

App.js - Project - Visual Studio Code

File View Go Run Terminal Help

```
js  X  JS Cuisines.js  JS Book_Coupon.js  db.json
src > App.js > App
    <Book_Coupon>
        </Link>
    </li>
</ul>
</div>
</nav>

<Switch>
  /*
    Add the routes as mentioned below:
    1. redirect to /cuisines for default URL i.e. http://localhost:3000
    2. load Book_Coupon component for /book/:cuisineName/:id
    3. load Cuisines component for /cuisines
    4. redirect to /cuisines for any invalid URL e.g. http://localhost:3000/aboutus
  */
  <Route exact path="/" render={()=><Redirect to="/cuisines"/>}></Route>
  <Route exact path="/book/:cuisineName/:id" component={Book_Coupon}></Route>
  <Route exact path="/cuisines" component={Cuisines}></Route>
  <Route exact path="/" render={()=><Redirect to="/cuisines"/>}></Route>
</Switch>
</Router>
</React.Fragment>

export default App;
```

Shot on OnePlus
By Sameer

```
JS App.js      JS Cuisines.js X  JS Book_Coupon.js  db.json
Frontend > src > components > JS Cuisines.js > Cuisines > handleClick
27    }
28  };
29
30
31  /* Implemented - No Changes Required */
32  componentDidMount() {
33    this.getAllCuisines();
34  }
35
36  /* To Be Implemented */
37  handleClick = (cuisine) => [
38    /*
39     | It should set the state redirectStatus to true
40     | and selectedCuisineName and selectedCuisineId as the cuisineName
41     | and id of cuisine Object received as parameter into the function
42     */
43    this.setState({
44      redirectStatus:true,
45      selectedCuisineName:cuisine.cuisineName,
46      selectedCuisineId:cuisine.id
47    })
48  ]
49
50  /* To Be Modified - iterate over the cuisinesArray state to display the cuisines */
51  displayCuisines = () => {
52    return (
53
54      this.state.cuisinesArray.map((cuisine)=>
55        <div className="col-md-3" key={cuisine.id}>
56          <div className="card shadow">
57            <img src={cuisine.cuisineImagePath} alt={cuisine.id}
58              style={{height: '200px'}} className="card-img-top" />
59              /* Add imagePath as source and id as the alternate text for the cuisine
60              image */
```

Frontend > src > components > **Cuisines.js** > Cuisines

```
48
49
50  /* To Be Modified - iterate over the cuisinesArray state to
51   | display the cuisine cards */
52   | displayCuisines = () => {
53   |   return (
54
55     |     this.state.cuisinesArray.map((cuisine)=>
56       |       <div className="col-md-3" key={cuisine.id}>
57         |           <div className="card shadow">
58           |             <img src={cuisine.cuisineImagePath} alt={cuisine.id}
59             |             style={{height: '200px'}} className="card-img-top" />
60             |             /* Add imagePath as source and id as the alternate text for the cuisines Image */
61             |             /* <img style={{ height: '200px' }} className="card-img-top" /> */
62             |             <div className="card-body gradient-buttons text-center">
63               |               <h4 className="text" name="cuisineName">
64                 |                   /* Display the cuisineName here */
65                 |                   {cuisine.cuisineName}
66               |             </h4>
67
68               |               /* Invoke handleClick method by passing proper cuisine Object when Book Coupon bu
69               |               <button name="bookCoupon" className="btn btn-primary"
70                 |                 onClick={()=>{this.handleClick(cuisine)}}>
71                   |                   Book Coupon
72                 |               </button>
73               |             </div>
74             |           </div>
75           |         </div>
76         |       );
77       |     );
78     |   );
79   | };
80 }
```

Shot on OnePlus
By Sameer

Cuisines.js - Project - Visual Studio Code

Selection View Go Run Terminal Help

App.js

JS Cuisines.js X

JS Book_Coupon.js

{ db.json }

frontend > src > components > JS Cuisines.js > Cuisines > render

```
63     <h4 className="text" name="cuisineName">
64       /* Display the cuisineName here */
65       {cuisine.cuisineName}
66     </h4>
67
68   /* Invoke handleClick method by passing proper cuisine Object
69   when Book Coupon button is clicked */
70   <button name="bookCoupon" className="btn btn-primary"
71     onClick={()=>{this.handleClick(cuisine)}}>
72     Book Coupon
73   </button>
74   </div>
75 </div>
76 </div>
77
78   );
79 }
80
81 /* To Be Modified */
82 render() {
83   let { redirectStatus } = this.state;
84   if (redirectStatus) {
85     /* Redirect to /book/:cuisineName/:id by passing proper cuisineName as the
86     return(
87       <Redirect to={`/book/${this.state.selectedCuisineName}
88         /${this.state.selectedCuisineId}`}></Redirect>
89
90     )
91
92   }
93   return ()
```

Shot on OnePlus

By Sameer

Frontend > src > components > **Cuisines.js** > **Cuisines** > **render**

```
79     );
80   };
81
82   /* To Be Modified */
83   render() {
84     let { redirectStatus } = this.state;
85     if (redirectStatus) [
86       /* Redirect to /book/:cuisineName/:id by passing proper
87        | cuisineName as the route parameter */
88       return(
89         <Redirect to={`/book/${this.state.selectedCuisineName}
90           /${this.state.selectedCuisineId}`}></Redirect>
91       )
92     ];
93
94     return (
95
96       <div>
97         <div className="container mt-5">
98           <h2
99             className="display-4"
100            style={{ fontWeight: "bold", color: "black" }}
101          >
102            Kill your hunger with just a click!!!
103          </h2>
104
105          <h5 className="display-5" style={{ fontWeight: "700", color: "bla
106            Book a Coupon now to experience the authentic taste of your favo
107          </h5>
108        </div>
109        <div className="container mt-5">
110          <div className="row">
```

Shot on OnePlus

By Sameer

Cuisines.js - Project - Visual Studio Code

Edit Selection View Go Run Terminal Help

JS App.js JS Cuisines.js X JS Book_Coupon.js {} db.json

Frontend > src > components > JS Cuisines.js > Cuisines > render

```
92
93
94 return (
95
96   <div>
97     <div className="container mt-5">
98       <h2
99         className="display-4"
100        style={{ fontWeight: "bold", color: "black" }}
101      >
102        | Kill your hunger with just a click!!!
103      </h2>
104
105      <h5 className="display-5" style={{ fontWeight: "700", color: "black" }}>
106        | Book a Coupon now to experience the authentic taste of your favourite dishes
107      </h5>
108    </div>
109    <div className="container mt-5">
110      <div className="row">
111        /*
112          | if cuisinesArray has cuisines Objects - invoke the displayCuisines method
113          | else render the errorMessage
114        */
115        {this.state.cuisinesArray.length ?
116          this.displayCuisines():
117            <div className="alert alert-danger">
118              {this.state.errorMessage}</div>
119            }
120          </div>
121        </div>
122      </div>
123    
```

Shot on OnePlus

By Sameer

File Edit Selection View Go Run Terminal Help

```
JS App.js    JS Cuisines.js X    JS Book_Coupon.js    db.json
Frontend > src > components > JS Cuisines.js > Cuisines > render
102           Kill your hunger with just a click!!!
103           </h2>
104
105           <h5 className="display-5" style={{ fontWeight: "700", color: "black" }}>
106           | Book a Coupon now to experience the authentic taste of your favourite dishes
107           </h5>
108           </div>
109           <div className="container mt-5">
110           <div className="row">
111           /*
112           | if cuisinesArray has cuisines Objects - invoke the displayCuisines method
113           | else render the errorMessage
114           */
115           {this.state.cuisinesArray.length ?
116           this.displayCuisines():
117           <div className="alert alert-danger">
118           | {this.state.errorMessage}</div>
119           }
120           </div>
121           </div>
122           </div>
123
124       );
125   }
126 }
127
128 export default Cuisines;
129
```

Shot on OnePlus
By Sameer

```
55
34
35 }
36
37 /* To Be Implemented */
38 handleChange = (event) => {
39
40 /*
41     1. use the name and value from the event object
42     and set the received value to the corresponding bookCForm state property
43     2. call the validatefield method
44 */
45 const{name,value}=event.target;
46 this.setState({
47     bookCForm:{
48         ...this.state.bookCForm,
49         [name]:value,
50     },
51 }, (property) Book_Coupon.validateField: (fieldName: any, value: any) => void
52 this.validateField(name,value)
53 )
54 };
55
56 /* To Be Modified */
57 validateField = (fieldName, value) => {
58     let { bookCFormErrorMessage, bookCFormValid } = this.state;
59     switch (fieldName) {
60         case "name":
61
62         /*
63             1. if name field is empty it should set the error message as "Field required"
64             and set the bookCFormValid for the name as false.

```

Shot on OnePlus

By Sameer

JS App.js JS Cuisines.js JS Book_Coupon.js X db.json

Frontend > src > components > JS Book_Coupon.js > Book_Coupon > validateField

```
56  /* To Be Modified */
57  validateField = (fieldName, value) => {
58    let { bookCFormErrorMessage, bookCFormValid } = this.state;
59    switch (fieldName) {
60      case "name":
61        /*
62         1. if name field is empty it should set the error message as "Field required"
63         | and set the bookCFormValid for the name as false.
64         2. if name field has more or less than two words and one space,
65         | it should set the error message as "Please enter a valid name"
66         | and set the bookCFormValid for the name as false.
67         3. for valid input it should reset the error message and set the bookCFormValid
68         | for the name as true.
69       */
70
71      const customerRegex=/^([A-Za-z]+)\s([A-Za-z]+)$/
72      if(value === ""){
73        bookCFormErrorMessage.name = "Field Required";
74        bookCFormValid.name = false;
75      } else if(!value.match(customerRegex)){
76        bookCFormErrorMessage.name = "Please enter a valid name";
77        bookCFormValid.name = false;
78      } else{
79        bookCFormErrorMessage.name = "";
80        bookCFormValid.name = true;
81      }
82
83      break;
84      case "dishSelected":
85
86      /*
87       1. if dishSelected field is empty it should set the error message as "Please select
88       and set the bookCFormValid for dishSelected as false
89     */
90   }
91 }
```

Shot on OnePlus
By Sameer

dit Selection View Go Run Terminal Help

JS App.js JS Cuisines.js JS Book_Coupon.js X db.json

Frontend > src > components > JS Book_Coupon.js > Book_Coupon > validateField

```
83     break;
84     case "dishSelected":
85
86     /*
87      1. if dishSelected field is empty it should set the error message as
88      | "Please select a dish"
89      | and set the bookCFormValid for dishSelected as false.
90      2. for valid input it should reset the error message and set the bookCFormValid
91      for dishSelected as true.
92     */
93     if (value === "") {
94       bookCFormErrorMessage.dishSelected = "Please select a dish";
95       bookCFormValid.dishSelected = false;
96     } else {
97       bookCFormErrorMessage.dishSelected = "";
98       bookCFormValid.dishSelected = true;
99     }
100
101    break;
102    case "noOfPlates":
103
104    /*
105      1. if noOfPlates field is empty it should set the error message as "Field required"
106      | and set the bookCFormValid for noOfPlates as false.
107      2. if noOfPlates field has value more than 4 or less than 1,
108      | it should set the error message as "Only 1 to 4 plates can be booked"
109      | and set the bookCFormValid for noOfPlates as false.
110      3. for valid input it should reset the error message and set the bookCFormValid
111      | for the noOfPlates as true.
112    */
113    if(value === ""){
```

Shot on OnePlus
By Sameer

Book_Coupon.js - Project - Visual Studio Code

Edit Selection View Go Run Terminal Help

JS App.js JS Cuisines.js JS Book_Coupon.js X db.json

Frontend > src > components > JS Book_Coupon.js > Book_Coupon > validateField

```
112
113     */
114     if(value === ""){
115         bookCFormErrorMessage.noOfPlates= "Field required";
116         bookCFormValid.noOfPlates=false;
117     } else if (value <1 || value > 4){
118         bookCFormErrorMessage.noOfPlates= "Should accept a value between 1 to 4";
119         bookCFormValid.noOfPlates=false;
120     } else{
121         bookCFormErrorMessage.noOfPlates="";
122         bookCFormValid.noOfPlates=true;
123     }
124     break;
125
126 case "usingDate":
127     let today = new Date().setHours(0, 0, 0, 0);
128     /*
129      1. pass the value to the date constructor and return the date object
130      in the usingDate variable.
131      2. if usingDate field is empty it should set the error message as "Field required"
132      and set the bookCFormValid for usingDate as false.
133      3. if usingDate field has past date,
134          it should set the error message as "Coupon can be booked for current date
135          and future date only"
136          and set the bookCFormValid for usingDate as false.
137      4. for valid input it should reset the error message and set the bookCFormValid
138          for usingDate as true.
139
140     */
141
142     break;
143 default:
```

Shot on OnePlus

By Sameer

File Selection View Go Run Terminal Help

JS App.js JS Cuisines.js JS Book_Coupon.js X db.json

Frontend > src > components > JS Book_Coupon.js > Book_Coupon > validateField

```
140
141
142     break;
143     default:
144         break;
145     }
146
147     bookCFormValid.buttonActive =
148         bookCFormValid.name &&
149         bookCFormValid.usingDate&&
150         bookCFormValid.dishSelected&&
151         bookCFormValid.noOfPlates;
152     this.setState({
153         bookCFormErrorMessage: bookCFormErrorMessage,
154         bookCFormValid: bookCFormValid,
155         successMessage: "",
156         errorMessage: "",
157     });
158 };
159
160 /* To Be Implemented */
161 bookCoupon = (event) => {
162     /*
163         1. should prevent the page reload on form submission
164         2. should reset successMessage and errorMessage state as ""
165         3. should make a POST request to http://localhost:4000/bookings
166         4. in success case, it should set successMessage as Coupon Booked
167         5. in error case, it should set errorMessage as 'Something went wrong'
168     */
169     event.preventDefault();
170     const {bookForm} = this.state;
171     this.setState({
172         successMessage: " ".
```

Shot on OnePlus

By Sameer

Edit Selection View Go Run Terminal Help

JS App.js JS Cuisines.js JS Book_Coupon.js X {} db.json

Frontend > src > components > JS Book_Coupon.js > Book_Coupon > bookCoupon

```
159
160  /* To Be Implemented */
161  bookCoupon = (event) => [
162    /*
163      1. should prevent the page reload on form submission
164      2. should reset successMessage and errorMessage state as ""
165      3. should make a POST request to http://localhost:4000/bookings with
166          bookCForm as the request body
167      4. in success case, it should set successMessage as
168          | Coupon Booking Successful!! Your Bill is of Rs. <<calculate bill>>
169      5. in error case, it should set errorMessage as 'Something went wrong'
170    */
171    event.preventDefault();
172    const{bookCForm}=this.state;
173    this.setState({
174      successMessage: " ",
175      errorMessage: " ",
176    })
177    axios.post(http://localhost:4000/bookings, bookCForm)
178    .then((response)=>{
179      successMessage: `Coupon Booking Successful!!`;
180      errorMessage: `Something went wrong`
181    })
182  ];
183  getDishes = async () => {
184    try {
185      let cuisineData = await axios.get(
186        http://localhost:4000/cuisines/ + this.state.cuisineId
187      );
188      this.setState({ selectedCouponObj: cuisineData.data });
189    } catch (err) {
```

Shot on OnePlus

By Sameer

```
pp.js          JS Cuisines.js    JS Book_Coupon.js ×  db.json
App.js > src > components > JS Book_Coupon.js > Book_Coupon > bookCoupon

componentDidMount() {
  this.getDishes();
}

render() {
  let { bookCForm, bookCFormErrorMessage } = this.state;
  return (
    <Fragment>
      <div className="container-fluid">
        <div className="row">
          <div className="col-md-6 offset-md-3">
            <div
              className="card mt-5"
              style={{ backgroundColor: "rgb(245, 243, 173)" }}>
              <div className="card-header text text-center shadow" style={{ backgroundColor: "#f2f2f2" }}>
                <h3 id="card-title">
                  Book Your Favourite /* Display selected Cuisine */{bookCForm.cuisineName}
                </h3>
              </div>
              <div className="card-body">
                /* ... */
                1. create the form with the fields as shown in QP
                2. All the form fields should be bound to the corresponding state
                3. All the form fields should have proper name and id
                4. handleChange() method should be invoked on change event
                5. name field should be of type text
                6. name error Message displaying tag should have id as nameError
                7. usingDate field should be of type date
                8. usingDate error Message displaying tag should have id as usingDateError
              </div>
            </div>
          </div>
        </div>
      </div>
    )
}
```

Shot on OnePlus
By Sameer

JS App.js JS Cuisines.js JS Book_Coupon.js X db.json

Frontend > src > components > JS Book_Coupon.js > Book_Coupon > render

```
201 render() {
202     let { bookCForm, bookCFormErrorMessage } = this.state;
203     return (
204         <Fragment>
205             <div className="container-fluid">
206                 <div className="row">
207                     <div className="col-md-6 offset-md-3">
208                         <div
209                             className="card mt-5"
210                             style={{ backgroundColor: "rgb(245, 243, 173)" }}>
211                         >
212                             <div className="card-header text text-center shadow"
213                                 style={{ backgroundColor: "rgb(255, 198, 0)" }}>
214                                 <h3 id="card-title">
215                                     Book Your Favourite /* Display selected Cuisine */
216                                     {bookCForm.cuisineName} Item
217
218                                 </h3>
219                             </div>
220                             <div className="card-body">
221                                 /*
```

1. create the form with the fields as shown in QP
2. All the form fields should be bound to the component
3. All the form fields should have proper name and type
4. handleChange() method should be invoked on change
5. name field should be of type text
6. name error Message displaying tag should have the correct message
7. usingDate field should be of type date
8. usingDate error Message displaying tag should have the correct message
9. dishSelected field should be a dropdown with the following options:
 - first option should be displayed as "Select Dish"
 - all other values should be populated dynamically

Shot on OnePlus

By Sameer

```
Selection View Go Run Terminal Help
App.js      JS Cuisines.js    JS Book_Coupon.js X  () db.json
intend > src > components > JS Book_Coupon.js > Book_Coupon > render
0 <i>body">
1
2 1. create the form with the fields as shown in QP
3 2. All the form fields should be bound to the corresponding bookCForm state property
4 3. All the form fields should have proper name and id - same as
| bookCForm state properties
5 4. handleChange() method should be invoked on change of values of every input field
6 5. name field should be of type text
7 6. name error Message displaying tag should have id as nameError and styling
| | using class p-2 alert-danger
8 7. usingDate field should be of type date
9 8. usingDate error Message displaying tag should have id as usingDateError and
| | styling using class p-2 alert-danger
10 9. dishSelected field should be a dropdown with the options as mentioned below:
| | - first option should be displayed as "Select Dish" and must have value as empty
| | string ("")
| | - all other values should be populated dynamically using the dishes array of
| | selectedCouponObj state
| | - e.g. if dishes array contains Rasogolla as an element, it should display as
| | Rasogolla in the dropdown and
| | - should have value as Rasogolla as well
11 10. dishSelected error message displaying tag should have id as dishSelectedError and
| | styling using class p-2 alert-danger
12 11. noOfPlates field should be of type number
13 12. noOfPlates error message displaying tag should have id as noOfPlatesError and
| | styling using class p-2 alert-danger
14 13. Confirm Booking button should have id as confirmBookingBtn
15 14. It should be disabled until buttonActive state property becomes true
16 15. Display errorMessage and successMessage on form submission appropriately
17 16. Both errorMessage and successMessage should not appear together
18 17. successMessage tag should have id as successMessage and styling using class p-2
| | alert alert-success
```

Shot on OnePlus

By Sameer

Frontend > src > components > JS Book_Coupon.js > Book_Coupon > render

```
246    13. Confirm Booking button should have id as confirmBookingBtn
247    14. It should be disabled until buttonActive state property becomes true
248    15. Display errorMessage and successMessage on form submission appropriately
249    16. Both errorMessage and successMessage should not appear together
250    17. successMessage tag should have id as successMessage and styling using class p-2
251    ||| alert alert-success
252    18. errorMessage tag should have id as errorMessage and styling using class p-2
253    ||| alert alert-danger
254    19. use proper styling classes to form fields and button to give appropriate look
255    ||| as required.

256  */
257 <form onSubmit={this.bookCoupon}>
258   <div className="form-group">
259     <label htmlFor="Name">
260       <h6>Name</h6>
261     </label>
262     <input type="text"
263       className="form-control"
264       id="name"
265       name="name"
266       value={bookCForm.name}
267       onChange={this.handleChange}/>
268     {bookCFormErrorMessage.name ? (
269       <div className="alert alert-danger" id="nameError">
270         {bookCFormErrorMessage.name}
271       </div>
272     ): (" ")}
273   </div>
274
275   <div className="form-group">
276     <label htmlFor="usingDate">
```

Shot on OnePlus
By Sameer

Frontend > src > components > **JS** Book_Coupon.js > **Book_Coupon** > **render**

```
275
276     <div className="form-group">
277         <label htmlFor="usingDate">
278             | <h6>Coupon using date</h6>
279         </label>
280         <input type="date"
281             className="form-control"
282             id="usingDate"
283             name="usingDate"
284             value={bookCForm.usingDate}
285             onChange={this.handleChange}
286         />
287         {bookCFormErrorMessage.usingDate ? (
288             <div className="alert alert-danger" id="usingDateError">
289                 {bookCFormErrorMessage.usingDate}
290             </div>
291         ) : (" ")}
292     }
293 </div>
294
295     <div className="form-group">
296         <label htmlFor="dishSelected">
297             | <h6>Select Dish</h6>
298         </label>
299         <select type="dropdown"
300             className="form-control"
301             id="dishSelected"
302             name="dishSelected"
303             value={bookCForm.dishSelected}
304             onChange={this.handleChange}>
305             {bookCFormErrorMessage.dishSelected ? (
306                 <div className="alert alert-danger" id="dishSelectedError">
```

Shot on OnePlus

By Sameer

Frontend > src > components > **js** Book_Coupon.js > Book_Coupon > render

```
293     </div>
294
295     <div className="form-group">
296         <label htmlFor="dishSelected">
297             <h6>Select Dish</h6>
298         </label>
299         <select type="dropdown"
300                 className="form-control"
301                 id="dishSelected"
302                 name="dishSelected"
303                 value={bookCForm.dishSelected}
304                 onChange={this.handleChange}>
305             {bookCFormErrorMessage.dishSelected ? (
306                 <div className="alert alert-danger" id="dishSelectedError">
307                     {bookCFormErrorMessage.dishSelected}
308                 </div>
309             ) : (" ")}
310         </select>
311     </div>
312
313     <div className="form-group">
314         <label htmlFor="noOfPlates">
315             <h6>Number of Plates</h6>
316         </label>
317         <input type="number"
318                 className="form-control"
319                 id="noOfPlates"
320                 name="noOfPlates"
321                 value={bookCForm.noOfPlates}
322                 onChange={this.handleChange}>
323             {bookCFormErrorMessage.noOfPlates ? (
324                 <div className="alert alert-danger" id="noOfPlatesError">
```

Shot on OnePlus

By Sameer

Edit Selection View Go Run Terminal Help

JS App.js

JS Cuisines.js

JS Book_Coupon.js X

{ db.json

Frontend > src > components > JS Book_Coupon.js > Book_Coupon > render

```
309     ): (" ")
310   }
311 </div>
312
313 <div className="form-group">
314   <label htmlFor="noOfPlates">
315     <h6>Number of Plates</h6>
316   </label>
317   <input type="number"
318     className="form-control"
319     id="noOfPlates"
320     name="noOfPlates"
321     value={bookCForm.noOfPlates}
322     onChange={this.handleChange}/>
323   {bookCFormErrorMessage.noOfPlates ? (
324     <div className="alert alert-danger" id="noOfPlatesError">
325       {bookCFormErrorMessage.noOfPlates}
326     </div>
327   ): (" ")
328 }
329 </div>
330
331 <div className="form-group">
332   <button className="btn-block form-control"
333     id="confirmBookingBtn"
334     disabled={this.state.bookCFormValid.buttonActive}
335     confirmBooking
336     disabled/>
337   </div>
338 </form>
339 <{this.state.successMessage ? (
340   <div className="alert alert-success" id="successMessage">
```

Shot on OnePlus

By Sameer

Book_Coupon.js - Project - Visual Studio Code

Edit Selection View Go Run Terminal Help

JS App.js JS Cuisines.js JS Book_Coupon.js X db.json

Frontend > src > components > Book_Coupon.js > Book_Coupon > render

```
327
328     }
329   </div>
330
331   <div className="form-group">
332     <button className="btn-block form-control"
333       id="confirmBookingBtn"
334       disabled={this.state.bookCFormValid.buttonActive}
335       confirmBooking
336       disabled/>
337   </div>
338 </form>
339 {this.state.successMessage ? (
340   <div className="alert alert-success" id="successMessage">
341     {this.state.successMessage}
342   </div>
343 ) : (" ")}
344 }
345 {this.state.errorMessage ? (
346   <div className="alert alert-danger" id="errorMessage">
347     {this.state.errorMessage}
348   </div>
349 ) : (" ")}
350 }
```

Shot on OnePlus
By Sameer

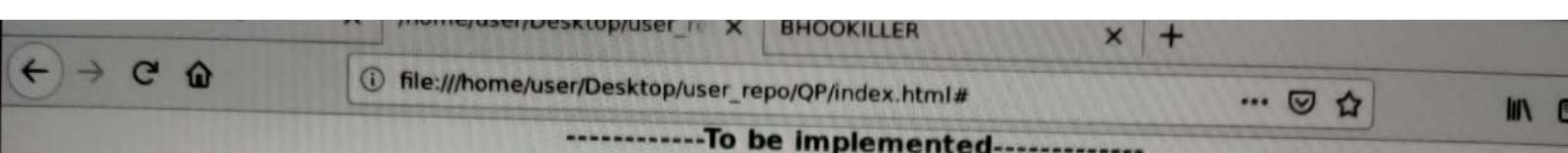
```
Frontend > src > components > Book_Coupon.js > Book_Coupon > render
321
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
```

```
    ...
    }
  </div>

  <div className="form-group">
    <button className="btn-block form-control"
      id="confirmBookingBtn"
      disabled={this.state.bookCFormValid.buttonActive}
      confirmBooking
      disabled/>
  </div>
</form>
{this.state.successMessage ?
  <div className="alert alert-success" id="successMessage">
    {this.state.successMessage}
  </div>
): (" ")
}
{this.state.errorMessage ?
  <div className="alert alert-danger" id="errorMessage">
    {this.state.errorMessage}
  </div>
): (" ")
}

  </div>
</div>
</div>
</div>
</div>
</Fragment>
);
}
```

Shot on OnePlus
By Sameer



Details of Bhookiller Front End Application

1. Configure the routes with the below information:

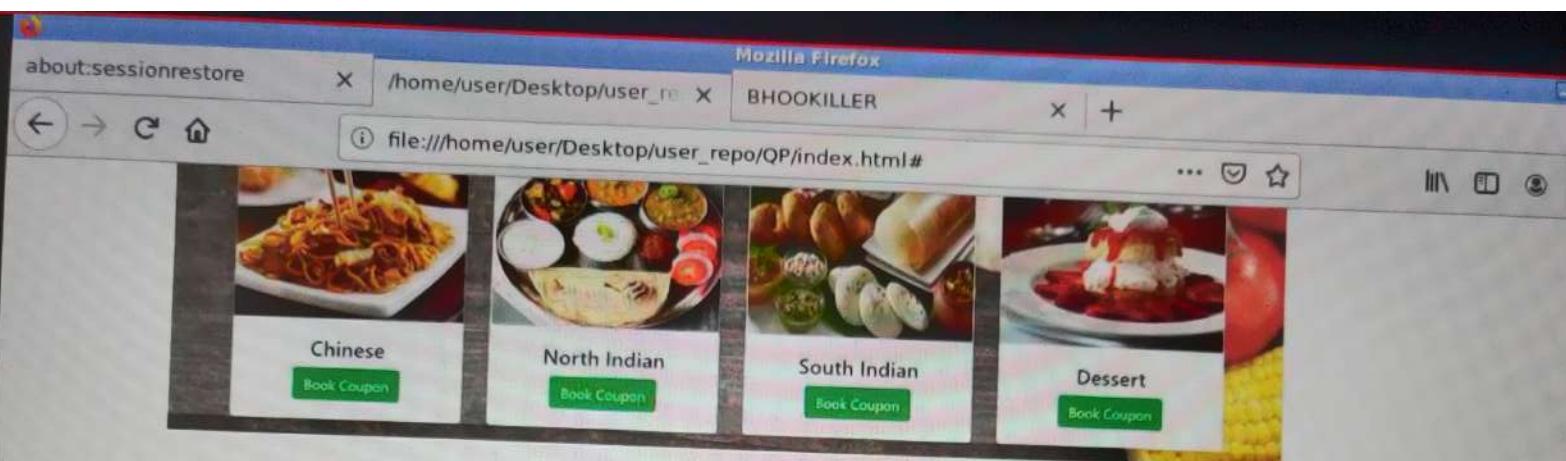
Route Path	Component
/cuisines	Cuisines Component
/book/:cuisineName/:id	Book_Coupon Component

- In case if an **Invalid URL** is provided, the user must be redirected to the **Cuisines Page**
- In case if the **Default(/) URL** is provided, the user must be redirected to the **Cuisines Page**

3. When **http://localhost:3000** is entered in browser, Destinations page should be displayed as below:



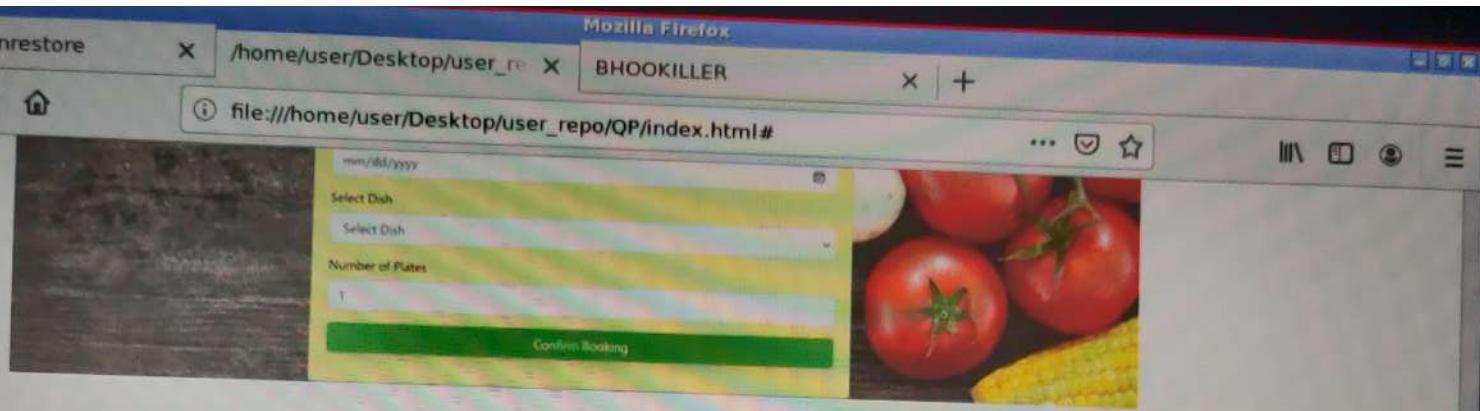
Shot on OnePlus
By Sameer



3. On Click of **Book Coupon** Button of any cuisines card, it should navigate to the **Book_Coupon** Component having the view as shown below:



Shot on OnePlus
By Sameer



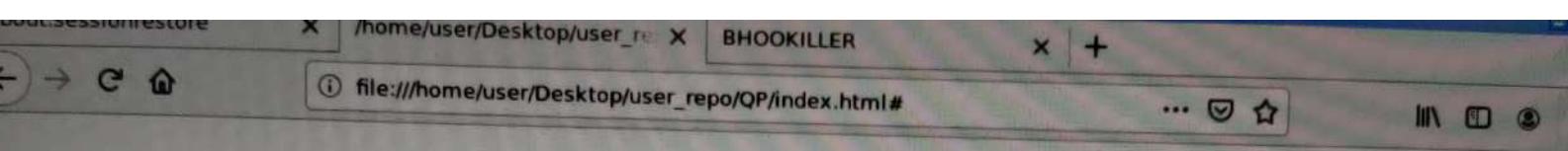
4. Code the validateField method to include the validations as mentioned:

Field name	form State Name	Validations	Default Value
Name	name	Field is mandatory Name must have two words with only alphabets and one space but should always start only with an alphabet. Example- John Doe	<Empty string i.e. ">"
Coupon using date	usingDate	Field is mandatory Package start date should be the current date or a future date.	<Empty string i.e. ">"
Select Dish	dishSelected	-----Already implemented-----	<Empty string i.e. ">"
Number Of Plates	noOfPlates	Field is mandatory Should accept a value between 1 to 4	1

- **Confirm Booking** Button should remain disabled until the form becomes valid

Execute

Shot on OnePlus
By Sameer



5. When **Book Coupon** button is clicked from cuisines Page, it should render the bookCForm as shown below:

A screenshot of a booking form titled 'Book Your Favourite Chinese Item'. The form consists of several input fields: 'Name' (placeholder 'John'), 'Coupon using date' (placeholder 'mm/dd/yyyy'), 'Select Dish' (dropdown menu open), 'Number of Plates' (value '1'), and a green 'Confirm Booking' button.

6. Validation Error messages should be rendered as below:

A screenshot of the same booking form as above, but with a validation error message. The 'Name' field is empty, and a red error message 'Field required' is displayed below it. The rest of the form fields are visible but empty.

Shot on OnePlus
By Sameer

6. Validation Error messages should be rendered as below:

Book Your Favourite Chinese Item

Name

Field required

Coupon using date
 mm/dd/yyyy
Field required

Select Dish

Please select a dish

Number of Plates

Field required

Book Your Favourite Chinese Item

Shot on OnePlus

By Sameer

Name

john

Field required

Confirm Booking

Book Your Favourite Chinese Item

Name

john

Please enter a valid name

Coupon using date

12/08/2022

Coupon can be booked for current date and future date only

Select Dish

Select Dish

Please select a dish

Number of Plates

7

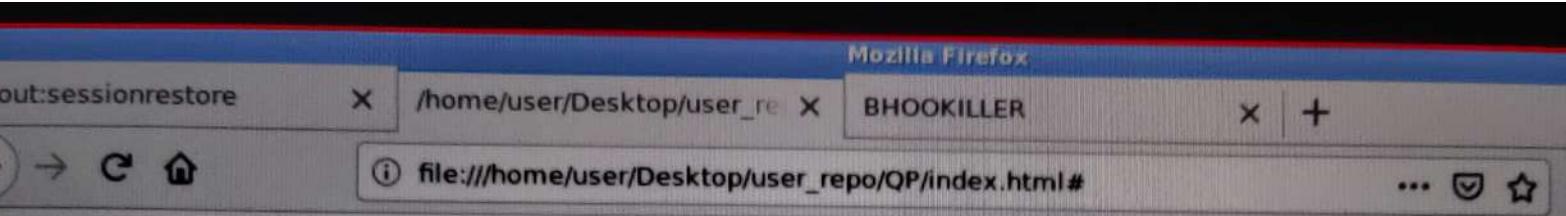
Only 1 to 4 plates can be booked

Confirm Booking

1. On successful execution, the success message should be disp

Shot on OnePlus

By Sameer



7. On successful execution, the success message should be displayed as shown below:

Book Your Favourite Chinese Item

Name
John Doe

Coupon using date
12/10/2022

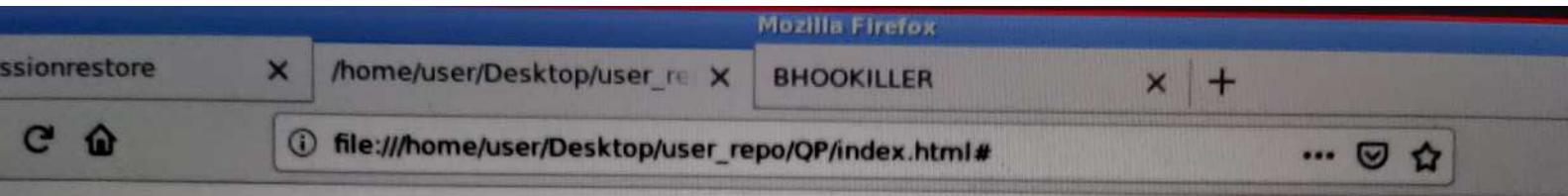
Select Dish
Egg fried-rice with chicken-chilly Price:90

Number of Plates
2

Confirm Booking

Coupon Booking Successful!!

Shot on OnePlus
By Sameer



8. In case if the booking fails due to server issues, it should render the error message as shown:

The image shows a booking form titled "Book Your Favourite Chinese Item". The form fields are as follows:

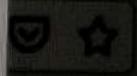
- Name: John Doe
- Coupon using date: 12/10/2022
- Select Dish: Egg fried-rice with chicken-chilly Price:90
- Number of Plates: 2

A green button at the bottom right is labeled "Confirm Booking". Below the form, a pink bar displays the error message "Something went wrong".

Shot on OnePlus
By Sameer

Result

X



Last modified on 25/11/2022 at 17:26:32

Module	Success	Failed	Total
Book_Coupon Component - noOfPlates field validation using validateField Method	2	1	3
Book_Coupon Component - usingDate field validation using validateField Method	0	2	2
Book_Coupon Component -- Testing Validation of name	1	1	2
Book_Coupon Component - bookCoupon Method	4	0	4
Book_Coupon Component -	2	0	2

Shot on OnePlus

By Sameer

handleChange Method

Book_Coupon Component -
form, button and messages 7 0 7

Book_Coupon Component -
noOfPlates field 5 0 5

Book_Coupon Component -
dishSelected field 4 0 4

Book_Coupon Component -
usingDate field 5 0 5

Book_Coupon Component -
name field 5 0 5

Book_Coupon Component -
Form Heading 1 0 1

Cuisines Component - Testing
render method 4 0 4

Cuisines Component - Testing
displayCuisines method 4 1 5

Shot on OnePlus

By Sameer

Cuisines Component - Testing 3 0 3

Book_Coupon Component - usingDate field	5	0	5
Book_Coupon Component - name field	5	0	5
Book_Coupon Component - Form Heading	1	0	1
Cuisines Component - Testing render method	4	0	4
Cuisines Component - Testing displayCuisines method	4	1	5
Cuisines Component - Testing handleClick Method	3	0	3
Testing Application Routing	4	0	4
Total	51	5	56

Shot on OnePlus
By Sameer

Close