColor,
8
9       }
10    return (
11      <header style={headerStyles }>
12  >       <div className="container">···
15          </div>
16
17      </header>
18    )
19  }
```

React JS

App.js U | Header.jsx U

src > App.js > App

```
55
56  import Header from "./components/Header"
57
58  function App(){
59      return (
60          <>
61          <Header />
62          <div className="container">
63              <h1>Keep Notes</h1>
64          </div>
65          </>
66      )
67  }
68  export default App
69
70
```

App.js U | Header.jsx U

src > components > Header.jsx > ...

```
4
5       const headerStyles ={
6           backgroundColor: bgColor,
7             color: textColor,
8
9       }
10    return (
11      <header style={headerStyles }>
12  >       <div className="container">...
15          </div>
16
17      </header>
18    )
19  }
20
21  header.defaultProps = {
22      text:'Feedback UI',
23      bgColor:'rgba(0,0,0,0.4)',
24      textColor:'#ff6a95'
25  }
26
27  Headers.propTypes = {
28      text:PropTypes.string,
29      bgColor: PropTypes.string,
30      textColor:PropTypes.string,
31  }
32
33  export default header
```

React JS

3. State & useState Hook

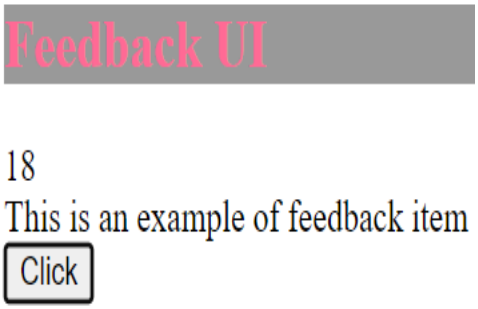
App.js U | FeedbackItem.jsx U

src > components > JS FeedbackItem.jsx > FeedbackItem

```jsx
import {useState} from 'react'

function FeedbackItem() {
    const [rating, setRating] = useState(7)
    const [text, setText] = useState('This is an example of feedback item')
    const handleClick = () =>{
        setRating((prev) => {
            console.log(prev)
            return prev +1
        })
    }
  return (
    <div className="card">
    <div className="num-display">{rating}</div>
    <div className="text-display">{text} </div>
    <button onClick={handleClick}>Click</button>
    </div>
  )
}

export default FeedbackItem
```

src > App.js > default

```jsx
import Header from "./components/Header"
import FeedbackItem from "./components/FeedbackItem"

function App(){
    return (
        <>
        <Header />
        <div className="container">
            <FeedbackItem/>
        </div>
        </>
    )
}
export default App
```



React JS

4. Managing Global State

src > components > JS FeedbackItem.jsx > FeedbackItem

```
12      function FeedbackItem({item}) {
13    return (
14      <div className="card">
15      <div className="num-display">{item.rating}</div>
16      <div className="text-display">{item.text} </div>
17      {/* <button onClick={handleClick}>Click</button> */}
18      </div>
19    )
20  }
21
22  export default FeedbackItem
```

src > JS App.js > ...

```
54
55  import { useState } from 'react'
56  import Header from "./components/Header"
57  import FeedbackData from './data/FeedbackData'
58  import FeedbackList from './components/FeedbackList'
59  function App(){
60
61      const [feedback, setFeedback] = useState(FeedbackData)
62      return (
63          <>
64          <Header />
65          <div className="container">
66              <FeedbackList feedback={feedback}/>
67
68          </div>
69          </>
70   )
71  }
72  export default App
```

React JS

src > components > JS FeedbackList.jsx > ...

```jsx
import FeedbackItem from "./FeedbackItem"

function FeedbackList({feedback}) {
    if(!feedback || feedback.length ===0){
        return <p> No Feedback Yet</p>
    }
  return (
    <div className="feedback-list">
    {feedback.map((item) => (
        <FeedbackItem key= {item.id} item = {item}/>
    ))}
    </div>
  )
}

export default FeedbackList
```

5. Card Component & Conditional Styles

src > components > shared > JS Card.jsx > Card

```jsx
function Card({children, reverse}) {
  return (
    <div className={`card ${reverse && 'reverse'}`}>
        {children}
    </div>
  )
}
```

src > components > JS FeedbackItem.jsx > FeedbackItem

```jsx
import PropTypes from 'prop-types'
```

React JS

src > components > JS FeedbackList.jsx > FeedbackList

```
18   FeedbackList.propTypes ={
19       feedback: PropTypes.arrayOf(
20           PropTypes.shape({
21               id: PropTypes.number.isRequired,
22               text:PropTypes.string.isRequired,
23               rating: PropTypes.number.isRequired,
24           })
25       ),
26   }
```

src > components > shared > JS Card.jsx > Card

```
11   import PropTypes from 'prop-types'
12   function Card({children, reverse}) {
13       return (
14         <div
15         className= 'card'
16         style={{
17           backgroundColor:reverse ? 'rgba(2, 2, 42, 0.9)' : '#fff',
18           color: reverse ? '#fff' :'#000',
19         }}
20         >
21             {children}
22         </div>
23       )
24     }
25
26     Card.defaultProps ={
27       reverse:true,
28     }
29
30   Card.propTypes = {
31       children: PropTypes.node.isRequired,
32       reverse: PropTypes.bool,
33   }
34
35   export default Card
```

React JS

6. Events & Prop Drilling

src > App.js > App > deleteFeedback

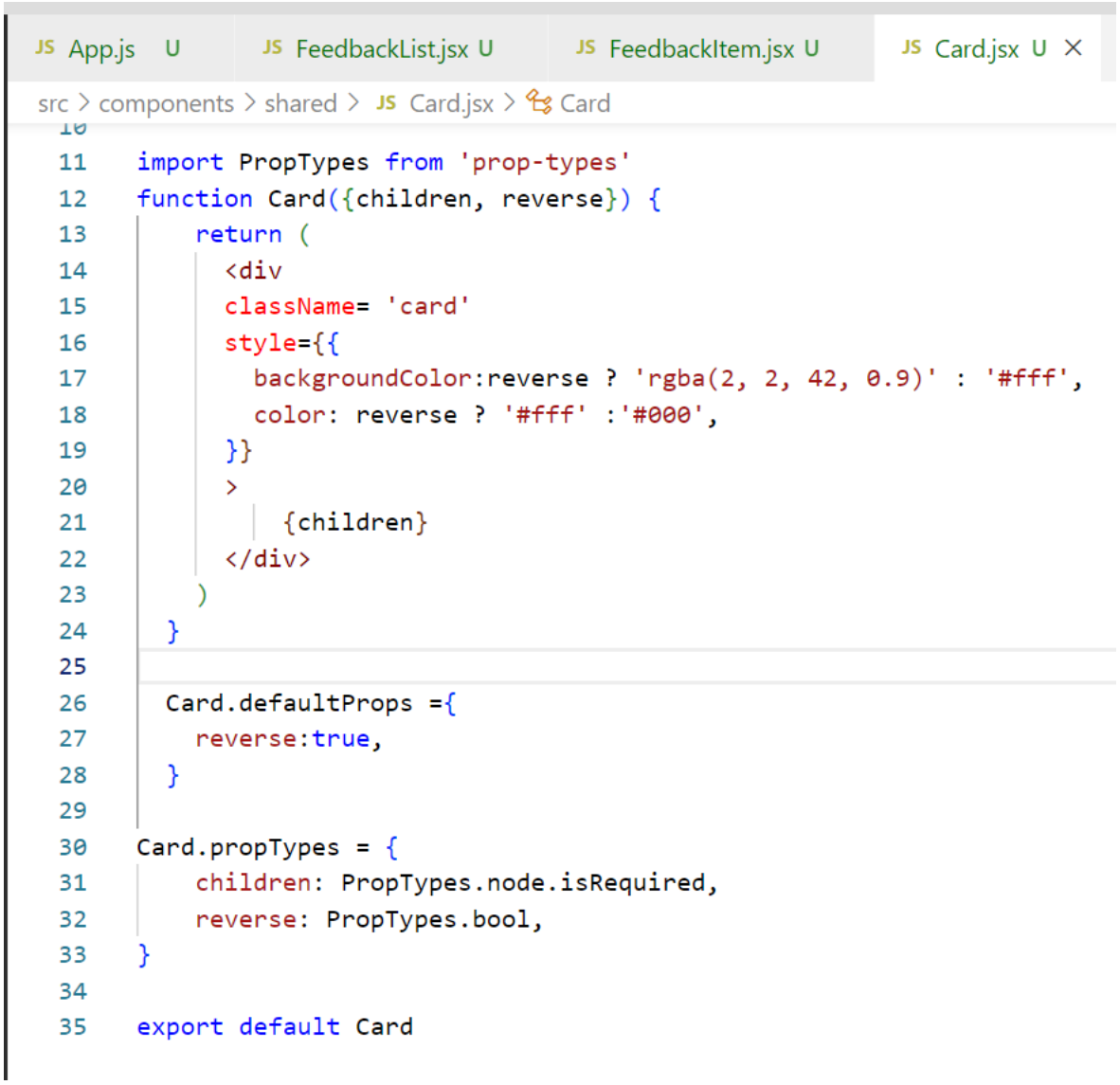
```
60  function App(){
61
62      const [feedback, setFeedback] = useState(FeedbackData)
63
64      const deleteFeedback = (id) => {
65          console.log(Object);
66      }
67
```

App.js U | FeedbackList.jsx U | FeedbackItem.jsx U | Card.jsx U

src > App.js > App > deleteFeedback

```
55  import { useState } from 'react'
56  import Header from "./components/Header"
57  import FeedbackData from './data/FeedbackData'
58  import FeedbackList from './components/FeedbackList'
59  import Card from "./components/shared/Card"
60  function App(){
61
62      const [feedback, setFeedback] = useState(FeedbackData)
63
64      const deleteFeedback = (id) => {
65          if(window.confirm('Are you sure you want to delete?')){
66              setFeedback(feedback.filter((item) => item.id !== id))
67          }
68      }
69
70
71      return (
72          <>
73          <Header />
74          <div className="container">
75              <FeedbackList feedback={feedback}
76                  handleDelete = {deleteFeedback}
77              />
78              <Card>
79                  Hello
80              </Card>
81
82          </div>
```

React JS


App.js U
FeedbackList.jsx U
FeedbackItem.jsx U
Card.jsx U
src > components > shared > Card.jsx > Card
import PropTypes from 'prop-types'
function Card({children, reverse}) {
return (
<div
className= 'card'
style={{
backgroundColor:reverse ? 'rgba(2, 2, 42, 0.9)' : '#fff',
color: reverse ? '#fff' :'#000',
}}
>
{children}

)
}
Card.defaultProps ={
reverse:true,
}
Card.propTypes = {
children: PropTypes.node.isRequired,
reverse: PropTypes.bool,
}
export default Card

React JS

App.js U | FeedbackList.jsx U | FeedbackItem.jsx U | Card.jsx U

src > components > FeedbackItem.jsx > FeedbackItem > constructor

```
12  import {FaTimes} from 'react-icons/fa'
13  import PropTypes from 'prop-types'
14  import Card from './shared/Card'
15      function FeedbackItem({item, handleDelete}) {
16
17
18    return (
19      <Card>
20      <div className="num-display">{item.rating}</div>
21      <button onClick={() => handleDelete(item.id)} className='close'>
22        <FaTimes color='purple'/>
23      </button>
24      <div className="text-display">{item.text} </div>
25      {/* <button onClick={handleClick}>Click</button> */}
26      </Card>
27    )
28  }
29
30  FeedbackItem.propTypes ={
31    item: PropTypes.object.isRequired
32  }
33  export default FeedbackItem
```

React JS

App.js U | FeedbackList.jsx U | FeedbackItem.jsx U

src > components > FeedbackList.jsx > FeedbackList > constructor

```
import PropTypes from 'prop-types'
import FeedbackItem from "./FeedbackItem"

function FeedbackList({feedback, handleDelete}) {
    if(!feedback || feedback.length ===0){
        return <p> No Feedback Yet</p>
    }
  return (
    <div className="feedback-list">
    {feedback.map((item) => (
        <FeedbackItem
        key= {item.id}
        item = {item}
        handleDelete={handleDelete}
        />
    ))}
    </div>
  )
}

FeedbackList.propTypes ={
    feedback: PropTypes.arrayOf(
        PropTypes.shape({
            id: PropTypes.number.isRequired,
            text:PropTypes.string.isRequired,
            rating: PropTypes.number.isRequired,
        })
    ),
}
```

React JS

7. FeedbackStats Component & Reactivity

App.js U | FeedbackStats.jsx U

src > components > FeedbackStats.jsx > FeedbackStats

```jsx
import PropTypes from 'prop-types'

function FeedbackStats({feedback}) {
// Calculate rating Average

let average = feedback.reduce((acc, cur) =>{
    return acc +cur.rating
},0) / feedback.length

average = average.toFixed(1).replace(/[.,]0$/, '')

  return (
    <div className="feedback-stats">
        <h4>{feedback.length} Reviews</h4>
        <h4> Average Rating: {isNaN(average) ? 0 : average}</h4>
    </div>
  )
}

FeedbackStats.propTypes = {
    feedback: PropTypes.array.isRequired,
}

export default FeedbackStats
```

React JS

4. Forms, Validation & Simple Animation

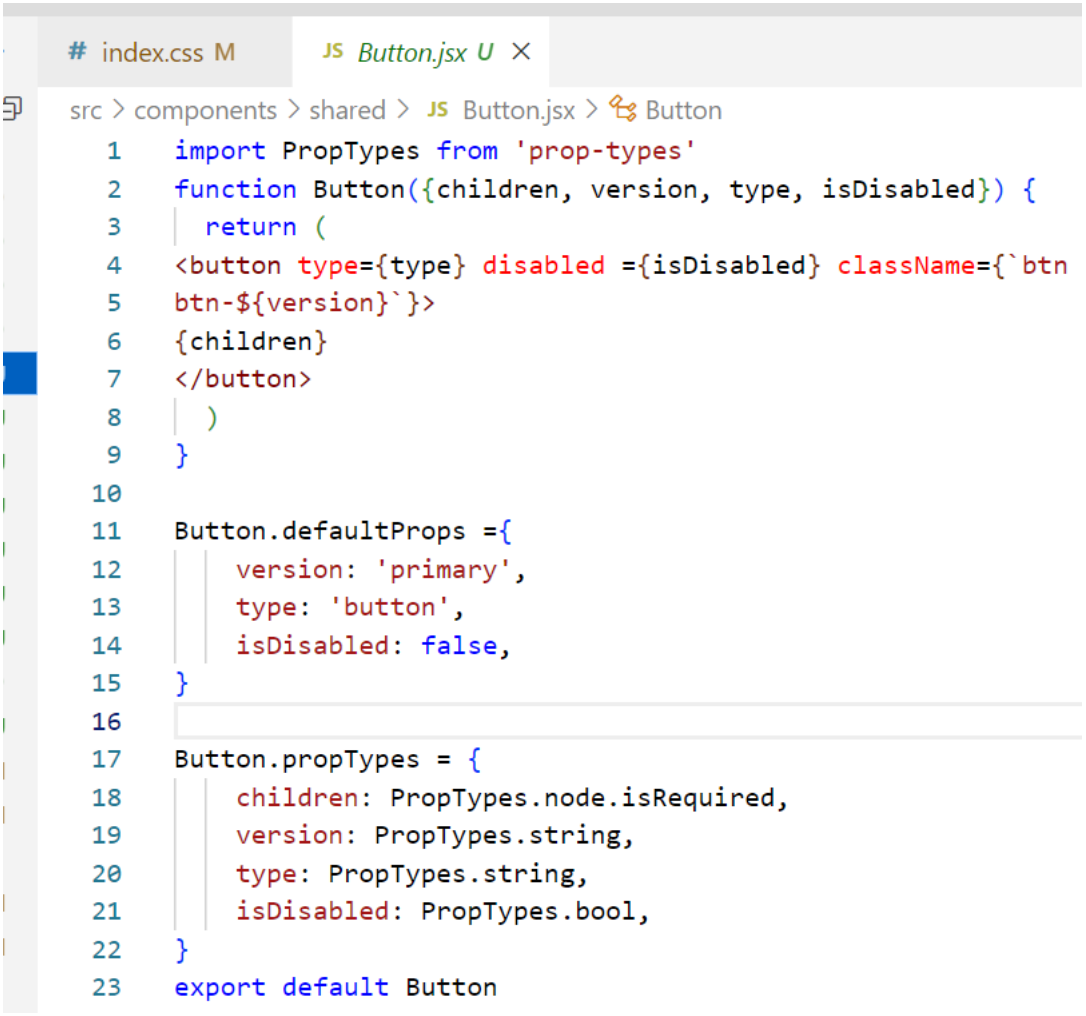
1. Form Input & State

App.js U | FeedbackStats.jsx U | FeedbackForm.jsx U

src > components > FeedbackForm.jsx > FeedbackForm

```jsx
import { useState } from "react"
import Card from "./shared/Card"
function FeedbackForm() {

    const [text, setText] = useState('')
    const handleTextChange= (e) => {
        setText(e.target.value)
    }
  return (
    <Card>
    <form>
        <h2>How Would you rate your service with us?</h2>
        {/* @todo - rating select component  */}
        <div className="input-group">
            <input
            onChange={handleTextChange}
            type= 'text'
            placeholder="Write a review"
                value = {text}
            />
            <button type="submit">Send</button>
        </div>
    </form>
    </Card>
  )
}
export default FeedbackForm
```

2. Custom Button Component



3. Real-Time Validation

JS FeedbackForm.jsx U ×

src > components > JS FeedbackForm.jsx > FeedbackForm > handleTextChange

```
1   import { useState } from "react"
2   import Card from "./shared/Card"
3   import Button from "./shared/Button"
4   function FeedbackForm() {
5
6       const [text, setText] = useState('')
7       const [btnDisabled, setBtnDisabled] = useState(true)
8       const [message, setMessage] =useState('')
9       const handleTextChange= (e) => {
10          if(text === ''){
11              setBtnDisabled(true)
12              setMessage(null)
13          } else if(text !== '' && text.trim().length <= 10){
14              setMessage('Text must be at least 10 characters')
15              setBtnDisabled(true)
16          } else {
17              setMessage(null)
18              setBtnDisabled(false)
19          }
20          setText(e.target.value)
21      }
22    return (
```

4. Rating Select Component

5. Add Feedback

6. Fade Animation With Framer Motion

React JS

React JS

React JS

React JS