

dit Selection View Go Run Terminal Help

```
JS App.js X JS AddPlan.js JS ViewPlan.js
src > JS App.js > App
1 import React from 'react';
2 import './App.css';
3 import { BrowserRouter, Routes, Route, Link, Outlet, Router } from 'react-router-dom';
4 import AddPlan from './AddPlan';
5 import ViewPlan from './ViewPlan';
6
7 function App() {
8   return <React.Fragment>
9
10  <BrowserRouter>
11    <div className="container-fluid">
12      <nav data-testid="navbar" className="navbar navbar-expand-lg navbar-light bg-custom">
13        <span className="navbar-brand">TeleCall Prepaid </span>
14        <ul className="navbar-nav">
15          {/* Bind the CSS class "nav-link" to the links */}
16          <li className="nav-item">
17            <Link to="/addPlan" data-testid="addplan-link" className="nav-link">Add Plan</Link>
18
19          </li>
20          <li className="nav-item">
21            <Link to="/viewPlan" data-testid="viewplan-link" className="nav-link">View Plan</Link>
22          </li>
23        </ul>
24      </nav>
25      <Routes>
26        <Route path='/' element={<AddPlan/>}/>
27        <Route path='/addPlan' element={<AddPlan/>}/>
28        <Route path='/viewPlan' element={<ViewPlan/>}/>
29    </Routes>
30  </BrowserRouter>
31
```

aster® Q 9040

In 22 Col 16 Errors: 0 JITER IE (/) JavaScn

```
File Edit Selection View Go Run Terminal Help
js App.js x js AddPlan.js js ViewPlan.js
src > js App.js > App
  ↗
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
```

```
<ul>
  </li>
  <li className="nav-item">
    <Link to="/viewPlan" data-testid="viewplan-link" className="nav-link">View
      Plan</Link>
  </li>
</ul>
</nav>
<Routes>
  <Route path='/' element={<AddPlan/>}/>
  <Route path='/addPlan' element={<AddPlan/>}/>
  <Route path='/viewPlan' element={<ViewPlan/>}/>
<Route configuration */>
</Routes>
</div>
</BrowserRouter>
</React.Fragment>
}
export default App;
```

File Edit Selection View Go Run Terminal Help

JS App.js JS AddPlan.js JS ViewPlan.js X

src > JS ViewPlan.js > ViewPlan

```
1 import React, { useState, useEffect, useRef } from "react";
2 import "./App.css";
3 import axios from 'axios';
4
5 const url = "http://localhost:4000/plans";
6
7 function ViewPlan() {
8
9     const [planDetails, setPlanDetails] = useState([]);
10    const [errorMessage, setErrorMessage] = useState('');
11    const [successMessage, setSuccessMessage] = useState('')
12
13    //A collection of few messages the component displays.
14    // Wherever applicable, pls use the following json to display the messages instead of
15    //hardcoding.
16
17    const [messages] = useState({
18
19        "ERROR": "Something went wrong",
20        "DELETE_SUCCESS": "Deleted Successfully!",
21
22    })
23    useEffect(() => {
24        fetchDetails()
25    }, [])
26
27    //useEffect can be used if required to fetch the plan details when the component is
28    //mounted
29
30    function fetchDetails() {
31        axios.get(url)
32    }
33}
```

```
File Edit Selection View Go Run Terminal Help
JS App.js JS AddPlan.js JS ViewPlan.js X
src > JS ViewPlan.js > ViewPlan
  30
  31     .then(res=>{
  32         setPlanDetails(res.data)
  33     })
  34     .catch(error=>{
  35         setErrorMessage(messages.ERROR)
  36     })
  37     // Axios call be made here for fetching details .
  38     //If the Axios call is successful, set the state planDetails with the response received.
  39     //if the Axios call fails, Set the error message as "Something went wrong"
  40 }
  41
  42 const remove = (id) => {
  43     axios.delete('http://localhost:4000/plans/${id}')
  44     .then(res=>{
  45         setSuccessMessage(messages.DELETE_SUCCESS)
  46     })
  47     .catch(error=>{
  48         setErrorMessage(messages.ERROR)
  49         // setErrorMessage("Something went wrong")
  50     })
  51     // Delete the appointment from the database by placing HTTP delete request to the
  52     // url - http://localhost:4000/plans/<plan ID>
  53     // If the Axios call is successful, Set the success message as "Deleted Successfully!" .
  54     // If the Axios call fails, "Something went wrong".
  55 }
  56
  57
  58 return (
  59     <React.Fragment> <br />
  60     /*Display the messages based on condition */
  61     {successMessage?<p data-testid="successMsg" className="text-success message">
  62      (successMessage)</p>;""}
```

```
File Edit Selection View Go Run Terminal Help  
JS App.js JS AddPlan.js JS ViewPlan.js X  
src > JS ViewPlan.js > ViewPlan  
57  
58     return (  
59         <React.Fragment> <br />  
60         {/*Display the messages based on condition */}  
61         {successMessage?<p data-testid="successMsg" className="text-success message">  
62             {successMessage}</p>:""}  
63         {errorMessage?<p data-testid="errMsg" className="text-danger message">[errorMessage]  
64             </p>:""}  
65         <h6>Plan Details</h6>  
66  
67         <div className="row">  
68             {/*Display the cards*/}  
69             {planDetails.map(plan=>(  
70                 <div key={plan.id} className="card" style={{  
71                     marginLeft: '85px', width: '320px', marginBottom: '2rem',  
72                     backgroundColor: '#E2E8EC'  
73                 }>  
74                 <div className="card-body">  
75                     <b> Plan ID: </b><span data-testid="planId" className="card-title mb-2">  
76                         {plan.id}/* Plan Id */</span><br />  
77                     <b> Plan Value: </b><span data-testid="planValue" className="card-subtitle mb-2">  
78                         {plan.planValue}/* Plan Value */</span><br />  
79                     <b> Data: </b><span data-testid="data" className="card-subtitle mb-2">  
80                         {plan.data}/* data */</span><br />  
81                     <b> Unlimited Calls: </b><span data-testid="calls" className="card-subtitle mb-2">  
82                         {plan.calls}/* Unlimited Calls */</span><br />  
83                     <b> Addons: </b><span data-testid="addons" className="card-text">{plan.  
84                         addOns}/*Addons */</span><br />  
85                     /* {Invoke the appropriate method by passing the suitable argument } */  
86                     <button type="submit" data-testid="cancel" className="btn btn-primary">  
87                         Cancel  
88                     </button>  
89                 </div>  
90             )})</div>  
91         </React.Fragment>  
92     )  
93     </div>  
94     <div>  
95         <button type="button" data-testid="cancel" className="btn btn-primary">  
96             Cancel  
97         </button>  
98     </div>  
99 
```

ViewPlan.js - reactapp - Visual Studio Code

File Edit Selection View Go Run Terminal Help

App.js AddPlan.js ViewPlan.js

```
src > ViewPlan.js > ViewPlan
  73   <div className="card-body">
  74     <b> Plan ID: </b><span data-testid="planId" className="card-title mb-2">
  75       {plan.id}/* Plan Id */</span><br/>
  76     <b> Plan Value: </b><span data-testid="planValue" className="card-subtitle mb-2">
  77       {plan.planValue}/* Plan Value */</span><br/>
  78     <b> Data: </b><span data-testid="data" className="card-subtitle mb-2">
  79       {plan.data}/* data */</span><br/>
  80     <b> Unlimited Calls: </b><span data-testid="calls" className="card-subtitle mb-2">
  81       {plan.calls}/* Unlimited Calls */</span><br/>
  82     <b> Addons: </b><span data-testid="addons" className="card-text">{plan.
  83       addOns}/*Addons */</span><br/>
  84       /* {Invoke the appropriate method by passing the suitable argument } */
  85     <button type="submit" data-testid="cancel" className="btn btn-primary"
  86       onClick={()=>remove(plan.id)}>Delete</button>
  87   </div>
  88   </div>
  89 <)
  90 <
  91 <
  92 export default ViewPlan;
  93
```

AddPlan.js - reactapp - Visual Studio Code

File Edit Selection View Go Run Terminal Help

JS App.js JS AddPlan.js X JS ViewPlan.js

```
src > JS AddPlan.js > ...  
1 import React, { useState } from 'react';  
2 import axios from 'axios';  
3  
4 const url = "http://localhost:4000/plans";  
5  
6 export default function AddPlan() {  
7     //State to hold the plan details that needs to be added  
8     const [planData, setPlanData] = useState({ planValue: '', data: '', calls: '', addons: '' });  
9  
10    //state to hold the individual validation errors of the form fields  
11    const [formErrors, setFormErrors] = useState({ planValue: "", data: "" });  
12  
13        //state variable to capture the Success Message  
14    const [successMessage, setSuccessMessage] = useState("");  
15  
16        //state variable to capture the Error Message  
17    const [errorMessage, setErrorMessage] = useState("");  
18  
19        //state variable to indicate whether user has given values to all the mandatory fields  
20        // of the form.  
21    const [mandatory, setMandatory] = useState(false);  
22  
23        // state variable to indicate whether all the values given to the form fields are valid  
24    const [valid, setValid] = useState(true);  
25  
26        //state that holds the min value that can be given to the 'planValue' input field  
27    const [minVal, setMinVal] = useState(200);  
28  
29        //state that holds the min value that can be given to the 'data' input field  
30    const [minData, setMinData] = useState(20);
```

AddPlan.js - reactapp - Visual Studio Code

File Edit Selection View Go Run Terminal Help

src > **AddPlan.js** > ...

```
28 //state that holds the min value that can be given to the 'data' input field
29 const [minData, setMinData] = useState(20);
30
31
32 //A collection of few messages that the component displays.
33 // Wherever applicable, pls use the following json to display the messages instead of
34 //hardcoding the messages.
35
36 const [messages]=useState({
37   "MANDATORY":"Enter all the form fields",
38   "ERROR":"Something went wrong",
39   "PLANVALUE_ERROR":"Plan Value cannot be less than 200",
40   "DATA_ERROR":"Data should be 20GB or more",
41 })
42 const submit = (e) => {
43
44   e.preventDefault();
45   if(planData.addOns==""||planData.calls==""||planData.data==""||planData.
46   planValue==""){
47     setMandatory(true)
48   }
49   else{
50     setMandatory(false)
51     let newPlan ={
52       planValue:planData.planValue,
53       data:planData.data,
54       calls:planData.calls,
55       addOns:planData.addOns
56     }
57     axios.post(url,newPlan)
58     .then(res=>{
59       console.log(res)
60       setMandatory(false)
61       setPlanData(res.data)
62     })
63   }
64 }
```

File Edit Selection View Go Run Terminal Help

```
JS App.js    JS AddPlan.js X  JS ViewPlan.js
src > JS AddPlan.js > ...
  52     data:planData.data,
  53     calls:planData.calls,
  54     addOns:planData.addOns
  55   }
  56   axios.post(url,newPlan)
  57   .then(res=>{
  58     setSuccessMessage(`Plan created successfully with Id ${res.data.id}`)
  59   })
  60   .catch(error=>{
  61     setErrorMessage(messages.ERROR)
  62   })
  63 }
  64 //if any of the fields in the form is empty, set mandatory state to true,
  65
  66 //else set it to false and add the new plan details to the database.(Axios call can be
  67 //made for the same) and
  68 //if the plan details are added successfully, set the appropriate message "Plan created
  69 //successfully with Id <Plan Id>".
  70 //else set the error message "Something went wrong".
  71
  72 const handleChange = (event) => {
  73
  74   setSuccessMessage("")
  75   setErrorMessage("")
  76   setMandatory(false);
  77   let { name, value } = event.target;
  78   // Set the details state
  79   setPlanData({...planData,[name]:value})
  80   switch (name) {
  81     case "planValue":
  82       if(value<minVal){
  83         setSuccessMessage("")
  84         setErrorMessage(`Plan value must be greater than or equal to ${minVal}.`)
  85         setMandatory(true);
  86       }
  87   }
  88 }
```

Ln 4, Col 43 Spaces: 2 UTF-8 LF () JavaScript

AddPlan.js > reactapp - Visual Studio Code

File Edit Selection View Go Run Terminal Help

App.js AddPlan.js ViewPlan.js

```
src > AddPlan.js > ...
77 // Set the details state
78 setPlanData({...planData,[name]:value})
79 switch (name) {
80   case "planValue":
81     if(value<minVal){
82       formErrors.planValue=messages.PLANVALUE_ERROR
83     }else{
84       formErrors.planValue=""
85       // setValid(true)
86     }
87   // Set the planValue of formErrors state to:
88   // 1.'Plan Value cannot be less than 200 ' if value of planValue state is less than
89   // value of minVal state
90   // 2.'' if value of planValue state is valid
91   break;
92
93 case "data":
94   if(value<minData){
95     formErrors.data=messages.DATA_ERROR
96   }else{
97     formErrors.data=""
98     // setValid(true)
99   }
100 // Set data of formErrors state to:
101 // 1.'Data should be 20GB or more' if value of data state is less than the value
102 // of minData state
103 // 2.'' if value of planValue state is valid
104   break;
105 default:
106   break;
```

AddPlan.js - reactapp - Visual Studio Code

File Edit Selection View Go Run Terminal Help

src > JS AddPlan.js > ...

```
104     break;
105     default:
106         break;
107     }
108     if(formErrors.data === "" || formErrors.planValue === ""){
109         setValid(true)
110     }
111     else{
112         setValid(false)
113     }
114     // If the value of all of the properties of formErrors state are equal to "", set value
115     // of valid state to true, else set it to false
116 };
117
118
119
120
121 return <React.Fragment>
122 <div>
123   <div className="row">
124     <div className="col-md-6 offset-md-3">
125       <br />
126       <div className="card">
127
128         <div className="card-header" style={{ backgroundColor: 'rgba(0, 33, 65, 0.65)', color: 'white' }}>
129           <h6>Create a new plan</h6>
130         </div>
131         <div className="card-body">
132           /* */
133           1 Write code to render form as per screenshot (use html form)
```

```
AddPlan.js - reactapp - Visual Studio Code
File Edit Selection View Go Run Terminal Help
JS App.js JS AddPlan.js X JS ViewPlan.js
src > JS AddPlan.js > ...
    color: 'white' })>
      <h6>Create a new plan</h6>
    </div>
    <div className="card-body">
    /* 
      1. Write code to render form as per screenshot (use html form)
      2. Display successMessage or errorMessage appropriately.
      3. The form Should be controlled form
      4. Event handling methods should be bound appropriately
      5. Invoke the appropriate method on form submission using <form> tag
    */
    <form className="form" data-testid="addPlan-form" onSubmit={(event)>submit(event)}>
      <div className="form-group">
        <label>Plan Value </label>
        <input type="number" data-testid="planValue" name="planValue" value={planData.planValue} onChange={handleChange} className="form-control" />
        /* Display the error corresponding to planValue field if any... */
        {formErrors.planValue?<span data-testid="planValue-error" className="text-danger">{formErrors.planValue}</span>:""}
      </div>
      <div className="form-group">
        <label>Data (GB/Month) </label>
        <input type="number" data-testid="planData" name="data" value={planData.data} onChange={handleChange} className="form-control" />
        /* Display the error corresponding to data field if any... */
      </div>
    </form>
  
```

AddPlans.js > reactapp > Visual Studio Code

File Edit Selection View Go Run Terminal Help

src > JS AddPlan.js > ...

```
149
150     <div className="form-group">
151         <label>Data (GB/Month) </label>
152         <input type="number" data-testid="planData" name="data" value={planData.
153             data} onChange={handleChange} className="form-control" />
154             /* Display the error corresponding to data field if any.. */
155             {formErrors.data?<span data-testid="planData-error"
156                 className="text-danger">{formErrors.data}</span>:""}
157
158     </div>
159
160
161     <div className="form-group">
162         <label>Unlimited Calls</label> <br />
163         <div className="form-check form-check-inline" >
164             <label className="form-check-label" htmlFor="yes">Yes</label>
165             <input className="form-check-input" id="yes" data-testid="calls-yes"
166                 type="radio" name="calls" value="Yes" onChange={handleChange}/>
167
168         </div>
169         <div className="form-check form-check-inline">
170             <label className="form-check-label" id="no" htmlFor="no">No</label>
171             <input className="form-check-input" data-testid="calls-no" type="radio"
172                 name="calls" value="No" onChange={handleChange} />
173
174         </div>
175     </div>
176     <div className="form-group">
177         <label>Add-ons</label>
```

AddPlan.js - reactapp - Visual Studio Code

File Edit Selection View Go Run Terminal Help

JS App.js JS AddPlan.js X JS ViewPlan.js

```
src > JS AddPlan.js > ...
169 <label className="form-check-label" id="no" htmlFor="no">No</label>
170 <input className="form-check-input" data-testid="calls-no" type="radio"
171 name="calls" value="No" onChange={handleChange} />
172 </div>
173 </div>
174
175 <div className="form-group">
176 <label>Add-ons</label>
177 <textarea name="addOns" data-testid="addons" cols="83" value={planData.
178 addOns} onChange={handleChange} className="form-control" />
179 </div>
180
181 <div className="form-group">
182 /* Bind the disabled attribute of the button to the value of 'valid'
183 state */
184 <button
185 type="submit"
186 className="btn btn-primary"
187 data-testid="button"
188 style={({
189 backgroundColor: "rgba(0, 33, 65)",
190 border: "rgba(0, 33, 65)",
191 fontSize: "15px",
192 })
193 disabled={!valid}
194 >
195 Add Plan
196 </button>
197 </div>
198 </form>
```

AddPlan.js - reactapp - Visual Studio Code

File Edit Selection View Go Run Terminal Help

src > JS AddPlan.js > ...

```
190   border: "rgba(0, 33, 65)",  
191   fontSize: "15px",  
192 }  
193 disabled={!valid}  
194 >  
195     Add Plan  
196     </button>  
197   </div>  
198 </form>  
199 </div>  
200  
201 /*Using the concept of conditional rendering,display success message, error  
messages related to required fields and axios calls */  
202  
203 {mandatory?<div data-testid="mandatory" className="text-danger">{messages.  
MANDATORY}</div>"}  
204  
205  
206 {successMessage?<div data-testid="success" className="text-success">  
{successMessage}</div>"}  
207  
208  
209 {errorMessage?<div data-testid="error" className="text-danger">{errorMessage}</div>"}  
210  
211  
212   </div>  
213 </div>  
214 </div>  
215 </div>  
216 </React.Fragment>
```