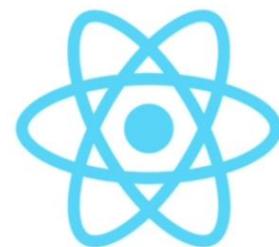


React JS

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

<App />

<MenuBar />

<MainPage />

<ListItem />

<ListItem />

<ListItem />

<SiteFooter />

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

DECLARATIVE CODE

```
<App>
  <Header title='Employee Manager' />
  <EmployeeSearch />
  <EmployeeList employees={employees} />
  <Footer />
</App>
```

React is very declarative.

We can picture the result of this code by the declarative elements

Components

Props

State

Events

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

The screenshot shows a code editor with two tabs at the top: 'index.js' and 'App.js'. The 'index.js' tab is active. Below the tabs, there are two code snippets.

index.js:

```
src > JS 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')
```

App.js:

```
src > JS App.js > [e] 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 > ...
1  function App(){
2      return (
3          <div className="container">
4              <h1> My App</h1>
5
6          </div >
7      )
8  }
9
10 export default App
11
12
13
14 // import React from 'react'
15
16 // function App(){
17 //     return React.createElement(
18 //         'div',
19 //         {className: 'container'},
20 //         React.createElement('h1', {}, 'My App')
21 //     )
22 // }
23
```

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

```
JS App.js U X # index.css M
src > JS 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
39
```

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 > [e] 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

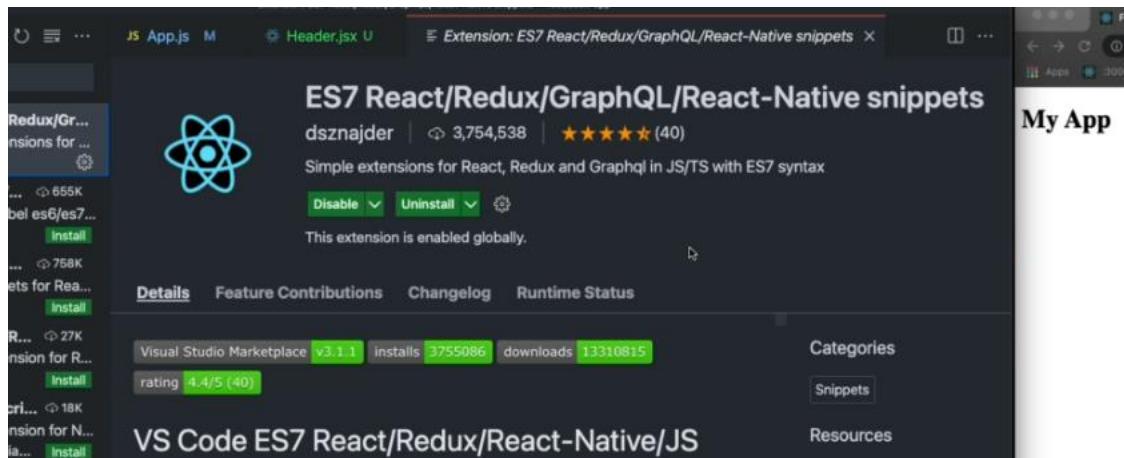
Comments (3)

- Comment one
- Comment two
- Comment three

React JS

3. Components, Props & State

1. Creating Your First Component & Props



rfce

```
JS App.js U # index.css M JS Header.jsx U
src > components > JS Header.jsx > ⚭ header
1 import React from 'react'
2
3 function header() {
4   return (
5     <div>header</div>
6   )
7 }
8
9 export default header
```

React JS

```
JS App.js U X JS Header.jsx U
src > JS 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
```

```
JS App.js U JS Header.jsx U X
src > components > JS 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 > JS 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
```

```
src > components > JS Header.jsx > header > constructor > headerStyles
JS App.js U JS Header.jsx U X
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">...
13            </div>
14
15        </header>
16    )
17
18
19 }
```

React JS

The screenshot shows a code editor interface with two tabs: "JS App.js" and "JS Header.js". The "Header.js" tab is active, displaying the following code:

```
JS App.js U X JS Header.js U
src > JS 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
```

The "Header.js" file contains the following code:

```
JS App.js U JS Header.js U X
src > components > JS Header.js > ...
4
5     const headerStyles ={
6         backgroundColor: bgColor,
7         color: textColor,
8
9     }
10    return (
11        <header style={headerStyles}>
12            <div className="container">...
13            </div>
14
15        </header>
16    )
17
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

```
JS App.js U JS FeedbackItem.jsx U X
src > components > JS FeedbackItem.jsx > ⚒ FeedbackItem
1  import {useState} from 'react'
2
3  function FeedbackItem() {
4      const [rating, setRating] = useState(7)
5      const [text, setText] = useState('This is an example of feedback item')
6      const handleClick = () =>{
7          setRating((prev) => {
8              console.log(prev)
9              return prev +1
10         })
11     }
12     return (
13         <div className="card">
14             <div className="num-display">{rating}</div>
15             <div className="text-display">{text}</div>
16             <button onClick={handleClick}>Click</button>
17         </div>
18     )
19 }
20
21 export default FeedbackItem
```

```
src > JS App.js > ⚒ default
54  import Header from "./components/Header"
55  import FeedbackItem from "./components/FeedbackItem"
56
57  function App(){
58      return (
59          <>
60          <Header />
61          <div className="container">
62              <FeedbackItem/>
63          </div>
64          </>
65      )
66  }
67  export default App
```

Feedback UI

18

This is an example of feedback item

Click

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
73
```

React JS

```
src > components > JS FeedbackList.jsx > ...
1   import FeedbackItem from "./FeedbackItem"
2
3   function FeedbackList({feedback}) {
4     if(!feedback || feedback.length ===0){
5       return <p> No Feedback Yet</p>
6     }
7     return (
8       <div className="feedback-list">
9         {feedback.map((item) => (
10           <FeedbackItem key={item.id} item={item}/>
11         )))
12       </div>
13     )
14   }
15
16 export default FeedbackList
```

5. Card Component & Conditional Styles



The screenshot shows a code editor with tabs for FeedbackList.jsx, Card.jsx, App.jsx, and Feedback. The Card.jsx tab is active, displaying the following code:

```
src > components > shared > JS Card.jsx > Card
1
2   function Card({children, reverse}) {
3     return (
4       <div className={`${'card '}${reverse ? 'reverse'}`}>
5         {children}
6       </div>
7     )
8   }
```

src > components > **JS** FeedbackItem.jsx > **FeedbackItem**

```
12   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 > JS 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

JS App.js U X JS FeedbackList.jsx U JS FeedbackItem.jsx U JS Card.jsx U

src > JS 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          </div>
82      
```

React JS

The screenshot shows a code editor interface with several tabs at the top: App.js, FeedbackList.jsx, FeedbackItem.jsx, and Card.jsx (which is the active tab). Below the tabs, a breadcrumb navigation bar indicates the file path: src > components > shared > Card.jsx. The main area displays the source code for the Card component:

```
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

```
JS App.js U      JS FeedbackList.jsx U      JS FeedbackItem.jsx U X      JS Card.jsx U
src > components > JS 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

The screenshot shows a code editor interface with the following details:

- File Tabs:** JS App.js (U), JS FeedbackList.jsx (U), JS FeedbackItem.jsx (U)
- Breadcrumbs:** src > components > JS FeedbackList.jsx > FeedbackList > constructor
- Code Content:** The code is for a `FeedbackList` component. It imports `PropTypes` and `FeedbackItem`. The component checks if `feedback` is empty and returns a message. Otherwise, it maps over `feedback` to render `FeedbackItem` components. It also defines `FeedbackList.propTypes` with `feedback` being an array of objects with `id`, `text`, and `rating` properties.

```
2 import PropTypes from 'prop-types'
3 import FeedbackItem from "./FeedbackItem"
4
5 function FeedbackList({feedback, handleDelete}) {
6     if(!feedback || feedback.length ===0){
7         return <p> No Feedback Yet</p>
8     }
9     return (
10        <div className="feedback-list">
11            {feedback.map((item) => (
12                <FeedbackItem
13                    key= {item.id}
14                    item = {item}
15                    handleDelete={handleDelete}
16                />
17            ))}
18        </div>
19    )
20 }
21
22 FeedbackList.propTypes ={
23     feedback: PropTypes.arrayOf(
24         PropTypes.shape({
25             id: PropTypes.number.isRequired,
26             text:PropTypes.string.isRequired,
27             rating: PropTypes.number.isRequired,
28         })
29     ),
30 }
```

React JS

7. FeedbackStats Component & Reactivity



```
JS App.js U      JS FeedbackStats.jsx U X
src > components > JS FeedbackStats.jsx > FeedbackStats
1 import PropTypes from 'prop-types'
2
3 function FeedbackStats({feedback}) {
4 // Calculate rating Average
5
6 let average = feedback.reduce((acc, cur) =>{
7   return acc +cur.rating
8 },0) / feedback.length
9
10 average = average.toFixed(1).replace(/[.,]0$/,'')
11
12 return (
13   <div className="feedback-stats">
14     <h4>{feedback.length} Reviews</h4>
15     <h4> Average Rating: {isNaN(average) ? 0 : average}</h4>
16   </div>
17 )
18 }
19
20 FeedbackStats.propTypes = {
21   feedback: PropTypes.array.isRequired,
22 }
23
24 export default FeedbackStats
```

React JS

4. Forms, Validation & Simple Animation

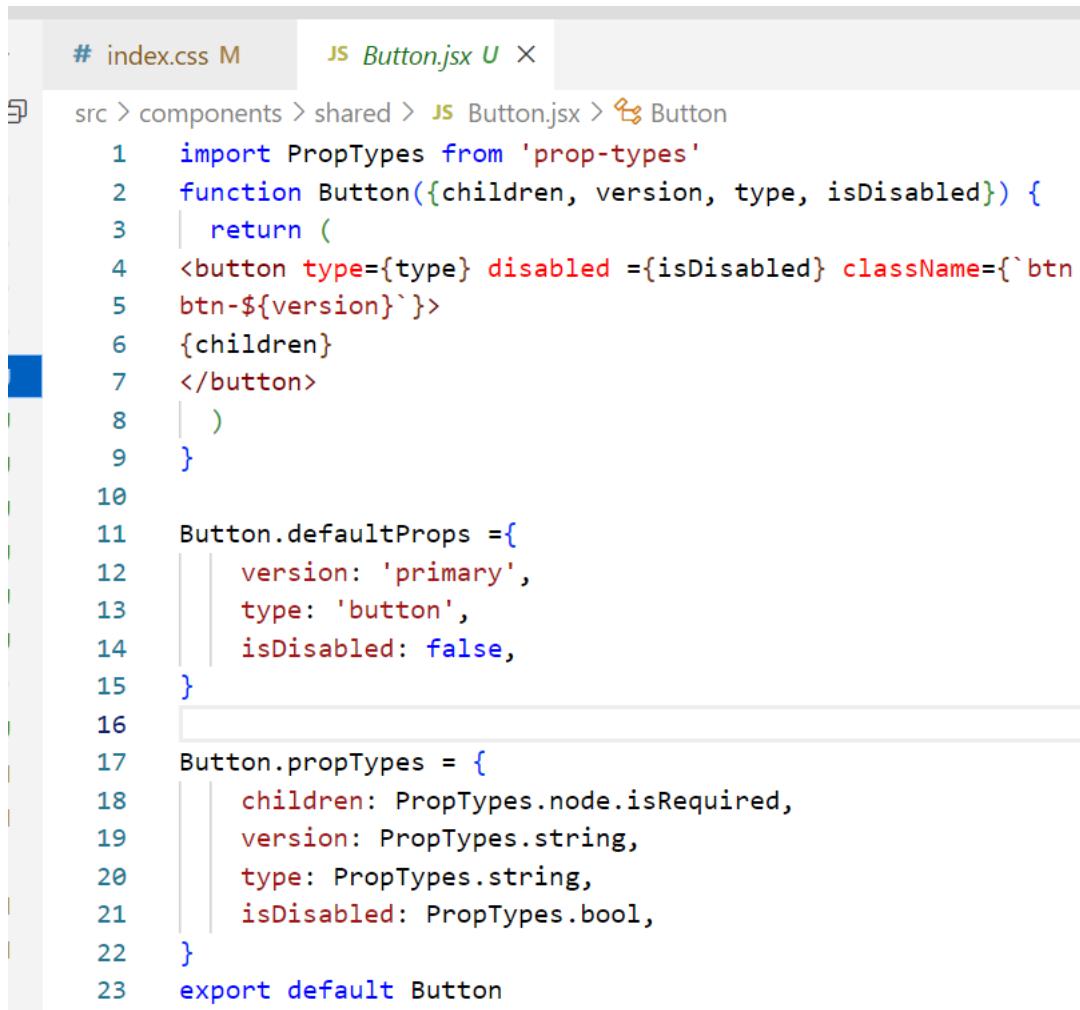
1. Form Input & State



```
JS App.js U JS FeedbackStats.jsx U JS FeedbackForm.jsx U ●
src > components > JS FeedbackForm.jsx > FeedbackForm
1 import { useState } from "react"
2 import Card from "./shared/Card"
3 function FeedbackForm() {
4
5     const [text, setText] = useState('')
6     const handleTextChange= (e) => {
7         setText(e.target.value)
8     }
9     return (
10        <Card>
11            <form>
12                <h2>How Would you rate your service with us?</h2>
13                {/* @todo - rating select component */}
14                <div className="input-group">
15                    <input
16                        onChange={handleTextChange}
17                        type= 'text'
18                        placeholder="Write a review"
19                        value = {[text]}
20                    />
21                    <button type="submit">Send</button>
22                </div>
23            </form>
24        </Card>
25    )
26 }
27 export default FeedbackForm
```

React JS

2. Custom Button Component



The screenshot shows a code editor with a tab bar at the top. The active tab is 'JS Button.jsx U X'. To its left is '# index.css M'. Below the tabs, a breadcrumb navigation bar shows the file path: 'src > components > shared > JS Button.jsx > 📁 Button'. The main area contains the following code:

```
1 import PropTypes from 'prop-types'
2 function Button({children, version, type, isDisabled}) {
3   return (
4     <button type={type} disabled ={isDisabled} className={`btn
5       btn-${version}`}
6     {children}
7     </button>
8   )
9 }
10
11 Button.defaultProps = {
12   version: 'primary',
13   type: 'button',
14   isDisabled: false,
15 }
16
17 Button.propTypes = {
18   children: PropTypes.node.isRequired,
19   version: PropTypes.string,
20   type: PropTypes.string,
21   isDisabled: PropTypes.bool,
22 }
23 export default Button
```

React JS

3. Real-Time Validation

```
JS FeedbackForm.jsx U X
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 (

```

React JS

4. Rating Select Component

The image shows a code editor interface with two tabs at the top: "JS FeedbackForm.jsx U" and "JS RatingSelect.jsx U X". The "RatingSelect.jsx" tab is active, showing its content. Below the tabs, a breadcrumb navigation bar indicates the file structure: "src > components > JS RatingSelect.jsx > ⚙ RatingSelect".

The "RatingSelect.jsx" code is as follows:

```
1 import {useState} from 'react'
2 function RatingSelect({select}) {
3
4     const [selected, setSelected] = useState(10);
5     const handleChange = (e) =>{
6         setSelected(+e.currentTarget.value)
7         select(+e.currentTarget.value)
8     }
9     return (
10        <ul className='rating'>
11            <li>
12                <input
13                    type='radio'
14                    id='num1'
15                    name='rating'
16                    value='1'
17                    onChange={handleChange}
18                    checked={selected ===1}
19                />
20                <label htmlFor='num1'>1</label>
21            </li>
22        </ul>
23    )
24}
```

The "FeedbackForm.jsx" code is as follows:

```
1 <Form>
2     <RatingSelect select={rating}>
3     </RatingSelect>
4 </Form>
```

The "RatingSelect.jsx" code is identical to the one shown above, indicating it is the source of the component used in "FeedbackForm.jsx".

React JS

The screenshot shows a code editor interface with two tabs at the top: "JS FeedbackForm.jsx U" and "JS RatingSelect.jsx U X". The "RatingSelect.jsx" tab is active. Below the tabs, the file path is displayed as "src > components > JS RatingSelect.jsx > RatingSelect". The main content area contains the following code:

```
48     <li>
49         <input
50             type='radio'
51             id='num4'
52             name='rating'
53             value='4'
54             onChange={handleChange}
55             checked={selected ===4}
56         />
57         <label htmlFor='num4'>4</label>
58     </li>
59
60     <li>
61         <input
62             type='radio'
63             id='num5'
64             name='rating'
65             value='5'
66             onChange={handleChange}
67             checked={selected ===5}
68         />
69         <label htmlFor='num5'>5</label>
70     </li>
71     <li>
72         <input
73             type = 'radio'
74             id='num6'
75             name='rating'
76             value = '6'
```

React JS

The screenshot shows a code editor interface with two tabs at the top: "JS FeedbackForm.jsx U" and "JS RatingSelect.jsx U X". The "RatingSelect.jsx" tab is active. Below the tabs, a breadcrumb navigation bar shows the path: "src > components > JS RatingSelect.jsx > RatingSelect". The main area displays the source code for the "RatingSelect" component. The code uses JSX syntax to render a list of radio buttons for ratings 6, 7, and 8. Each radio button is controlled by a state variable "selected" and has an "onChange" event handler named "handleChange". The radio buttons are labeled with their respective values (6, 7, 8) and have "htmlFor" attributes set to "num6", "num7", and "num8" respectively.

```
JS FeedbackForm.jsx U JS RatingSelect.jsx U X
src > components > JS RatingSelect.jsx > RatingSelect

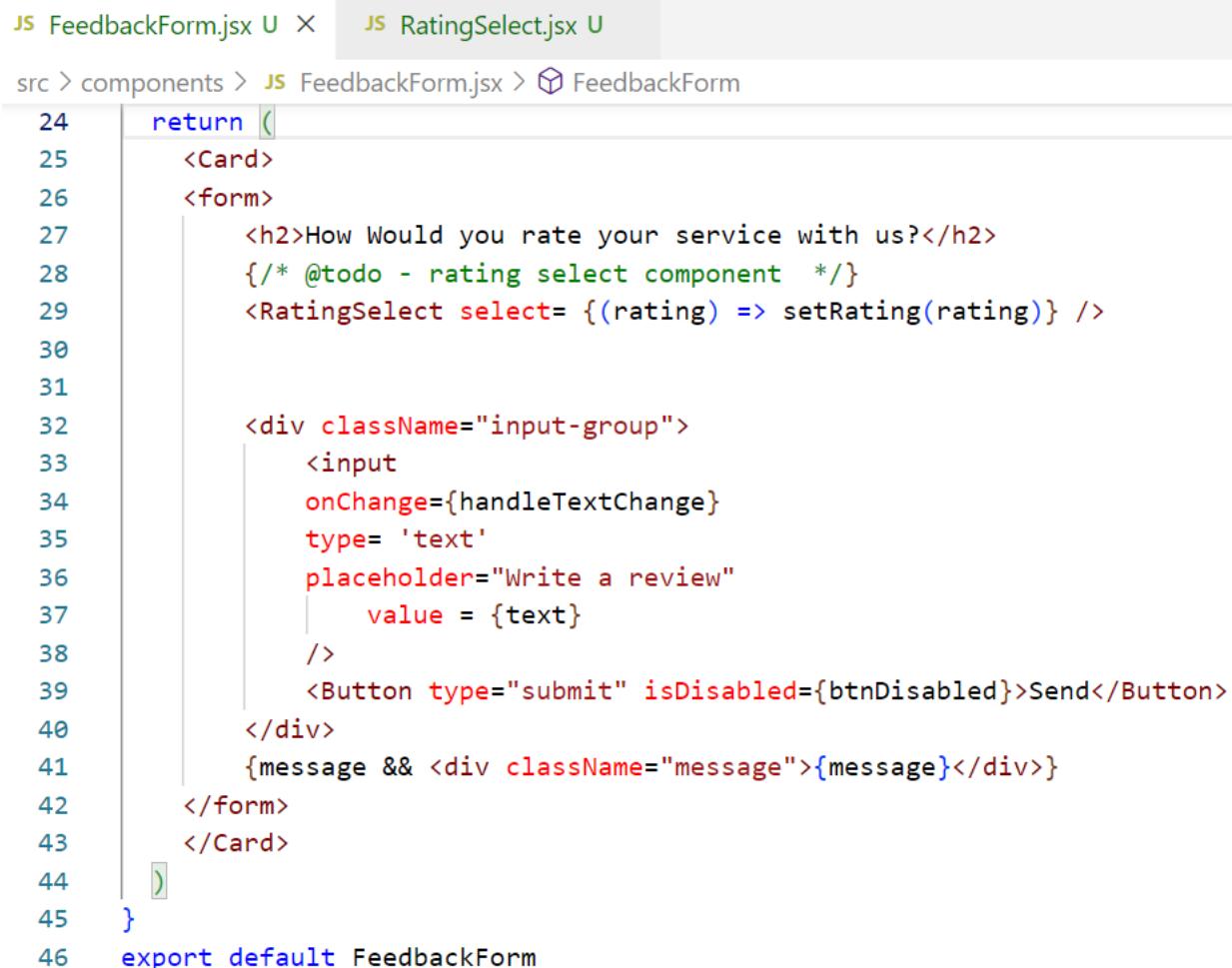
76     value = '6'
77     onChange={handleChange}
78     checked={selected === 6}
79   />
80   <label htmlFor='num6>6</label>
81 </li>
82
83   <li>
84     <input
85       type = 'radio'
86       id='num7'
87       name='rating'
88       value = '7'
89       onChange={handleChange}
90       checked={selected === 7}
91     />
92     <label htmlFor='num7>7</label>
93   </li>
94
95   <li>
96     <input
97       type = 'radio'
98       id='num8'
99       name='rating'
100      value = '8'
101      onChange={handleChange}
102      checked={selected === 8}
103    />
104    <label htmlFor='num8>8</label>
```

React JS

The screenshot shows a code editor interface with two tabs at the top: "FeedbackForm.jsx" and "RatingSelect.jsx". The "RatingSelect.jsx" tab is currently selected. Below the tabs is a breadcrumb navigation bar showing the file structure: "src > components > RatingSelect.jsx". The main area displays the source code for the "RatingSelect" component.

```
JS FeedbackForm.jsx U JS RatingSelect.jsx U X
src > components > JS RatingSelect.jsx > RatingSelect
101   onChange={handleChange}
102   checked={selected === 8}
103   />
104   <label htmlFor='num8>8</label>
105   </li>
106
107   <li>
108     <input
109       type = 'radio'
110       id='num10'
111       name='rating'
112       value ='10'
113       onChange={handleChange}
114       checked={selected === 10}
115     />
116     <label htmlFor='num10>10</label>
117   </li>
118   </ul>
119
120
121 )
122 }
123
124 export default RatingSelect
```

React JS



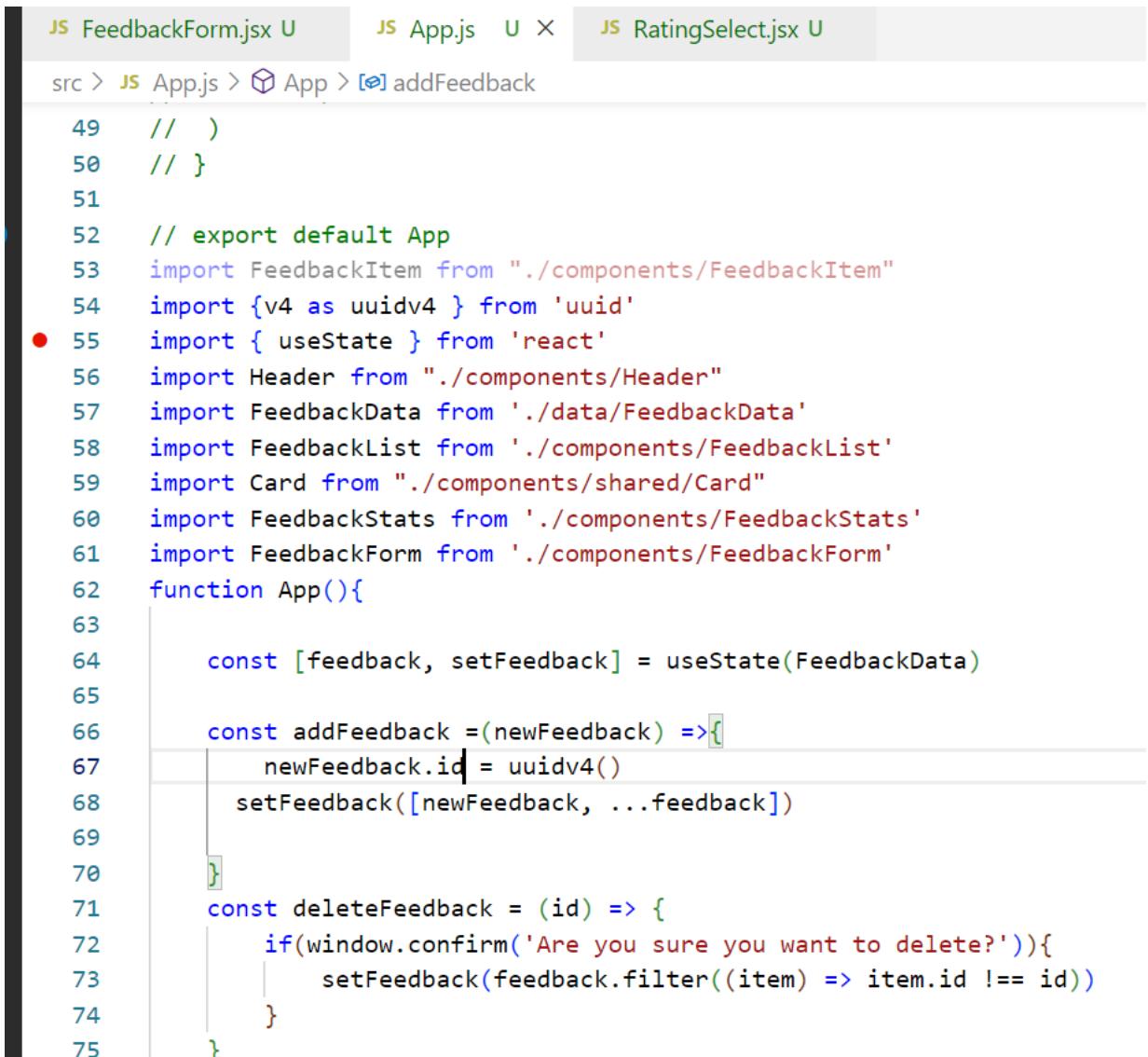
JS FeedbackForm.jsx U X JS RatingSelect.jsx U

src > components > JS FeedbackForm.jsx > FeedbackForm

```
24  return (
25    <Card>
26      <form>
27        <h2>How Would you rate your service with us?</h2>
28        {/* @todo - rating select component */}
29        <RatingSelect select= { (rating) => setRating(rating)} />
30
31
32        <div className="input-group">
33          <input
34            onChange={handleTextChange}
35            type= 'text'
36            placeholder="Write a review"
37            value = {text}
38          />
39          <Button type="submit" isDisabled={btnDisabled}>Send</Button>
40        </div>
41        {message && <div className="message">{message}</div>}
42      </form>
43    </Card>
44  )
45
46  export default FeedbackForm
```

React JS

5. Add Feedback



The screenshot shows a code editor with three tabs at the top: "FeedbackForm.jsx U", "App.js U X", and "RatingSelect.jsx U". The "App.js" tab is active. Below the tabs, the file structure is shown: "src > App.js > App > addFeedback". The code is as follows:

```
JS FeedbackForm.jsx U   JS App.js U X   JS RatingSelect.jsx U
src > JS App.js > App > addFeedback

49  // )
50  // }
51
52  // export default App
53  import FeedbackItem from './components/FeedbackItem'
54  import {v4 as uuidv4 } from 'uuid'
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  import FeedbackStats from './components/FeedbackStats'
61  import FeedbackForm from './components/FeedbackForm'
62  function App(){
63
64      const [feedback, setFeedback] = useState(FeedbackData)
65
66      const addFeedback =(newFeedback) =>{
67          newFeedback.id = uuidv4()
68          setFeedback([newFeedback, ...feedback])
69      }
70  }
71  const deleteFeedback = (id) => {
72      if(window.confirm('Are you sure you want to delete?')){
73          setFeedback(feedback.filter(item) => item.id !== id)
74      }
75  }
```

A red dot is placed next to the line "import { useState } from 'react'".

React JS

The screenshot shows a code editor interface with three tabs at the top: 'FeedbackForm.jsx U', 'App.js U X', and 'RatingSelect.jsx U'. The 'App.js' tab is active. Below the tabs, a breadcrumb navigation bar shows 'src > App.js > ...'. The main area displays the code for the 'App' component:

```
77
78     return (
79       <>
80         <Header />
81         <div className="container">
82           <FeedbackForm handleAdd={addFeedback} />
83           <FeedbackStats feedback={feedback}/>
84
85           <FeedbackList feedback={feedback}>
86             |   handleDelete = {deleteFeedback}
87             |   />
88             <Card>
89               |   Hello
90             </Card>
91
92           </div>
93         </>
94       )
95     }
96   export default App
97 
```

React JS

```
JS FeedbackForm.jsx U X JS App.js U JS RatingSelect.jsx U
src > components > JS FeedbackForm.jsx > FeedbackForm > [e] handleSubmit
10  const [message, setMessage] = useState('')
11  const handleTextChange = (e) => {
12    if(text === ''){
13      setBtnDisabled(true)
14      setMessage(null)
15    } else if(text !== '' && text.trim().length <= 10){
16      setMessage('Text must be at least 10 characters')
17      setBtnDisabled(true)
18    } else {
19      setMessage(null)
20      setBtnDisabled(false)
21    }
22    setText(e.target.value)
23  }
24
25  const handleSubmit = (e) => {
26    e.preventDefault()
27    if(text.trim().length > 10 ) {
28      const newFeedback = {
29        text,
30        rating,
31      }
32
33      handleAdd(newFeedback)
34
35      setText('')
36    }
37  }
38
return (

```

6. Fade Animation With Framer Motion

React JS

5. Creating Routes & Links

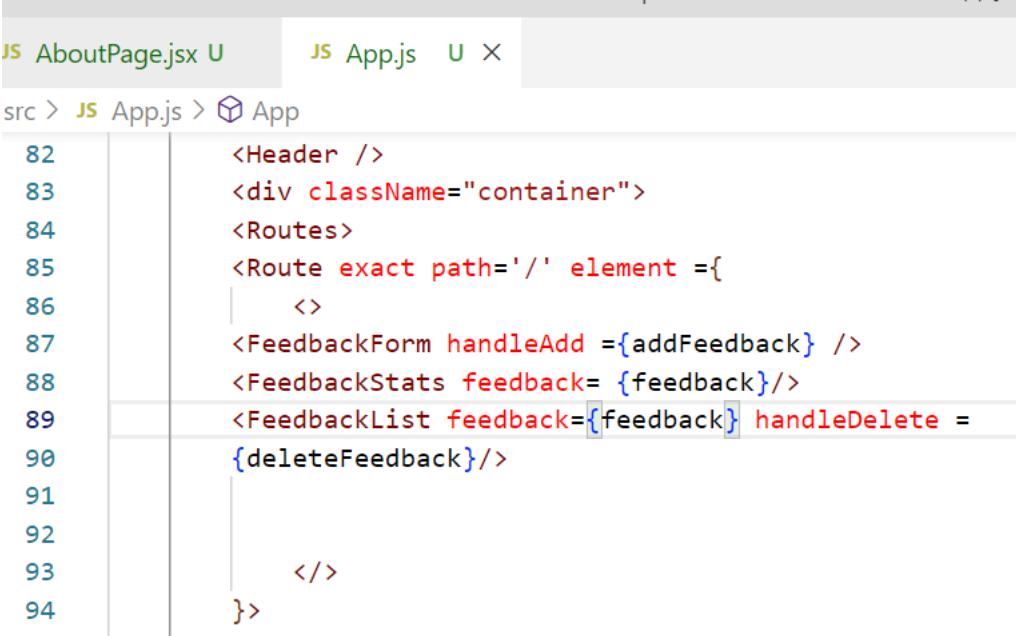
1. Creating Routes (React Router 5)

```
JS AboutPage.jsx U X JS App.js U
src > pages > JS AboutPage.jsx > AboutPage
1 import Card from "../components/shared/Card"
2
3 function AboutPage() {
4   return <Card>
5   <div className="about">
6     <h1>About This Project</h1>
7     <p>This is a React app to leave feedback for a product or service</p>
8     <p>Version: 1.0.0</p>
9
10    <p>
11
12    </p>
13
14  </div>
15
16
17
18  </Card>
19
20
21 export default AboutPage
```

```
JS AboutPage.jsx U JS App.js U X
src > JS App.js > App
55 import {BrowserRouter as Router, Route } from' react-router-dom'
56 import { useState } from 'react'
57 import Header from "./components/Header"
58 import FeedbackData from './data/FeedbackData'
59 import FeedbackList from './components/FeedbackList'
60 import Card from "./components/shared/Card"
61 import FeedbackStats from './components/FeedbackStats'
62 import FeedbackForm from './components/FeedbackForm'
63 import AboutPage from './pages/AboutPage'
```

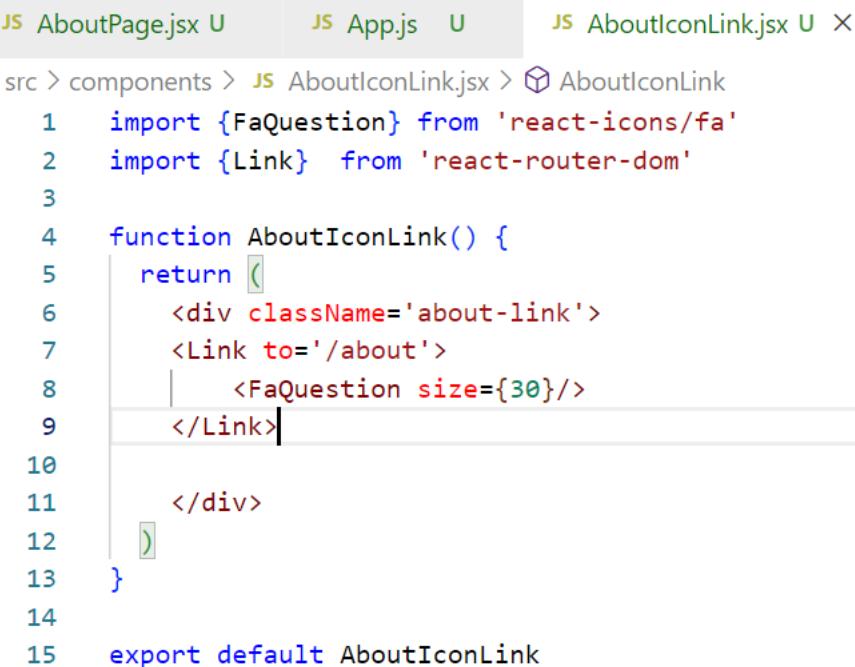
React JS

2. Upgrading To React Router 6



```
JS AboutPage.jsx U JS App.js U X
src > JS App.js > App
82     <Header />
83     <div className="container">
84       <Routes>
85         <Route exact path="/" element ={(
86           <>
87             <FeedbackForm handleAdd ={addFeedback} />
88             <FeedbackStats feedback= {feedback}/>
89             <FeedbackList feedback={feedback} handleDelete =
90               {deleteFeedback}>/>
91           </>
92         )}>
93       </>
94     >
```

3. Creating Links (v5 & v6)



```
JS AboutPage.jsx U JS App.js U JS AboutIconLink.jsx U X
src > components > JS AboutIconLink.jsx > AboutIconLink
1   import {FaQuestion} from 'react-icons/fa'
2   import {Link} from 'react-router-dom'
3
4   function AboutIconLink() {
5     return (
6       <div className='about-link'>
7         <Link to='/about'>
8           <FaQuestion size={30}/>
9         </Link>
10      </div>
11    )
12  }
13
14
15  export default AboutIconLink
```

React JS

```
JS AboutPage.jsx U X JS App.js U JS AboutIconLink.jsx U

src > pages > JS AboutPage.jsx > AboutPage
1 import Card from "../components/shared/Card"
2 import {Link} from 'react-router-dom'
3 function AboutPage() {
4     return <Card>
5     <div className="about">
6         <h1>About This Project</h1>
7         <p>This is a React app to leave feedback for a product or service</p>
8         <p>Version: 1.0.0</p>
9
10        <p>
11            <Link to="/">Back To Home</Link>
12        </p>
13    </div>
14
15
16
17    </Card>
18 }
19
20 export default AboutPage
```

```
JS AboutPage.jsx U JS App.js U X JS AboutIconLink.jsx U

src > JS App.js > App
80
81     return (
82         <Router>
83             <Header />
84             <div className="container">
85                 <Route path="/">
86                     <FeedbackForm handleAdd={addFeedback} />
87                     <FeedbackStats feedback={feedback}/>
88
89                     <FeedbackList feedback={feedback} handleDelete =
90                         {deleteFeedback}>/>
91                 </Route>
92                 <Route path="/about" component={AboutPage}>/>
93
94
95                 <AboutIconLink/>
96             </div>
97         </Router>
98     )
99
100 export default App
```

React JS

4. NavLink & useParams

```
JS AboutPage.jsx U      JS App.js  U      JS Post.jsx  U X
src > components > JS Post.jsx > Post
1  import {useParams} from 'react-router-dom'
2
3  function Post() {
4      const params = useParams()
5      return (
6          <div>
7              <h1>Post {params.id}</h1>
8              <p>Post {params.id}</p>
9          </div>
10     )
11 }
12
13 export default Post
```

React JS

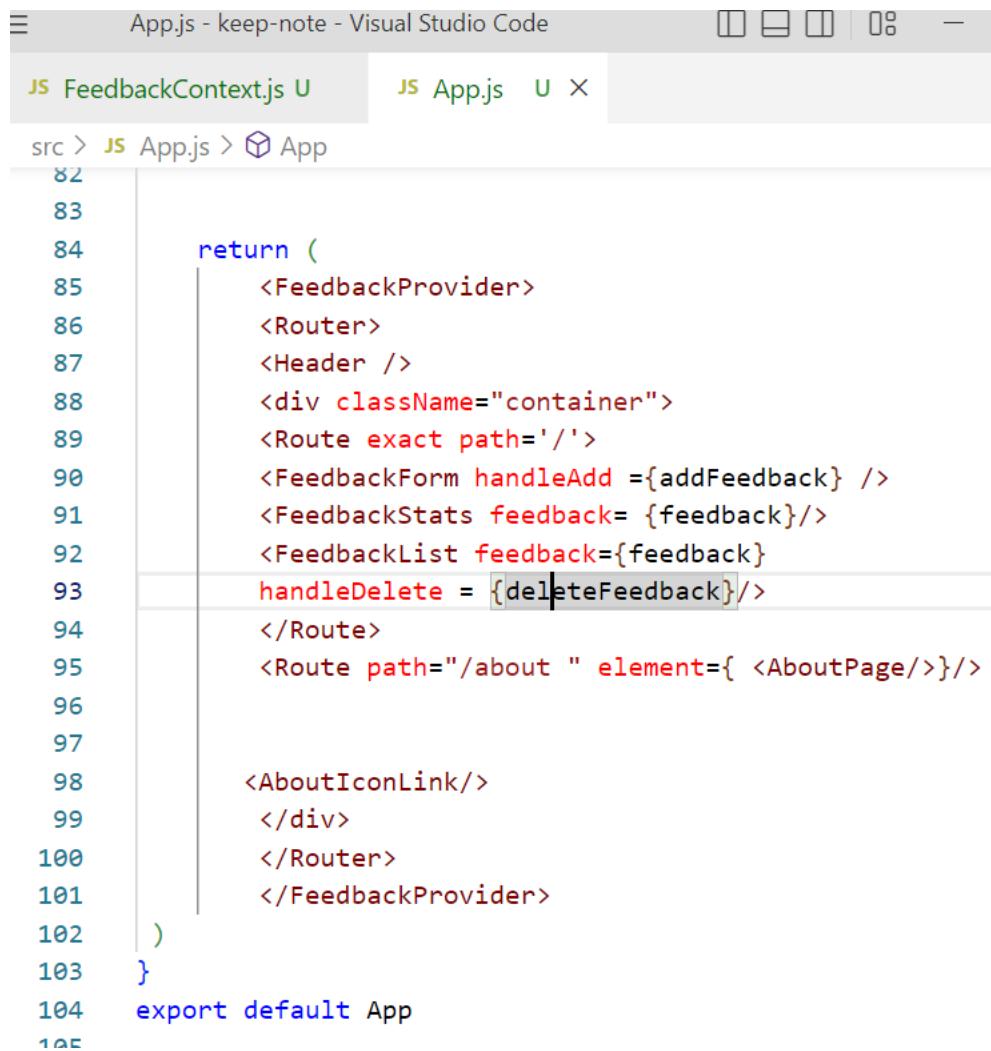
5. Navigate & Nested Routes

```
JS Post.jsx U X
src > components > JS Post.jsx > ...
1   import {Navigate, useNavigate} from 'react-router-dom'
2
3   function Post() {
4       const status = 200
5       const navigate = useNavigate()
6
7       const onClick= () =>{
8           console.log("hello")
9           navigate('/about')
10      }
11
12      if(status ===404){
13          return <Navigate to='notfound'/>
14      }
15      return (
16          <div>
17              <h1>Post</h1>
18              <button onClick={onClick}>Click</button>
19          <Routes>
20              <Route path='/show' element= {<h1> Hello World</h1>}/>
21          </Routes>
22          </div>
23      )
24  }
25
26  export default Post
```

React JS

6. Context API, useContext Hook & Deployment

1. Create a Context & Provider



The screenshot shows the Visual Studio Code interface with the title bar "App.js - keep-note - Visual Studio Code". Below the title bar, there are two tabs: "JS FeedbackContext.js U" and "JS App.js U X". The "App.js" tab is active. The code editor displays the following code:

```
src > JS App.js > ⚙ App
82
83
84     return (
85         <FeedbackProvider>
86             <Router>
87                 <Header />
88                 <div className="container">
89                     <Route exact path="/">
90                         <FeedbackForm handleAdd={addFeedback} />
91                         <FeedbackStats feedback={feedback}/>
92                         <FeedbackList feedback={feedback}>
93                             handleDelete={deleteFeedback}/>
94                         </FeedbackList>
95                     <Route path="/about" element={<AboutPage/>} />
96
97                     <AboutIconLink/>
98                 </div>
99             </Router>
100        </FeedbackProvider>
101    )
102 }
103
104 export default App
105
```

React JS

The screenshot shows a Visual Studio Code window with the title bar "FeedbackContext.js - keep-note - Visual Studio Code". Below the title bar, there are tabs for "FeedbackContext.js" and "App.js". The main editor area displays the following code:

```
src > context > JS FeedbackContext.js < default
1 import { createContext, useState } from "react"
2
3 const FeedbackContext = createContext()
4
5 export const FeedbackProvider = ({children}) => {
6   const [feedback, setFeedback] = useState([
7     {
8       id:1,
9       text:'This item is from context',
10      rating: 10
11    }
12  ])
13
14  return <FeedbackContext.Provider value={{
15    feedback,
16
17  }}>
18    {children}
19
20  </FeedbackContext.Provider>
21
22
23 export default FeedbackContext
```

React JS

2. Get Global State With The useContext Hook



The screenshot shows the code for `App.js` in a code editor. The file imports several components and a provider from other files. It uses the `useState` hook to manage the `feedback` state and provides it to the `FeedbackList` component via props.

```
src > JS App.js > App
  53 import FeedbackItem from "./components/FeedbackItem"
  54 import {v4 as uuidv4 } from 'uuid'
  55 import {BrowserRouter as Router, Route } from' react-router-dom'
● 56 import { useState } from 'react'
  57 import Header from "./components/Header"
  58 import FeedbackData from './data/FeedbackData'
  59 import FeedbackList from './components/FeedbackList'
  60 import Card from "./components/shared/Card"
  61 import FeedbackStats from './components/FeedbackStats'
  62 import FeedbackForm from './components/FeedbackForm'
  63 import AboutPage from './pages/AboutPage'
  64
  65 import { FeedbackProvider } from "./context/FeedbackContext"
  66
  67 import AboutIconLink from "./components/AboutIconLink"
  68 function App(){
  69
  70     const [feedback, setFeedback] = useState(FeedbackData)
  71
  72     const addFeedback =(newFeedback) =>{
  73         newFeedback.id = uuidv4()
  74         setFeedback([newFeedback, ...feedback])
  75     }
  76
  77     const deleteFeedback = (id) => {
  78         if(window.confirm('Are you sure you want to delete?')){
  79             setFeedback(feedback.filter((item) => item.id !== id))
  80         }
  81     }
  ~~
```

React JS

The screenshot shows a code editor interface with a tab bar at the top. The active tab is 'src > App.js'. The code itself is a React component named 'App'.

```
JS FeedbackContext.js U   JS FeedbackList.jsx U   JS FeedbackStats.jsx U   JS API.js U
src > JS App.js > App
83     return (
84       <FeedbackProvider>
85         <Router>
86           <Header />
87           <div className="container">
88             <Route exact path="/">
89               <FeedbackForm handleAdd ={addFeedback} />
90               <FeedbackStats />
91             <FeedbackList
92               handleDelete = {deleteFeedback}/>
93             </Route>
94             <Route path="/about" element={ <AboutPage/>} />
95
96
97             <AboutIconLink/>
98           </div>
99         </Router>
100        </FeedbackProvider>
101      )
102    )
103  }
104 export default App
---
```

React JS

```
JS FeedbackContext.js U JS FeedbackList.jsx U JS FeedbackStats.jsx U X JS App.js
src > components > JS FeedbackStats.jsx > ...
2 // import PropTypes from 'prop-types'
3 import { useContext } from 'react'
4 import FeedbackContext from '../context/FeedbackContext'
5
6 function FeedbackStats({}) {
7   const {feedback} = useContext(FeedbackContext)
8   // Calculate rating Average
9
10  let average =
11    feedback.reduce((acc, cur) =>{
12      return acc +cur.rating
13    },0) / feedback.length
14
15  average = average.toFixed(1).replace(/[.,]0$/,'')
16
17  return (
18    <div className="feedback-stats">
19      <h4>{feedback.length} Reviews</h4>
20      <h4> Average Rating: {isNaN(average) ? 0 : average}</h4>
21    </div>
22  )
23}
24
25 // FeedbackStats.propTypes = {
26 //   feedback: PropTypes.array.isRequired,
27 // }
28
29 export default FeedbackStats
```

React JS

JS FeedbackContext.js U JS FeedbackList.jsx U X JS FeedbackStats.jsx U

src > components > JS FeedbackList.jsx > FeedbackList > feedback.map() callback

```
1 import {motion, AnimatePresence} from 'framer-motion'
2 import { useContext } from 'react'
3 // import PropTypes from 'prop-types'
4 import FeedbackItem from "./FeedbackItem"
5 import FeedbackContext from '../context/FeedbackContext'
6
7 function FeedbackList({ handleDelete }) {
8     const {feedback} = useContext(FeedbackContext)
9     if(!feedback || feedback.length ===0){
10         return <p> No Feedback Yet</p>
11     }
12     return (
13         <div className="feedback-list">
14             <AnimatePresence>
15                 {feedback.map((item) => (
16                     <motion.div
17                         key={item.id}
18                         initial={{ opacity: 0}}
19                         animate={{opacity: 1}}
20                         exit={{opacity: 0}}>
21
22                         <FeedbackItem
23                             key= {item.id}
24                             item = {item}
25                             handleDelete={handleDelete}
26                         />
27                     </motion.div>
28                 )))
29     )
30 }
```

React JS

```
  S FeedbackContext.js U      JS FeedbackList.jsx U X      JS FeedbackStats
src > components > JS FeedbackList.jsx > FeedbackList > feedback.ma
29     </AnimatePresence>
30
31   </div>
32 }
33 }
34
35 // FeedbackList.propTypes ={
36 //   feedback: PropTypes.arrayOf(
37 //     PropTypes.shape({
38 //       id: PropTypes.number.isRequired,
39 //       text:PropTypes.string.isRequired,
40 //       rating: PropTypes.number.isRequired,
41 //     })
42 //   ),
43 // }
44
45
46 export default FeedbackList

JS FeedbackContext.js U X      JS FeedbackList.jsx U      JS FeedbackStats
src > context > JS FeedbackContext.js > default
1 import { createContext, useState } from "react"
2
3 const FeedbackContext = createContext()
4
5 export const FeedbackProvider = ({children}) => {
6   const [feedback, setFeedback] = useState([
7     {
8       id:1,
9       text:'This item is from context',
10      rating: 10
11    }
12  ])
13
14  return <FeedbackContext.Provider value={{
15    feedback,
16
17  }}>
18    {children}
19
20  </FeedbackContext.Provider>
21}
22
23 export default FeedbackContext
```

React JS

3. Moving Functions To Context

```
JS FeedbackContext.js U X JS App.js U JS FeedbackForm.js  
src > context > JS FeedbackContext.js > [FeedbackProvider]  
1 import {v4 as uuidv4 } from 'uuid'  
2 import { createContext, useState } from "react"  
3  
4 const FeedbackContext = createContext()  
5  
6 export const FeedbackProvider = ({children}) =>  
7   const [feedback, setFeedback] = useState([  
8     {  
9       id:1,  
10      text:'This is feedback item 1',  
11      rating: 10  
12    },  
13    {  
14      id:2,  
15      text:'This is feedback item 2',  
16      rating: 8  
17    },  
18    {  
19      id:3,  
20      text:'This is feedback item 3',  
21      rating: 9  
22    },  
23  ])  
24  
25  const addFeedback =(newFeedback) =>{  
26    newFeedback.id = uuidv4()  
27    setFeedback([newFeedback, ...feedback])  
28  }  
29 }
```

React JS

```
JS FeedbackContext.js U X JS App.js U JS FeedbackForm.jsx U JS Feedbackite
src > context > JS FeedbackContext.js > [?] FeedbackProvider
30
31     const deleteFeedback = (id) => {
32         if(window.confirm('Are you sure you want to delete?')){
33             setFeedback(feedback.filter(item => item.id !== id))
34         }
35     }
36
37     return <FeedbackContext.Provider value={{
38         feedback,
39         deleteFeedback,
40         addFeedback,
41     }}>
42         {children}
43     </FeedbackContext.Provider>
44
45
46 }
47
48 export default FeedbackContext
```

```
JS App.js U X JS FeedbackContext.js U JS FeedbackForm.jsx U JS Feedbackite
src > JS App.js > [?] default
105 import FeedbackItem from './components/FeedbackItem'
106 import {BrowserRouter as Router, Route } from 'react-router-dom'
107 import Header from './components/Header'
108 import FeedbackList from './components/FeedbackList'
109 import Card from './components/shared/Card'
110 import FeedbackStats from './components/FeedbackStats'
111 import FeedbackForm from './components/FeedbackForm'
112 import AboutPage from './pages/AboutPage'
113 import { FeedbackProvider } from './context/FeedbackContext'
114 import AboutIconLink from './components/AboutIconLink'
115 function App(){
116 }
```

React JS

JS App.js U X JS FeedbackContext.js U JS FeedbackForm.jsx U .

```
src > JS App.js > [e] default
116
117     return (
118         <FeedbackProvider>
119             <Router>
120                 <Header />
121                 <div className="container">
122                     <Route exact path="/">
123                         <FeedbackForm />
124                         <FeedbackStats />
125                         <FeedbackList />
126                     </Route>
127                     <Route path="/about" element={ <AboutPage/> } />
128
129
130                     <AboutIconLink />
131                 </div>
132             </Router>
133         </FeedbackProvider>
134     )
135 }
136 export default App
```

JS FeedbackForm.jsx U X JS App.js U JS FeedbackContext.js U JS Feed

```
src > components > JS FeedbackForm.jsx > [e] FeedbackForm > [e] handleSubmit
1 import { useState, useContext } from "react"
2 import Card from "./shared/Card"
3 import Button from "./shared/Button"
4 import RatingSelect from "./RatingSelect"
5 import FeedbackContext from "../context/FeedbackContext"
6 function FeedbackForm() {
7
8     const [text, setText] = useState('')
9     const [rating, setRating] = useState(10)
10    const [btnDisabled, setBtnDisabled] = useState(true)
11    const [message, setMessage] = useState('')
12
13    const {addFeedback} = useContext(FeedbackContext)
14    const handleTextChange= (e) => {
15        if(text === ''){
16            setBtnDisabled(true)
17            setMessage(null)
18        } else if(text !== '' && text.trim().length <= 10){
19            setMessage('Text must be at least 10 characters')
20            setBtnDisabled(true)
21        } else {
22            setMessage(null)
23            setBtnDisabled(false)
24        }
25    }
26
27    const handleRatingSelect= (rating) => {
28        setRating(rating)
29    }
30
31    const handleFeedbackSubmit= (e) => {
32        e.preventDefault()
33        addFeedback({text, rating})
34    }
35
36    return (
37        <Card>
38            <form onSubmit={handleFeedbackSubmit}>
39                <div>
40                    <RatingSelect rating={rating} handleRatingSelect={handleRatingSelect} />
41                    <input type="text" value={text} onChange={handleTextChange} />
42                </div>
43                <Button type="submit" disabled={btnDisabled} />
44            </form>
45        </Card>
46    )
47}
```

React JS

JS FeedbackForm.jsx U X JS App.js U JS FeedbackContext.js U JS FeedbackItem.jsx U

src > components > JS FeedbackForm.jsx > FeedbackForm > [e] handleSubmit

```
28  const handleSubmit = (e) => {
29    e.preventDefault()
30    if(text.trim().length >10 ) {
31      const newFeedback = {
32        text,
33        rating,
34      }
35
36      addFeedback(newFeedback)
37
38      setText('')
39    }
40  }
41  return (
42    <Card>
43    <form onSubmit={handleSubmit}>
44      <h2>How Would you rate your service with us?</h2>
45      {/* @todo - rating select component */}
46      <RatingSelect select= {(rating) => setRating(rating)} />
47
48
49      <div className="input-group">
50        <input
51          onChange={handleTextChange}
52          type= 'text'
53          placeholder="Write a review"
54          value = {text}
55        />

```

JS FeedbackForm.jsx U X JS App.js U JS FeedbackContext.js U JS FeedbackItem.jsx U

src > components > JS FeedbackForm.jsx > FeedbackForm > [e] handleSubmit

```
55    />
56    <Button type="submit" isDisabled={btnDisabled}>Send</Button>
57  </div>
58  {message && <div className="message">{message}</div>}
59  </form>
60  </Card>
61 )
62 }
63 export default FeedbackForm
```

React JS

JS FeedbackItem.jsx U X JS FeedbackForm.jsx U JS App.js U JS FeedbackContext.js

src > components > JS FeedbackItem.jsx > 📄 FeedbackItem > 📥 constructor

```
12 import {FaTimes} from 'react-icons/fa'
13 import { useContext } from 'react'
14 import PropTypes from 'prop-types'
15 import Card from './shared/Card'
16 import FeedbackContext from '../context/FeedbackContext'
17 function FeedbackItem({item}) {
18
19     const {deleteFeedback} = useContext(FeedbackContext)
20
21
22     return (
23         <Card>
24             <div className="num-display">{item.rating}</div>
25             <button onClick={() => deleteFeedback(item.id)} className='close'>
26                 <FaTimes color='purple'/>
27             </button>
28             <div className="text-display">{item.text} </div>
29             {/* <button onClick={handleClick}>Click</button> */}
30         </Card>
31     )
32 }
33
34 FeedbackItem.propTypes ={
35     item: PropTypes.object.isRequired
36 }
37 export default FeedbackItem
```

React JS

4. Edit Feedback Event

```
JS FeedbackItem.jsx U X JS FeedbackContext.js U
src > components > JS FeedbackItem.jsx > FeedbackItem > constructor
12   import {FaTimes, FaEdit } from 'react-icons/fa'
13   import { useContext } from 'react'
14   import PropTypes from 'prop-types'
15   import Card from './shared/Card'
16   import FeedbackContext from '../context/FeedbackContext'
17   function FeedbackItem({item }) {
18
19     const {deleteFeedback, editFeedback} = useContext( FeedbackContext)
20
21
22     return (
23       <Card>
24         <div className="num-display">{item.rating}</div>
25         <button onClick={() => deleteFeedback(item.id)} className='close'>
26           <FaTimes color='purple'/>
27         </button>
28         <button onClick={() => editFeedback(item)} className='edit'>
29           <FaEdit color='purple'/>
30
31         </button>
32         <div className="text-display">{item.text} </div>
33         {/* <button onClick={handleClick}>Click</button> */}
34       </Card>
35     )
36   }
```

```
JS FeedbackItem.jsx U X JS FeedbackContext.js U
src > components > JS FeedbackItem.jsx > FeedbackItem > constructor
37
38   FeedbackItem.propTypes ={
39     item: PropTypes.object.isRequired
40   }
41   export default FeedbackItem
```

React JS

```
JS FeedbackContext.js U X JS FeedbackItem.jsx U
src > context > JS FeedbackContext.js > [?] FeedbackProvider
1  import {v4 as uuidv4 } from 'uuid'
2  import { createContext, useState } from "react"
3
4  const FeedbackContext = createContext()
5
6  export const FeedbackProvider = ({children}) => {
7      const [feedback, setFeedback] = useState([
8          {
9              id:1,
10             text:'This is feedback item 1',
11             rating: 10
12         },
13         {
14             id:2,
15             text:'This is feedback item 2',
16             rating: 8
17         },
18         {
19             id:3,
20             text:'This is feedback item 3',
21             rating: 9
22         },
23     ])
24
25     const [feedbackEdit, setFeedbackEdit] = useState({
26         item: {},
27         edit: false
28     })
~~
```

React JS

JS FeedbackContext.js U X JS FeedbackItem.jsx U

src > context > JS FeedbackContext.js > [🔗] FeedbackProvider > [🔗] addFeedback

```
31     //Add Feedback
32     const addFeedback = (newFeedback) =>{
33         newFeedback.id = uuidv4()
34         setFeedback([newFeedback, ...feedback])
35
36     }
37     //Delete Feedback
38     const deleteFeedback = (id) => {
39         if(window.confirm('Are you sure you want to delete?')){
40             setFeedback(feedback.filter((item) => item.id !== id))
41         }
42     }
43     // Set item to be updated
44     const editFeedback = (item) =>{
45         setFeedbackEdit({
46             item,
47             edit:true
48         })
49     }
50     return <FeedbackContext.Provider value={{
51         feedback,
52         deleteFeedback,
53         addFeedback,
54         editFeedback,
55     }}>
56         {children}
57     </FeedbackContext.Provider>
58 }
59 export default FeedbackContext
```

React JS

5. Side Effects With useEffect:

```
JS FeedbackContext.js U JS FeedbackForm.jsx U JS RatingSelect.jsx U X
src > components > JS RatingSelect.jsx > RatingSelect
1  import {useState, useContext, useEffect} from 'react'
2  import FeedbackContext from "../context/FeedbackContext"
3
4
5  function RatingSelect({select}) {
6
7      const [selected, setSelected] = useState(5)
8
9      const { feedbackEdit } = useContext(FeedbackContext)
10
11     useEffect(() => {
12         setSelected(feedbackEdit.item.rating)
13     }, [feedbackEdit])
14
15     const handleChange = (e) => {
16         setSelected(+e.currentTarget.value)
17         select(+e.currentTarget.value)
18     }
19
20     return (
21         <ul className='rating'>
22             <li>
23                 <input
24                     type='radio'
25                     id='num1'
26                     name='rating'
27                     value='1'
28                     onChange={handleChange}
29                     checked={selected === 1}>
30             />
31         </li>
32     )
33 }
34
35 export default RatingSelect
```

React JS

JS FeedbackItem.jsx U X JS FeedbackContext.js U JS FeedbackForm.jsx U JS RatingSelect.

src > components > JS FeedbackItem.jsx > FeedbackItem > constructor

```
12 import {FaTimes, FaEdit } from 'react-icons/fa'
13 import { useContext } from 'react'
14 import PropTypes from 'prop-types'
15 import Card from './shared/Card'
16 import FeedbackContext from '../context/FeedbackContext'
17 function FeedbackItem({item }) {
18
19     const {deleteFeedback, editFeedback} = useContext( FeedbackContext)
20     return (
21         <Card>
22             <div className="num-display">{item.rating}</div>
23             <button onClick={() => deleteFeedback(item.id)} className='close'>
24                 <FaTimes color='purple'/>
25             </button>
26             <button onClick={() => editFeedback(item)} className='edit'>
27                 <FaEdit color='purple'/>
28
29         </button>
30         <div className="text-display">{item.text} </div>
31         {/* <button onClick={handleClick}>Click</button> */}
32     </Card>
33 )
34 }
35 FeedbackItem.propTypes ={
36     item: PropTypes.object.isRequired
37 }
38 export default FeedbackItem
```

React JS

JS FeedbackContext.js U X JS FeedbackItem.jsx U JS FeedbackForm.jsx U JS Rat

src > context > JS FeedbackContext.js > [edit] FeedbackProvider

```
37     //Delete Feedback
38     const deleteFeedback = (id) => {
39         if(window.confirm('Are you sure you want to delete?')){
40             setFeedback(feedback.filter((item) => item.id !== id))
41         }
42     }
43 // Set item to be updated
44     const editFeedback = (item) =>{
45         setFeedbackEdit({
46             item,
47             edit:true
48         })
49     }
50     return <FeedbackContext.Provider value={{
51         feedback,
52         deleteFeedback,
53         addFeedback,
54         editFeedback,
55         feedbackEdit,
56     }}>
57         {children}
58     </FeedbackContext.Provider>
59 }
60 export default FeedbackContext
```

React JS

```
src > components > JS FeedbackForm.jsx U > JS FeedbackForm > useEffect() callback
1 import { useState, useContext, useEffect } from "react"
2 import Card from "./shared/Card"
3 import Button from "./shared/Button"
4 import RatingSelect from "./RatingSelect"
5 import FeedbackContext from "../context/FeedbackContext"
6 function FeedbackForm() {
7
8     const [text, setText] = useState('')
9     const [rating, setRating] = useState(10)
10    const [btnDisabled, setBtnDisabled] = useState(true)
11    const [message, setMessage] = useState('')
12
13    const {addFeedback, feedbackEdit} = useContext(FeedbackContext)
14
15    useEffect(const setBtnDisabled: (value: React.SetStateAction<boolean>) => void
16    {
17        if(f
18            setBtnDisabled(false)
19            setText(feedbackEdit.item.text)
20            setRating(feedbackEdit.item.rating)
21
22    }, [feedbackEdit])
23    const handleChange= (e) => {
24        if(text == ''){
25            setBtnDisabled(true)
26            setMessage(null)
27        } else if(text != '' && text.trim().length <= 10){
28            setMessage('Text must be at least 10 characters')
29            setBtnDisabled(true)
```

React JS

The screenshot shows a code editor interface with several tabs at the top: 'FeedbackContext.js U', 'FeedbackItem.jsx U', 'FeedbackForm.jsx U X', and 'RatingSele'. Below the tabs, a breadcrumb navigation bar indicates the file path: 'src > components > FeedbackForm.jsx'. A callout bubble highlights the 'useEffect()' callback function. The main content area displays the code for 'FeedbackForm.jsx'.

```
44     addFeedback(newFeedback)
45
46     setText('')
47   }
48 }
49
50 return (
51   <Card>
52     <form onSubmit={handleSubmit}>
53       <h2>How Would you rate your service with us?</h2>
54       {/* @todo - rating select component */}
55       <RatingSelect select= {(rating) => setRating(rating)} />
56
57
58       <div className="input-group">
59         <input
60           onChange={handleTextChange}
61           type= 'text'
62           placeholder="Write a review"
63           value = {text}
64         />
65         <Button type="submit" isDisabled={btnDisabled}>Send</Button>
66       </div>
67       {message && <div className="message">{message}</div>}
68     </form>
69   </Card>
70 )
71 }
72 export default FeedbackForm
```

React JS

6. Update Feedback Item



```
JS FeedbackContext.js U X JS FeedbackForm.jsx U
src > context > JS FeedbackContext.js > [e] FeedbackProvider
30
31     //Add feedback
32     const addFeedback = (newFeedback) =>{
33         newFeedback.id = uuidv4()
34         setFeedback([newFeedback, ...feedback])
35     }
36
37     //Delete Feedback
38     const deleteFeedback = (id) => {
39         if(window.confirm('Are you sure you want to delete?')){
40             setFeedback(feedback.filter((item) => item.id !== id))
41         }
42     }
43
44     // Update feedback item
45
46     const updateFeedback = (id, updItem) =>{
47         setFeedback(feedback.map((item) => item.id === id? { ...
48             item, ...updItem} : item))
49     }
50
51
52     // Set item to be updated
53     const editFeedback = (item) =>{
54         setFeedbackEdit({
55             item,
56             edit:true
57         })
58     }
```

React JS

JS FeedbackContext.js U X JS FeedbackForm.jsx U

src > context > JS FeedbackContext.js > [o] FeedbackProvider

```
51
52 // Set item to be updated
53     const editFeedback = (item) =>{
54         setFeedbackEdit({
55             item,
56             edit:true
57         })
58     }
59     return <FeedbackContext.Provider value={{
60         feedback,
61         deleteFeedback,
62         addFeedback,
63         editFeedback,
64         feedbackEdit,
65         updateFeedback,
66     }}>
67         {children}
68     </FeedbackContext.Provider>
69 }
70 export default FeedbackContext
```

React JS

JS FeedbackContext.js U

JS FeedbackForm.jsx U X

```
src > components > JS FeedbackForm.jsx > ⚡ FeedbackForm > [⌚] handleSubmit
  1 import { useState, useContext, useEffect } from "react"
  2 import Card from "./shared/Card"
  3 import Button from "./shared/Button"
  4 import RatingSelect from "./RatingSelect"
  5 import FeedbackContext from "../context/FeedbackContext"
  6 function FeedbackForm() {
  7
  8     const [text, setText] = useState('')
  9     const [rating, setRating] = useState(10)
10     const [btnDisabled, setBtnDisabled] = useState(true)
11     const [message, setMessage] = useState('')
12
13     const {addFeedback, feedbackEdit, updateFeedback} = useContext(FeedbackContext)
14
15     useEffect(() =>{
16         if(feedbackEdit.edit === true){
17             setBtnDisabled(false)
18             setText(feedbackEdit.item.text)
19             setRating(feedbackEdit.item.rating)
20         }
21
22     }, [feedbackEdit])
23     const handleTextChange= (e) => {
24         if(text === ''){
25             setBtnDisabled(true)
26             setMessage(null)
27         } else if(text !== '' && text.trim().length <= 10){
28             setMessage('Text must be at least 10 characters')
29             setBtnDisabled(true)

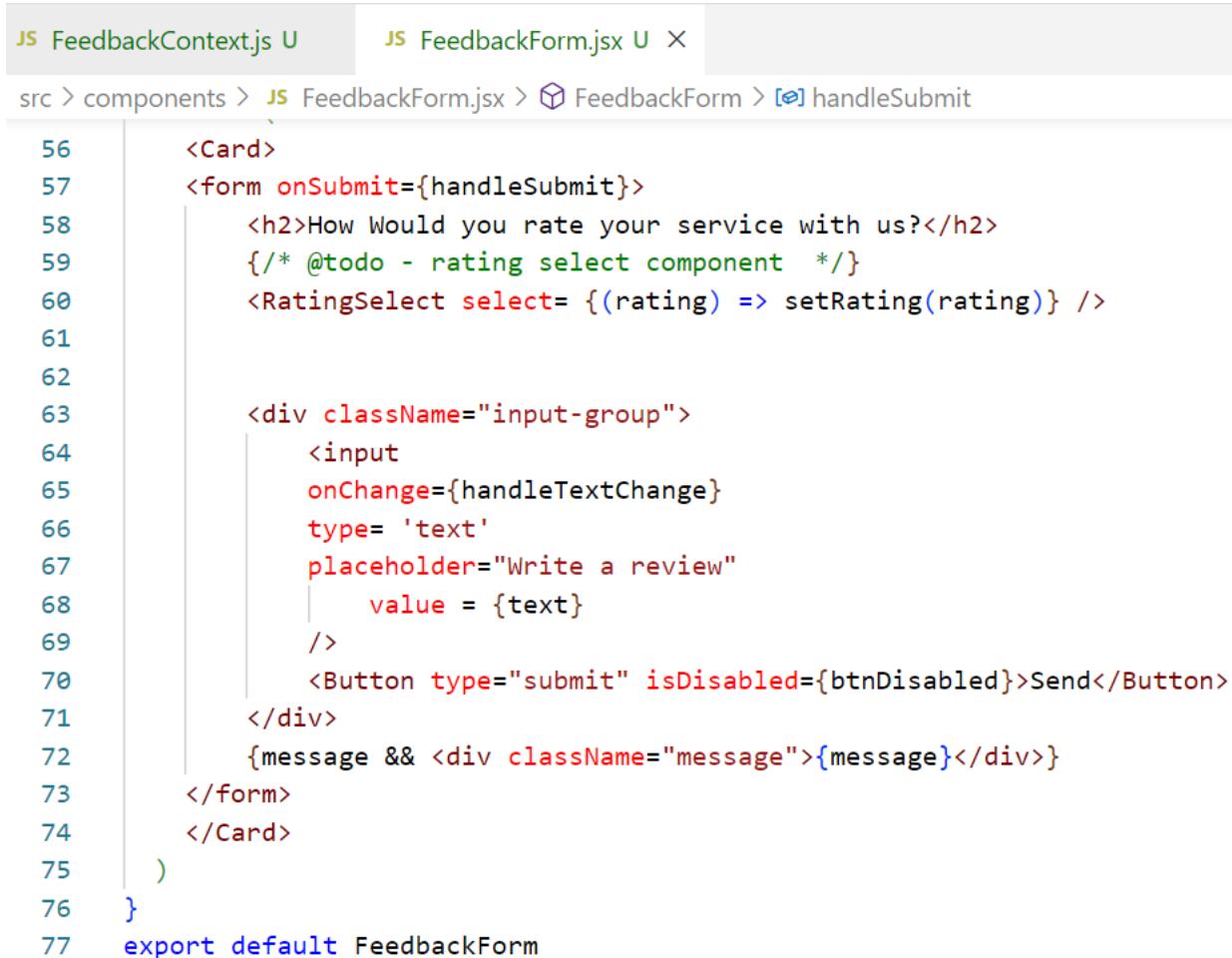
```

React JS

The screenshot shows a code editor interface with two tabs at the top: "JS FeedbackContext.js U" and "JS FeedbackForm.jsx U". The "FeedbackForm.jsx" tab is currently selected, indicated by a grey background. Below the tabs, the file path is shown as "src > components > JS FeedbackForm.jsx > FeedbackForm > handleSubmit". The code itself is a JavaScript function named "handleSubmit" with several nested blocks and conditional statements. The code uses standard ES6 syntax like arrow functions and const/let declarations.

```
JS FeedbackContext.js U      JS FeedbackForm.jsx U X
src > components > JS FeedbackForm.jsx > FeedbackForm > handleSubmit
30     } else {
31         setMessage(null)
32         setBtnDisabled(false)
33     }
34     setText(e.target.value)
35 }
36
37     const handleSubmit = (e) => {
38         e.preventDefault()
39         if(text.trim().length >10 ) {
40             const newFeedback = {
41                 text,
42                 rating,
43             }
44
45             if(feedbackEdit.edit === true){
46                 updateFeedback(feedbackEdit.item.id, newFeedback)
47             } else {
48                 addFeedback(newFeedback)
49             }
50
51             setText('')
52         }
53     }
54 }
55 return (
```

React JS



The screenshot shows a code editor interface with two tabs at the top: 'FeedbackContext.js' and 'FeedbackForm.jsx'. The 'FeedbackForm.jsx' tab is active, showing the following code:

```
src > components > FeedbackForm.jsx > FeedbackForm > handleSubmit
```

```
56   <Card>
57     <form onSubmit={handleSubmit}>
58       <h2>How Would you rate your service with us?</h2>
59       {/* @todo - rating select component */}
60       <RatingSelect select= {rating) => setRating(rating)} />
61
62
63       <div className="input-group">
64         <input
65           onChange={handleTextChange}
66           type= 'text'
67           placeholder="Write a review"
68           value = {text}
69         />
70         <Button type="submit" isDisabled={btnDisabled}>Send</Button>
71       </div>
72       {message && <div className="message">{message}</div>}
73     </form>
74     </Card>
75   )
76 }
77 export default FeedbackForm
```

7. Deploy To Netlify

React JS

7. APIs & HTTP Requests

1. APIs & Requests Explained

FULLSTACK APPLICATION OVERVIEW

- Data is kept in a database on a server
- There is some kind of backend API (Written in any language)
- The API has routes that we can access from the client (React) in order to create, read, update and delete data
- Data is brought into our state and used in our React app



HTTP METHODS

GET

Retrieves data from the server

POST

Submit data to the server

PUT / PATCH

Update data already on the server

DELETE

Delete data from the server

GET	/feedback
GET	/feedback/1
POST	/feedback
PUT	/feedback/1
DELETE	/feedback/1

React JS

HTTP STATUS CODES

1xx: Informational

Request received / processing

2xx: Success

Successfully received

3xx: Redirect

Further action must be taken / redirect

4xx: Client Error

Request does not have what it needs

5xx: Server Error

Server failed to fulfill a valid request

2. Setting Up JSON-Server Mock Backend

```
up to date, audited 1573 packages in 2s

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

  11 vulnerabilities (5 moderate, 6 high)

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

Run `npm audit` for details.
PS C:\Users\DELL\Desktop\Sudip\keep-note> █
```

The screenshot shows a code editor interface with two tabs: 'package.json' and 'FeedbackForm.jsx'. The 'package.json' tab is active, displaying the following script section:

```
{} package.json M X JS FeedbackContext.js U JS FeedbackForm.jsx U
{} package.json > {} scripts
  ↴ Debug
18   "scripts": {
19     "start": "react-scripts start",
20     "build": "react-scripts build",
21     "test": "react-scripts test",
22     "eject": "react-scripts eject",
23     "server": "json-server --watch db.json --port 5000"
24   },
```

React JS

The screenshot shows the Postman application interface. At the top, there's a header with a port dropdown set to '5000' and a URL field containing 'http://localhost:5000/feedback'. Below the header, the URL 'http://localhost:5000/feedback' is displayed again. A navigation bar includes 'GET', 'Params', 'Authorization', 'Headers (6)', 'Body', 'Pre-request Script', 'Tests', and 'Settings'. The 'Params' tab is selected, showing a table for 'Query Params' with columns 'KEY' and 'VALUE'. Under the 'Body' tab, which is selected, the response is shown in 'Pretty' format as a JSON array:

```
1 [ { "id": 1, "rating": 10, "text": "this is feedback item 1 coming from the backend" }, { "id": 2, "rating": 7, "text": "this is feedback item 2 coming from the backend" }, { "id": 3, "rating": 5, "text": "this is feedback item 3 coming from the backend" } ]
```

React JS

```
{} db.json  u X
{} db.json > [ ] posts
1  {
2
3      "feedback": [
4          {
5              "id": 1,
6              "rating": 10,
7              "text": "this is feedback item 1 coming from the backend"
8          },
9          {
10             "id": 2,
11             "rating": 7,
12             "text": "this is feedback item 2 coming from the backend"
13         },
14         {
15             "id": 3,
16             "rating": 4,
17             "text": "this is feedback item 3 coming from the backend"
18         }
19     ],
20 }
```

React JS

```
{ } db.json  U X
{ } db.json > [ ]feedback
3   {
4     "id": 1,
5     "rating": 10,
6     "text": "this is feedback item 1 coming from the backend"
7   },
8   {
9     "id": 2,
10    "rating": 7,
11    "text": "this is feedback item 2 coming from the backend"
12  },
13  {
14    "id": 3,
15    "rating": 4,
16    "text": "this is feedback item 3 coming from the backend"
17  },
18  {
19    "rating": "5",
20    "text": "This is new react json project at the date 2079-0624",
21    "id": 4
22  },
23  {
24    "rating": "8",
25    "text": "This is new one react json project at the date 2079-0624",
26    "id": 5
27 }
```

React JS

The screenshot shows the Postman application interface. At the top, there are two tabs: 'GET http://localhost:5000/f' and 'POST http://localhost:5000/'. The 'POST' tab is selected. Below the tabs, the URL 'http://localhost:5000/feedback' is displayed. The main area is titled 'POST http://localhost:5000/feedback'. Under the 'Body' tab, the content type is set to 'x-www-form-urlencoded'. The data table contains two entries:

KEY	VALUE
rating	8
text	This is new one react json project at the d...

At the bottom of the body section, there are tabs for 'Pretty', 'Raw', 'Preview', 'Visualize', and 'JSON'. The 'Pretty' tab is selected. The JSON output is:

```
1 {  
2   "rating": "8",  
3   "text": "This is new one react json project at the date 2079-0624",  
4   "id": 5  
5 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER

Read error has been fixed :)
db.json has changed, reloading...

DELETE /feedback/5 200 26.429 ms - 2
[]

React JS

The screenshot shows the Postman application interface. At the top, there are three tabs: Import (selected), GET http://localhost:5000/f (disabled), POST http://localhost:5000/ (disabled), and DEL http://localhost:5000/f (disabled). To the right of these tabs are buttons for creating a new item (+), cloning (ooo), and selecting an environment (No Environment). Below the tabs, the URL `http://localhost:5000/feedback/5` is entered, and there are Save, Edit, and Delete buttons.

The main area shows a DELETE request to `http://localhost:5000/feedback/5`. The request details section includes tabs for Params (selected), Authorization, Headers (6), Body, Pre-request Script, Tests, Settings, and Cookies. The Headers tab shows 6 items. The Body tab is selected, showing the following table:

KEY	VALUE	DESCRIPTION	ooo	Bulk Edit
Key	Value	Description		

Below the table, the response section shows the following details: Body, Cookies, Headers (13), Test Results. The Headers tab shows 13 items. The response status is 200 OK, 33 ms, 395 B. The response body is shown in Pretty format as `1 [{}]`.

React JS

3. Run Client & Server With Concurrently

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

  11 vulnerabilities (5 moderate, 6 high)

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

  Run `npm audit` for details.
PS C:\Users\DELL\Desktop\Sudip\keep-note>
```



The screenshot shows a code editor with the file 'package.json' open. The code is as follows:

```
{}
{} package.json M X

{} package.json > {} eslintConfig > [ ] extends > abc 1

19   "scripts": {
20     "start": "react-scripts start",
21     "build": "react-scripts build",
22     "test": "react-scripts test",
23     "eject": "react-scripts eject",
24     "server": "json-server --watch db.json --port 5000",
25     "dev": "concurrently \"npm run server\" \"npm start\""
26   },
```

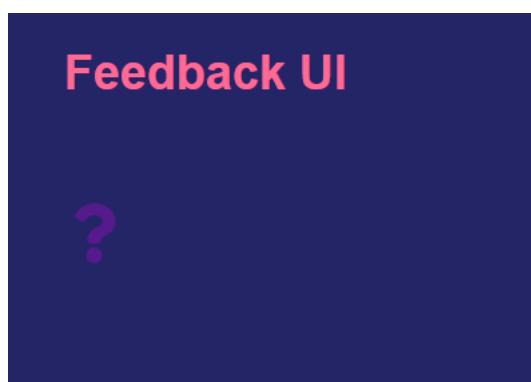
React JS

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER

PS C:\Users\DELL\Desktop\Sudip\keep-note> `npm run dev`

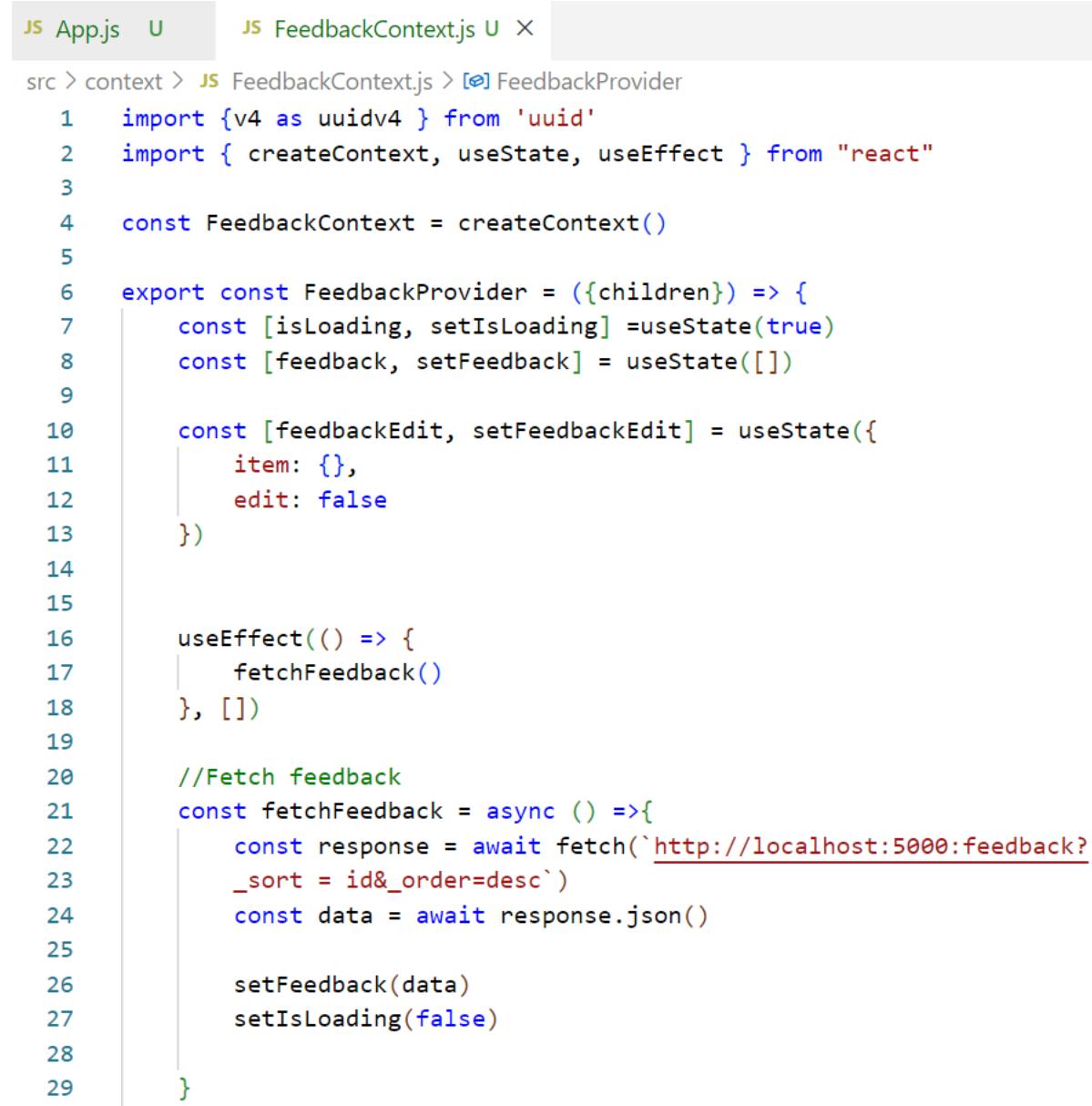
```
> keep-note@0.1.0 dev
> concurrently "npm run server" "npm start"
```

```
[0]
[0] > keep-note@0.1.0 server
[0] > json-server --watch db.json --port 5000
[0]
[1]
[1] > keep-note@0.1.0 start
[1] > react-scripts start
[1]
[0]
[0] \{^_^\}/ hi!
[0]
[0] Loading db.json
[0] Done
[0]
[0] Resources
[0] http://localhost:5000/feedback
[0]
[0] Home
[0] http://localhost:5000
[0]
[0] Type s + enter at any time to create a snapshot of the database
[0] Watching...
[0]
[1] Line 6:24: Unexpected empty object pattern no-empty-pattern
```



React JS

4. Fetch Data From JSON-Server Backend



The image shows a code editor interface with two tabs: "App.js" and "FeedbackContext.js". The "FeedbackContext.js" tab is active, displaying the following code:

```
src > context > JS FeedbackContext.js U X
src > context > JS FeedbackContext.js > [?] FeedbackProvider
1 import {v4 as uuidv4 } from 'uuid'
2 import { createContext, useState, useEffect } from "react"
3
4 const FeedbackContext = createContext()
5
6 export const FeedbackProvider = ({children}) => {
7   const [isLoading, setIsLoading] =useState(true)
8   const [feedback, setFeedback] = useState([])
9
10  const [feedbackEdit, setFeedbackEdit] = useState({
11    item: {},
12    edit: false
13  })
14
15
16  useEffect(() => {
17    fetchFeedback()
18  }, [])
19
20 //Fetch feedback
21 const fetchFeedback = async () =>{
22   const response = await fetch(`http://localhost:5000:feedback?
23   _sort = id&_order=desc`)
24   const data = await response.json()
25
26   setFeedback(data)
27   setIsLoading(false)
28
29 }
```

React JS

```
JS App.js U JS FeedbackContext.js U X
src > context > JS FeedbackContext.js > [?] FeedbackProvider

31
32     //Add feedback
33     const addFeedback =(newFeedback) =>{
34         newFeedback.id = uuidv4()
35         setFeedback([newFeedback, ...feedback])
36
37     }
38     //Delete Feedback
39     const deleteFeedback = (id) => {
40         if(window.confirm('Are you sure you want to delete?')){
41             setFeedback(feedback.filter((item) => item.id !== id))
42         }
43     }
44
45     // Update feedback item
46
47     const updateFeedback = (id, updItem) =>{
48         setFeedback(feedback.map((item) => item.id === id? { ...
49             item, ...updItem} : item))
50     }
51
52
53     // Set item to be updated
54     const editFeedback = (item) =>{
55         setFeedbackEdit({
56             item,
57             edit:true
58         })
59     }

```

```
JS App.js U JS FeedbackContext.js U X
src > context > JS FeedbackContext.js > [?] FeedbackProvider

60     return <FeedbackContext.Provider value={[
61         feedback,
62         deleteFeedback,
63         addFeedback,
64         editFeedback,
65         feedbackEdit,
66         updateFeedback,
67         isLoading,
68     ]}>
69     {children}
70   </FeedbackContext.Provider>
71 }
72 export default FeedbackContext
```

React JS

5. Spinner Component

The screenshot shows a code editor interface with three tabs at the top: 'App.js' (marked with a 'U'), 'FeedbackContext.js' (marked with a 'U'), and 'FeedbackLi...'. Below the tabs, a breadcrumb navigation bar indicates the file structure: 'src > components > shared > Spinner.jsx'. The main area displays the code for the 'Spinner' component:

```
1 import '../assets/spinner.gif'
2
3 function Spinner() {
4   return <img
5     src={Spinner}
6     alt='Loading...'
7     style={{
8       width: '100px',
9       margin: 'auto',
10      display:'block'
11    }}/>
12 }
13
14 export default Spinner
```

React JS

```
JS App.js U   JS FeedbackContext.js U   JS FeedbackList.jsx U X   JS Spinner.jsx U
src > components > JS FeedbackList.jsx > FeedbackList
1  import {motion, AnimatePresence} from 'framer-motion'
2  import { useContext } from 'react'
3  // import PropTypes from 'prop-types'
4  import FeedbackItem from './FeedbackItem'
5  import Spinner from './shared/Spinner'
6  import FeedbackContext from '../context/FeedbackContext'
7
8  function FeedbackList({ }) {
9      const {feedback, isLoading} = useContext(FeedbackContext)
10     if(!isLoading && !feedback || feedback.length ===0){
11         return <p> No Feedback Yet</p>
12     }
13
14     return isLoading ?
15         (<Spinner/>) :(
16             <div className="feedback-list">
17                 <AnimatePresence>
18                     {feedback.map((item) => (
19                         <motion.div
20                             key={item.id}
21                             initial={{ opacity: 0}}
22                             animate={{opacity: 1}}
23                             exit={{opacity: 0}}>
24
25                             <FeedbackItem
26                                 key= {item.id}
27                                 item = {item}
28
29                         />
30                         </motion.div>
31                     )))
32                 </AnimatePresence>
33
34             </div>
35         )
36 }
```

React JS

JS App.js U JS FeedbackContext.js U X JS FeedbackList.jsx U JS Spinner.jsx U

```
src > context > JS FeedbackContext.js > [?] default
  1 import {v4 as uuidv4 } from 'uuid'
  2 import { createContext, useState, useEffect } from "react"
  3
  4 const FeedbackContext = createContext()
  5
  6 export const FeedbackProvider = ({children}) => {
  7   const [isLoading, setIsLoading] =useState(true)
  8   const [feedback, setFeedback] = useState([])
  9
 10   const [feedbackEdit, setFeedbackEdit] = useState({
 11     item: {},
 12     edit: false
 13   })
 14
 15
 16   useEffect(() => {
 17     fetchFeedback()
 18   }, [])
 19
 20   //Fetch feedback
 21   const fetchFeedback = async () =>{
 22     const response = await fetch(`http://localhost:5000/feedback?
 23     _sort=id&_order=desc`)
 24     const data = await response.json()
 25
 26     setFeedback(data)
 27     setIsLoading(false)
 28   }
 29 }
}
return <FeedbackContext.Provider value={{
  feedback,
  deleteFeedback,
  addFeedback,
  editFeedback,
  feedbackEdit,
  updateFeedback,
  isLoading,
}}>
```

React JS

6. Add Feedback & Setting a Proxy

The screenshot shows a code editor with three tabs open:

- FeedbackContext.js**: Contains code for a context provider. It includes a state object with properties `edit` and `feedback`, and a `useEffect` hook that calls `fetchFeedback()`. The `fetchFeedback` function uses `await fetch('/feedback?_sort=')` to get data.
- package.json**: A JSON file defining the project metadata. It includes fields for name, version, private status, proxy URL (`"proxy": "http://localhost:5000"`), and dependencies.
- FeedbackList.js**: A component that sets the `feedback` state to the value returned by `fetchFeedback()`.

The screenshot shows a code editor with two tabs open:

- App.js**: Contains code for the main application component. It includes a `useEffect` hook that calls `addFeedback(newFeedback)` when the `edit` state changes.
- FeedbackContext.js**: Contains the context provider code from the previous screenshot.

React JS

7. Update & Delete From JSON-Server

```
src > context > FeedbackContext.js > [FeedbackProvider]
46     }
47     //Delete Feedback
48     const deleteFeedback = async(id) => {
49         if(window.confirm('Are you sure you want to delete?')) {
50             await fetch(`/feedback/${id}`, {method: 'DELETE'})
51             setFeedback(feedback.filter((item) => item.id !== id))
52         }
53     }
54
55     // Update feedback item
56
57     const updateFeedback = async(id, updItem) =>{
58         const response = await fetch(`/feedback/${id}`,{
59             method: 'PUT',
60             headers:{
61                 'Content-Type':'application/json',
62             },
63             body: JSON.stringify(updItem),
64         })
65
66         const data = await response.json()
67         setFeedback(feedback.map((item) => item.id === id? { ...
68             item, ...data} : item))
69     }
70 }
```

React JS