

```
JS App.js      JS Appointment.js      JS PatientPortal.js M X  {} db.json
src > JS PatientPortal.js > [o] PatientPortal
1 //v1
2 import axios from "axios";
3 import React, { useState } from "react";
4 import "./App.css";
5
6
7 const url = "http://localhost:4000/appointments/";
8
9 const PatientPortal = () => {
10  // state to hold the appointmentID, the value gets updated when user enters the value
11  // in the input field
12  const [appointmentID, setappointmentID] = useState("");
13  // state variable used to display the error message when the data is not
14  // found for the patientID entered
15  const [errorMessage, setErrorMessage] = useState("");
16  // state variable holds the appointment details
17  const [appointment, setAppointment] = useState([]);
18  // state variable used to display the error message when delete axios call fails
19  const [deletedFail, setDeleteFail] = useState("");
20  // state variable used to display the error message when delete axios call is successful
21  const [deleteSuccess, setDeleteSuccess] = useState("");
22
23  //A collection of few messages that the component displays.
24  // Wherever applicable, pls use the following json to display
25  // the messages instead of hardcoding the messages.
26
27  const [messages]=useState({
28
29    "ERROR":"Something went wrong",
30    "APPT_ID_ERROR":"Appointment ID not found!",
31    "DELETION_SUCCESS":"Deleted Successfully,refresh the page",
32  })
```

```
js App.js      js Appointment.js      js PatientPortal.js M X  db.json
src > js PatientPortal.js > [e] PatientPortal
28      "ERROR": "Something went wrong",
29      "APPT_ID_ERROR": "Appointment ID not found!",
30      "DELETION_SUCCESS": "Deleted Successfully, refresh the page",
31
32
33    }
34
35    const change = (event) => {
36      setappointmentID(event.target.value)
37      /*
38       | update the appointmentID
39      */
40      setErrorMessage("")
41      setAppointment([])
42    };
43
44    const fetchDetails = (e) => {
45
46
47      /* Fetch the appointment details for the specific appointment id
48      by placing HTTP get request to url - http://localhost:4000/appointments/<Appointment ID>
49      and set the appropriate state */
50      // Set the appropriate error message "Appointment ID not found!" as per the screenshot
51
52      e.preventDefault();
53      axios.get("http://localhost:4000/appointments/" + appointmentID)
54        .then(res=>{
55          setAppointment(res.data)
56          console.log(appointment);
57        })
58        .catch(err=>{
59          setErrorMessage(messages.APPT_ID_ERROR)
60        })

```

```
src > JS App.js JS Appointment.js JS PatientPortal.js M X db.json  
      .then(res=>{  
        setAppointment(res.data)  
        console.log(appointment);  
      })  
      .catch(err=>{  
        setErrorMessage(messages.APPT_ID_ERROR)  
      })  
    }:  
  
  const cancel = (id) => {  
    axios.delete("http://localhost:4000/appointments/"+id)  
    .then(res=>{setDeleteSuccess(messages.DELETION_SUCCESS)})  
    .catch(err=>{setDeleteFail("Something went wrong")})  
  
    // Delete the appointment from the database by placing HTTP delete request to the  
    // url - http://localhost:4000/appointments/<Appointment ID>  
    // Set the appropriate success and error message "Deleted Successfully, refresh the page"  
    // and "Something went wrong" as per the screenshot  
  };  
  return (  
    <React.Fragment>  
      <div>  
        <div className="row">  
          <div className="col-md-6 offset-md-3">  
            <br />  
            <div className="card">  
              <div  
                className="card-header"  
                style={({  
                  backgroundColor: "rgba(0, 33, 65, 0.651)",  
                  color: "white",  
                })}>
```

```
JS App.js      JS Appointment.js    JS PatientPortal.js M X  () db.json
src > JS PatientPortal.js > [e] PatientPortal
84          |   backgroundColor: "rgba(0, 33, 65, 0.651)",
85          |   color: "white",
86          | }
87          |
88          | <h6>VIEW APPOINTMENT DETAILS</h6>
89          | </div>
90          | <div className="card-body">
91          | /*
92          | 1. Write code to render form as per screenshot
93          | 2. Display successMessage or errorMessage after deletion
94          | 3. Bind appropriate event handling methods
95          | 4. Invoke the appropriate method on form submission using the <form> tag
96          | */
97          | <form className="form" data-testid="patient-form"
98          |   onSubmit={(event)=>{fetchDetails(event)}}>
99          |   <div className="form-group">
100         |     <label>Appointment ID </label>
101         |     <input
102           |       type="text"
103           |       name="appointmentID"
104           |       data-testid="appointmentID"
105           |       required
106           |       className="form-control"
107           |       value={appointmentID}
108           |       onChange={change}
109           |     />
110         |   </div>
111         |   <div className="form-group">
112           |     <button
113             |       type="submit"
114             |       data-testid="submit-button"
115           |     </button>
116         |   </div>
117       | </form>
118     | </div>
119   | </div>
120 </div>
```

```
js App.js      js Appointment.js    js PatientPortal.js M X  () db.json
src > js PatientPortal.js > [e] PatientPortal
  ...
116         style={{ 
117           backgroundColor: "rgba(0, 33, 65)", 
118           border: "rgba(0, 33, 65)", 
119           fontSize: "15px", 
120         }} 
121       > 
122         Submit 
123       </button> 
124     </div> 
125   </form> 
126 </div> 
127 <br /> 
128 <p className="message text-danger" data-testid="errMsg"> 
129   {errorMessage} 
130 </p> 
131 
132 /* {display the appropriate error message (Using conditional rendering)} */ 
133 /* If the data is available, display the data in tabular format using conditional 
134   /* Sample data of appointment state - { 
135     id: 9001, 
136     emailId: 'Sunny@gmail.com', 
137     date: '2022-10-30', 
138     location: 'Bangalore' 
139   } 
140   */ 
141   <table className="table"> 
142     <thead> 
143       <tr> 
144         <th>Email</th> 
145         <th>Location</th> 
146         <th>Date</th> 
147       </tr> 
148     </thead> 
149     <tbody> 
150       <tr> 
151         <td>Sunny@gmail.com</td> 
152         <td>Bangalore</td> 
153         <td>2022-10-30</td> 
154       </tr> 
155     </tbody> 
156   </table> 
157 
```

```
JS App.js      JS Appointment.js      JS PatientPortal.js  46
src > JS PatientPortal.js > [o] PatientPortal
144         <thead>
145             <tr>
146                 <th>Email</th>
147                 <th>Location</th>
148                 <th>Date</th>
149             </tr>
150         </thead>
151         <tbody>
152             {/* render the appointment details */}
153             <tr>
154                 <td data-testid="app-email">{appointment.emailId}</td>
155                 <td data-testid="app-location">{appointment.location}</td>
156                 <td data-testid="app-date">{appointment.date}</td>
157                 <td>
158                     {/* {Invoke the appropriate method by passing the suitable argument } */}
159                     <button data-testid="cancel-button" onClick={()=>{cancel(appointment.id)}}>
160                         Cancel
161                     </button>
162                 </td>
163             </tr>
164         </tbody>
165     </table>
166 
167     {/* {display the appropriate success and error messages(Using conditional rendering)} */
168     {deleteSuccess?<p className="message text-success" data-testid="deleteSuccessMsg">
169         {deleteSuccess}
170     </p>"}
171 
172     {deletedFail?<p className="message text-danger" data-testid="deleteFailMsg">
173         {deletedFail}
174     </p>"}
175 }
```

Ln 135, Col 13 Spaces: 2 UTF-8 LF (JavaScript

```
JS App.js      JS Appointment.js    JS PatientPortal.js M X  () db.json
src > JS PatientPortal.js > (e) PatientPortal
159          <button data-testid="cancel-button" onClick={()=>{cancel(appointment.id)}}
160          |   Cancel
161          |   </button>
162          |   </td>
163          |   </tr>
164          |   </tbody>
165          </table>
166
167      /* {display the appropriate success and error messages(Using conditional renderi
168      {deleteSuccess?<p className="message text-success" data-testid="deleteSuccessMsg">
169      | {deleteSuccess}
170      </p>"}
171
172
173      {deletedFail?<p className="message text-danger" data-testid="deleteFailMsg">
174      | {deletedFail}
175      </p>"}
176
177      </div>
178      </div>
179      </div>
180      </React.Fragment>
181  );
182 };
183
184 export default PatientPortal;
185
```

```
JS App.js      JS Appointment.js M X   JS PatientPortal.js    () db.json
src > JS Appointment.js > (e) Appointment
1 //v1
2 import axios from "axios";
3 import React, { useState } from "react";
4
5
6 const url = "http://localhost:4000/appointments";
7
8 const Appointment = () => [
9   // state variable holding the form values. When user enters the values the state gets
10  // updated
11  const [details, setDetails] = useState({
12    emailId: "",
13    location: "",
14    date: ""
15  });
16  // state variables for displaying validation errors for email and date form fields
17  const [formErrors, setFormErrors] = useState({email:'',date:''});
18  // state variable to display success message once the appointment is booked
19  const [successMessage, setSuccessMessage] = useState("");
20  // state variable to display error message when there is some error with appointment
21  // booking
22  const [errorMessage, setErrorMessage] = useState("");
23  // state variable to display the mandatory field error meesage
24  const [mandatory, setMandatory] = useState(false);
25  // state variable used to disable the button when any of the given form values is invalid
26  const [valid, setValid] = useState(true);
27
28  //A collection of few messages that the component displays.
29  // Wherever applicable, pls use the following json to display the messages instead of
30  // hardcoding the messages.
31
32  const [messages]=useState({
```

```
JS App.js      JS Appointment.js M X   JS PatientPortal.js    () db.json
src > JS Appointment.js > (e) Appointment
28   collection of few messages that the component displays,
29   // Wherever applicable, pls use the following json to display the messages instead of
30   // hardcoding the messages.
31
32   const [messages]=useState({
33     "MANDATORY": "Enter all the form fields",
34     "ERROR": "Something went wrong",
35     "DATE_ERROR": "Date should be greater than today's date",
36     "EMAIL_ERROR": "Invalid email",
37   }
38   const submit = (e) => {
39     e.preventDefault();
40     if(details.emailId === "" || details.location === "" || details.date === "") {
41       setMandatory(true)
42     } else {
43       axios.post(url, details)
44         .then(res=>{
45           setSuccessMessage(`Appointment booked successfully. Appointment id is: ${res.data.id}`)
46         })
47         .catch(err=>{
48           setErrorMessage(messages.ERROR)
49         })
50     }
51   }
52   // check whether all the form fields are entered.
53   // If form fields are not entered set the mandatory state variable.
54   // if the form fields values are entered then make axios call
55   // pass the appropriate state as data to the axios call
56   // If the axios call is successful, set the appropriate message.
57   // Appointment booked successfully. Appointment id is: `${id}` as per the screen shot
58   // If the axios call is not successful, set the error message. Something went wrong.
59
```

```
js App.js      js Appointment.js M X  js PatientPortal.js    () db.json
src > js Appointment.js > (e) Appointment > (e) handleChange
57      // APPOINTMENT BOOKED SUCCESSFULLY. APPOINTMENT ID IS: 5101 AS PER THE SCREEN SHOT
58      // If the axios call is not successful, set the error message. Something went wrong.
59  };
60
61  const handleChange = (event) => {
62
63    setSuccessMessage("")
64    setErrorMessag(" ")
65    setMandatory(false);
66
67
68  let { name, value } = event.target;
69  // Set the details state
70  setDetails({...details,[event.target.name]: event.target.value})
71  let regex = /^[^\w]+@[^\w]+\.\w+$/i;
72  let errors = formErrors;
73  switch (name) {
74    // if email is invalid set formErrors field with message 'Invalid email' and ''
75    // if email is valid
76    case "emailId":
77      if (!regex.test(value)) {
78        formErrors.email='Invalid email';
79        setValid(true);
80      } else {
81        formErrors.email="";
82        setValid(false);
83      }
84      break;
85    case "date":
86      let formdate = new Date(value);
87      let todayDate = new Date();
88      if(formdate<=todayDate){
89        formErrors.date="Date should be greater than today's date";

```

```
js App.js      js Appointment.js M X  js PatientPortal.js  db.json
src > js Appointment.js > (e) Appointment > (e) handleChange
  04
  05      case "date":
  06          let formdate = new Date(value);
  07          let todayDate = new Date();
  08          if(formdate<=todayDate){
  09              formErrors.date="Date should be greater than today's date";
  10              setValid(true);
  11          } else {
  12              formErrors.date="";
  13              setValid(false);
  14          }
  15
  16          break;
  17      default:
  18          break;
  19      // similar to email check whether the date entered is greater than today's date
  20      // if date is invalid set formErrors field with message
  21      // 'Date should be greater than today's date' and '' if date is valid
  22  }
  23  if (Object.values(errors).every((value) => value === "")) {
  24      setValid(true);
  25  } else {
  26      setValid(false);
  27  }
  28  // set the formErrors state
  29  setFormErrors(errors);
  30 };
  31
  32 return (
  33     <React.Fragment>
  34         <div>
  35             <div className="row">
  36                 <div className="col-md-6 offset-md-3">
```

```
JS App.js      JS Appointment.js M X  JS PatientPortal.js  () db.json
src > JS Appointment.js > (e) Appointment
112     return (
113       <React.Fragment>
114         <div>
115           <div className="row">
116             <div className="col-md-6 offset-md-3">
117               <br />
118               <div className="card">
119                 <div
120                   className="card-header"
121                   style={({
122                     backgroundColor: "rgba(0, 33, 65, 0.651)",
123                     color: "white",
124                   })
125                 >
126                   <h6>Book an Appointment</h6>
127                 </div>
128                 <div className="card-body">
129                   /*
130                     1. Write code to render form as per screenshot
131                     2. Display successMessage or errorMessage after submission of form
132                     3. Form should be controlled
133                     4. Event handling methods should be binded appropriately
134                     5. Invoke the appropriate method on form submission using the <form> tag
135                   */
136                   <form className="form" data-testid="appointment-form"
137                     onSubmit={(event)=>{
138                       submit(event)
139                         /*Call appropriate method here.*})>
140                         <div className="form-group">
141                           <label>Email Id</label>
142                           <input
143                             data-testid="emailId"
```

```
src > App.js      Appointment.js M X  PatientPortal.js  db.json
140 <div className="form-group">
141   <label>Email Id</label>
142   <input
143     data-testid="emailId"
144     type="text"
145     name="emailId"
146     className="form-control"
147     value={details.emailId}
148     onChange={handleChange}
149   />
150 </div>
151 /* Check whether the email error is set if set display the
152 corresponding error message using conditional rendering
153 */
154
155 <span data-testid="email-error"
156   className="text-danger">{formErrors.email}</span>
157
158 <div className="form-group">
159   <label>Location</label>
160   <select
161     autoFocus
162     name="location"
163     data-testid="location"
164     className="form-control"
165     value={details.location}
166     onChange={handleChange}
167   >
168     <option value="">..Select..</option>
169     <option value="New York">New York</option>
170     <option value="Detroit">Detroit</option>
171   </select>

```

```
JS App.js      JS Appointment.js M X   JS PatientPortal.js    db.json
src > JS Appointment.js > (e) Appointment
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
<option value="">--Select--</option>
<option value="New York">New York</option>
<option value="Detroit">Detroit</option>
<option value="Manchester">Manchester</option>
<option value="Paris">Paris</option>
</select>
</div>

<div className="form-group">
  <label>Date</label>
  <input
    type="date"
    name="date"
    className="form-control"
    data-testid="date"
    value={details.date}
    onChange={handleChange}>
  />
</div>
/* Check whether the date error is set if set display the
corresponding error message using conditional rendering
*/
{formErrors.date}</span>

<div className="form-group">
```

```
src > JS Appointment.js M X JS PatientPortal.js () db.json
193
194     <span data-testid="date-error"
195         className="text-danger">{formErrors.date}</span>
196
197
198     <div className="form-group">
199         <button
200             type="submit"
201             className="btn btn-primary"
202             data-testid="button"
203             style={{
204                 backgroundColor: "rgba(0, 33, 65)",
205                 border: "rgba(0, 33, 65)",
206                 fontSize: "15px",
207             }}
208             disabled={!valid}
209         >
210             Submit
211         </button>
212         /* Bind the disabled attribute to the button to the valid state
213         variable */
214     </div>
215     </form>
216 </div>
217
218     /*Using the concept of conditional rendering,display success message,
219     error messages related to required fields and axios calls */
220     /* {if the form fields are not entered then set the message as
221     'Enter all the form fields'} */
222     {{mandatory}}?<div data-testid="mandatory" className="text-danger">
223         {{messages.MANDATORY}}
224     </div>"}}
```

```
JS App.js    JS Appointment.js M X  JS PatientPortal.js    db.json
src > JS Appointment.js > (e) Appointment
212             /* Bind the disabled attribute to the button to the valid state
213             variable */
214         </div>
215     </form>
216 </div>
217
218     /*Using the concept of conditional rendering,display success message,
219     error messages related to required fields and axios calls */
220     /* {if the form fields are not entered then set the message as
221     'Enter all the form fields'} */
222     {{mandatory}}<div data-testid="mandatory" className="text-danger">
223         {messages.MANDATORY}</div>;
224
225
226     {successMessage}<div data-testid="success" className="text-success">
227         {successMessage}</div>;
228
229
230     {errorMessage}<div data-testid="error" className="text-danger">
231         {errorMessage}</div>;
232
233     </div>
234     </div>
235   </div>
236 </div>
237 </React.Fragment>
238 );
239 }
240 export default Appointment;
241
```

```
JS App.js  X  JS Appointment.js M  JS PatientPortal.js  (1) db.json
src > JS App.js > App
1 //V1
2 import React from "react";
3 import "./App.css";
4 import { BrowserRouter, Link, Route, Routes } from "react-router-dom";
5 import Home from './Home';
6 import Appointment from './Appointment';
7 import PatientPortal from './PatientPortal';
8
9 function App() {
10   return (
11     <React.Fragment>
12       <BrowserRouter>
13         <nav data-testid="navbar" className="navbar navbar-expand-lg navbar-light bg-custom">
14           <span className="navbar-brand">IMed HMS</span>
15           <ul className="navbar-nav">
16             {/* Bind the 'to' attribute of the Link component as mentioned below
17             /home -> ABOUT US
18             /appointment - APPOINTMENT
19             /portal - PATIENT PORTAL
20             */}
21           <li className="nav-item">
22             <Link to="/home" className="nav-link" data-testid="home-link" >
23               ABOUT US
24             </Link>
25           </li>
26           <li className="nav-item">
27             <Link to="/appointment" className="nav-link" data-testid="appointment-link" >
28               APPOINTMENT
29             </Link>
30           </li>
31           <li className="nav-item">
```

```
JS App.js X JS Appointment.js M JS PatientPortal.js db.json
src > JS App.js > App
28     | APPPOINTMENT
29     |     </Link>
30     |     </li>
31     |     <li className="nav-item">
32     |         <Link to="/portal" className="nav-link" data-testid="patient-link" >
33     |             PATIENT PORTAL
34     |         </Link>
35     |     </li>
36     |     </ul>
37     | </nav>
38     <div className="container-fluid" style={{ marginTop: "60px" }}>
39         <Routes>
40             <Route path="/" element={<Home/>}/>
41             <Route path="/home" element={<Home/>}/>
42             <Route path="/appointment" element={<Appointment/>}/>
43             <Route path="/portal" element={<PatientPortal/>}/>
44         </Routes>
45         /* Configure the routes as mentioned below
46
47             / -> Home component
48             /home -> Home Component
49             /appointment -> Appointment component
50             /portal -> Patient Portal component
51
52         */
53     </div>
54     </BrowserRouter>
55     </React.Fragment>
56   );
57 }
58
59 export default App;
```