

```
JS Locations.js
components > JS FarmVisit.js >
  this.state = {
    bookVisitFor: '',
    city: this.props.city,
    visitDate: '',
    phoneNumbe: '',
    customerNa: ''
  },
  bookVisitFor: {
    city: '',
    visitDate: '',
    phoneNumbe: '',
    customerNa: ''
  },
  bookVisitFor: {
    city: false,
    visitDate: '',
    customerNa: '',
    phoneNumbe: '',
    buttonActi: ''
  },
  successMessa: '',
  errorMessage: ''
}
}

/* To Be Implemented
handleChange = (e) => {
  /* should use
  const target = e.target;
  const value = target.type === 'checkbox' ? target.checked : target.value;
  const name = target.name;
  this.setState({[name]: value});
}
```

FarmVisit Component - phoneNumber field	1	1	2
FarmVisit Component - customerName field	2	0	2
FarmVisit Component - visitDate field	2	0	2
FarmVisit Component - city field	2	1	3
Locations Component - Testing render method	4	0	4
Locations Component - Testing displayLocations method	2	3	5
Locations Component - Testing handleClick Method	2	0	2
Testing Application Routing	2	1	3
Total	37	17	54



Edit Selection View Go Run Terminal Help

JS App.js X JS Locations.js JS FarmVisit.js

JS App.js > ⚙ App

```

1 import React from 'react';
2 import { Link, Redirect, Route, Switch } from "react-router-dom";
3 import './App.css';
4 import FarmVisit from './components/FarmVisit';
5 import Locations from './components/Locations';
6
7 function App() {
8   return (
9     <div>
10       <nav className="navbar navbar-expand-lg navbar-light bg-warning">
11         <Link className="navbar-brand " to="/">Infy Green</Link>
12       </nav>
13       <Switch>
14         {/* Add the routes as mentioned below:
15           1. It should redirect to /locations page for default URL i.e. http://localhost:3001/
16           2. It should load FarmVisit component for /farmVisit/:locationName
17           3. It should load Locations component for /locations
18           4. It should redirect to /locations page for any invalid URL e.g. http://localhost:3001/somethingelse
19         */
20         <Route path='/' render={()=><Redirect to='/locations'></Redirect>>}/>
21         <Route path='/farmVisit/:locationName' component={FarmVisit}/>
22         <Route path='/locations' component={Locations}/>
23         <Route path='*' render={()=><Redirect to='/locations'></Redirect>>}/>
24       </Switch>
25     </div>
26   );
27 }
28
29

```



```
33     this.getAllLocations();
34 }
35
36     /* To Be Implemented */
37 handleClick = (location) => [
38     /*
39      It should set the state redirectStatus to true
40      and selectedLocationName as the locationName of location Obj
41     */
42     const newState={
43         redirectStatus:true,
44         selectedLocationName:location.locationName
45     }
46     this.setState(newState);
47 ]
48
49 /* To Be Modified - iterate over the locationsArray state to display
50 displayLocations = () => {
51     return (
52         <div className="col-md-3" key="SOME KEY">
53             <div className="card shadow">
```

app.js

JS Locations.js X

JS FarmVisit.js

Components > JS Locations.js > Locations

```

/*
  It should set the state redirectStatus to true
  and selectedLocationName as the locationName of location Object received as parameter
*/
const newState={
  redirectStatus:true,
  selectedLocationName:location.locationName
}
this.setState(newState);

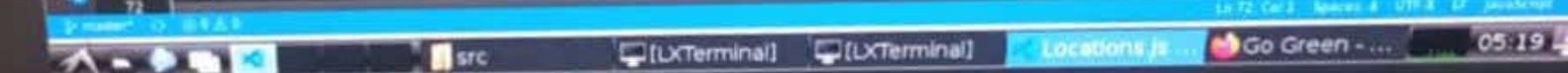
/* To Be Modified - iterate over the locationsArray state to display the location cards */
displayLocations = () => {
  return (this.state.locationsArray.map((farmLocation,index)=>{
    <div className="col-md-3" key={index}>
      <div className="card shadow">
        {/* Add locationImagePath as source and id as the alternate text for the image */}
        <img style={{ height: '200px' }} className="card-img-top" />
        <img style={{height:'200px'}} className="card-img-top" src={farmLocation.locationImage} alt={farmLocation.locationName} />
        <div className="card-body gradient-buttons text-center">
          <h4 className="text" name="locationName">
            {/* Display the locationName of the location here */}
            {farmLocation.locationName}
          </h4>
          {/* Invoke handleClick method by passing proper location Object when Book Visit button is clicked */}
          <button name="bookVisit" className="btn btn-primary" onClick={()=>{this.handleClick(farmLocation)}}>
            Book Visit
          </button>
        </div>
    </div>
  ...
})

```



```
edit Selection View DO FILE FARMVISIT
App.js M Locations.js X json FarmVisit.js
components > Locations.js > Locations
25
26
27 /* Implemented - No Changes Required */
28 componentDidMount() {
29   this.getAllLocations();
30 }
31
32 /* To Be Implemented */
33 handleClick = (location) => {
34   this.setState({ redirectStatus:true })
35   this.setState({ selectedLocationName:location.locationName })
36   /*
37     It should set the state redirectStatus to true
38     and selectedLocationName as the locationName of location Object received as parameter into the function
39   */
40 }
41
42 /* To Be Modified - iterate over the locationsArray state to display the location cards */
43 displayLocations = () => {
44   return (this.state.locationsArray.map((farmLocation) => {
45     return<div className="row">
46       <div className="col-md-3" key={farmLocation.id}>
47         <div className="card shadow">
48           /* Add locationImagePath as source and id as the alternate text for the location Image */
49           <img style={{ height: '200px' }} className="card-img-top" src={farmLocation.locationImagePath} alt={(farmLocation.id)}/>
50           <div className="card-body gradient-buttons text-center">
51             <h4 className="text" name="locationName">
52               /* Display the locationName of the location here */
53               {(farmLocation.locationName)}
54             </h4>
55             /* Invoke handleClick method by passing proper location Object when Book Visit button is clicked */
56             <button name="bookVisit" className="btn btn-primary" onClick={()=>(this.handleClick(farmLocation))}>
57               Book Visit
58             </button>
59           </div>
60         </div>
61       </div>
62     </div>
63   )
64 }
65
66
67
68
69
70
71
72

```



Edit Selection View Go Run Terminal Help

JS App.js

JS Locations.js X

JS FarmVisit.js

components > JS Locations.js > Locations

```
67      }
68  }
69
70
71  /* To Be Modified */
72  render() {
73    let { redirectStatus } = this.state;
74    if (redirectStatus) {
75      /* Redirect to /farmVisit/:locationName by passing proper locationName as the route parameter
76      return<Redirect to={"/farmVisit/"+this.state.selectedLocationName}></Redirect>
77    }
78
79    return (
80      <div className="container-fluid mt-5">
81        {this.state.locationsArray.length?
82          <div className="row">{this.displayLocations()}</div>
83        :((
84          <h2 className="text-light text-center">{this.state.errorMessage}</h2>
85        )
86        /*
87         if locationsArray has location Objects - invoke the displayLocations()
88         else render the errorMessage
89        */
90      )
91    )
92  )
93
94
95
```



```
app.js          JS Locations.js X   JS FarmVisit.js
Components > JS Locations.js > Locations
  <div className="card-body gradient-buttons text-center">
    <h4 className="text" name="locationName">
      /* Display the locationName of the location here */
      {farmLocation.locationName}
    </h4>
    /* Invoke handleClick method by passing proper location Object when Book Visit
    <button name="bookVisit" className="btn btn-primary" onClick={()=>{th
      | Book Visit
    </button>
  </div>
</div>
</div>
}

/* To Be Modified */
render() {
  let { redirectStatus } = this.state;
  if (redirectStatus) {
    /* Redirect to /farmVisit/:locationName by passing proper locationName as the route
    return<Redirect to={"/farmVisit/" + this.state.selectedLocationName}></Redirect>
  }

  return (
    <div className="container-fluid mt-5">
      {this.state.locationsArray.length?(
        <div className="row">{this.displayLocations()}</div>
      ) : (
        <div style={{height: 200, width: 100, margin: "auto", border: "1px solid black", padding: 10}}>
          No Locations Found
        </div>
      )
    </div>
  )
}
```



```
tionName}

    method by passing proper location Object when Book Visit button is clicked */
    t" className="btn btn-primary" onClick={()=>{this.handleClick(farmLocation)}} >

nName by passing proper locationName as the route parameter */
this.state.selectedLocationName}></Redirect>

-5">
gth?(
.displayLocations()}</div>

text-center">{this.state.errorMessage}</h2>
```

location Objects - invoke the displayLocations method

Ln 70, Col 1 Spaces: 4 UTF-8 LF JavaScript

Execute

Project

Locations.js ...

[Go Green - ...]

[LXTerminal]

[LXTerminal]

hp



dson -
en

FarmVisit.js - src - Visual Studio Code

File Edit Selection View Go Run Terminal Help

JS App.js JS Locations.js JS FarmVisit.js X

components > JS FarmVisit.js > FarmVisit > validateField

```
1 import React, { Component } from "react";
2 import axios from "axios";
3
4 const url = "http://localhost:4000/bookings";
5
6 class FarmVisit extends Component {
7   constructor(props) {
8     super(props);
9     this.state = {
10       bookVisitForm: {
11         city: this.props.match.params.locationName,
12         visitDate: "",
13         phoneNumber: "",
14         customerName: ""
15       },
16       bookVisitFormErrorMessage: {
17         city: "",
18         visitDate: "",
19         phoneNumber: "",
20         customerName: ""
21     }
22   }
23 }
```



File Edit Selection View Go Run Terminal Help

JS App.js

JS Locations.js

JS FarmVisit.js X

```
components > JS FarmVisit.js > 🐾 FarmVisit > 🔑 handleChange > 📡 newState > 🔑 bookVisitForm
33
34     /* To Be Implemented */
35     handleChange = (event) => {
36         /*
37             should use the name and value from the event object
38             and set the received value to the corresponding bookVisitForm state
39         */
40         const fieldName=event.target.value;
41         const value=event.target.value;
42         const newState={
43             bookVisitForm:[
44                 ...this.state.bookVisitForm,
45                 [fieldName]:value,
46             ]
47         }
48         this.setState(newState,this.validateField(fieldName,value));
49     }
50
51     /* To Be Modified */
52     validateField = (fieldName, value) => {
```

Ex



File Edit Selection View Go Run Terminal Help

JS App.js

JS Locations.js

JS FarmVisit.js X

```
components > JS FarmVisit.js > 📁 FarmVisit > 🔗 handleChange > 🏷 newState > 🔗 bookVisitForm
46      }
47    }
48    this.setState(newState, this.validateField(fieldName, value));
49  }
50
51  /* To Be Modified */
52  validateField = (fieldName, value) => {
53    let bookVisitFormErrorMessage = this.state.bookVisitFormErrorMessage;
54    let bookVisitFormValid = this.state.bookVisitFormValid;
55    const date='27/07/2022';
56    switch (fieldName) {
57      case "city":
58        if (value === "") {
59          bookVisitFormErrorMessage.city = "Field Required"
60          bookVisitFormValid.city = false
61        }
62        else {
63          bookVisitFormErrorMessage.city = ""
64          bookVisitFormValid.city = true
65        }

```



File Edit Selection View Go Run Terminal Help



JS App.js



FarmVisit.js - src - Visual Studio Code

```
File Edit Selection View Go Run Terminal Help
js App.js js Locations.js db.json FarmVisit.js x booking.js
components > js FarmVisit.js > FarmVisit > validateField
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
```

/* To Be Implemented */
handleChange = (event) => {
 const name = event.target.name;
 const value = event.target.value;
 const newState = {
 bookVisitForm:
 ...this.state.bookVisitForm,[name]:value
 }
 this.setState(newState,this.validateField(name,value))
 /*
 * should use the name and value from the event object
 * and set the received value to the corresponding bookVisitForm state property
 */
}

/* To Be Modified */
validateField = (fieldName, value) => {
 let bookVisitFormErrorMessage = this.state.bookVisitFormErrorMessage;
 let bookVisitFormValid = this.state.bookVisitFormValid;
 const date='27/07/2022'
 switch (fieldName) {
 case "city":
 if (value === "") {
 bookVisitFormErrorMessage.city = "Field Required"
 bookVisitFormValid.city = false
 } else {
 bookVisitFormErrorMessage.city = ""
 bookVisitFormValid.city = true
 }
 break;
 case "customerName":
 if(value ===""){
 bookVisitFormErrorMessage.customerName='Field Required'
 bookVisitFormValid.customerName = false
 }else if(value<0 || value>10){
 bookVisitFormErrorMessage.customerName='Name should contain two words and each word should have atleast 3 characters'
 bookVisitFormValid.customerName = false
 }
 else{
 bookVisitFormErrorMessage.customerName=""
 bookVisitFormValid.customerName = true
 }
 }
}

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

user@ip-39-8-42-76:~/Desktop/user_repo/Project/Frontend/srcs []

I bash



```
app.js      Locations.js      db.json      FarmVisit.js x      booking.js
Components > FarmVisit.js > FarmVisit > validateField
  ...
    bookVisitFormErrorMessage.city = ""
    bookVisitFormValid.city = true
  }
  break;
case "customerName":
  if(value === ""){
    bookVisitFormErrorMessage.customerName='Field Required'
    bookVisitFormValid.customerName = false
  }else if(value<0 || value >10){
    bookVisitFormErrorMessage.customerName='Name should contain two words and each word should have at least 3 characters'
    bookVisitFormValid.customerName = false
  }
  else{
    bookVisitFormErrorMessage.customerName=''
    bookVisitFormValid.customerName = true
  }
}
1. if customerName field is empty it should set the error message as "Field Required"
2. if customerName field does not contain two words with atleast 3 characters each of alphabets
3. for valid input it should reset the error message appropriately
*/
break;
```

File Edit Selection View Go Run Terminal Help



JS App.js



JS Locations.js



JS FarmVisit.js X



```
components > JS FarmVisit.js > 📁 FarmVisit > 🔍 handleChange > 🌐 newState > 🔑 bookVisitForm
85     case "visitDate":
86         /*
87             1. if visitDate field is empty it should set the error message
88             2. if visitDate field is not a future date, it should set the error message
89             3. for valid input it should reset the error message appropriately
90         */
91         if(value===''){
92             bookVisitFormErrorMessage.visitDate="Field Required"
93             bookVisitFormValid.visitDate=false
94         }else{
95             bookVisitFormErrorMessage.visitDate=""
96             bookVisitFormValid.visitDate=true
97         }
98         break;
99     case "phoneNumber":
100        const numberRegex=/^\d{10}$/r
101        /*
102            1. if phoneNumber field is empty it should set the error message
103            2. if phoneNumber field has less than or greater than 10 digits
104            3. for valid input it should reset the error message appropriately
105        */
106    }
```



JS App.js

JS Locations.js

JS FarmVisit.js X

```
components > JS FarmVisit.js > 📁 FarmVisit > ⚡ handleChange > [o] newState > ⚡ bookVisitForm
  98     break;
  99   case "phoneNumber":
100     const numberRegex=/^\d{10}$/;
101     /*
102       1. if phoneNumber field is empty it should set the error message
103       2. if phoneNumber field has less than or greater than 10 digits
104       3. for valid input it should reset the error message appropriate
105     */
106     if(value===""){
107       bookVisitFormErrorMessage.phoneNumber="Field Required"
108       bookVisitFormValid.phoneNumber=false
109     }else if(!value.match(numberRegex)){
110       bookVisitFormErrorMessage.phoneNumber="Phone number should be of 10 digits"
111       bookVisitFormValid.phoneNumber=false
112     }else{
113       bookVisitFormErrorMessage.phoneNumber=""
114       bookVisitFormValid.phoneNumber=true
115     }
116     break;
117   default:
118     break;
```

Ex



```
submitVisit = (event) => {
  event.preventDefault();
  this.successMessage =
  this.errorMessage =
  axios.post("http://localhost:4000/bookings",this.state.bookVisitForm)
  .then(()=>
    this.setState({successMessage:'Your visit is successfully booked for
      ${this.state.bookVisitForm.visitDate}'})
  )
  .catch(()=>
    this.setState({errorMessage:'Something went wrong'})
  )
}

/*
  1. should prevent the page reload on form submission
  2. should reset successMessage and errorMessage state as ""
  3. should make a POST request to http://localhost:4000/bookings with
    bookVisitForm as the request body

```

In 37, Col 5 S



```
is JS Locations.js JS FarmVisit.js X
ents > JS FarmVisit.js > 📁 FarmVisit > 🔑 handleChange > [!] newState > 🛡 bookVisitForm
this.errorMessage=""
axios.post("http://localhost:4000/bookings",this.state.bookVisitForm)
.then(()=>{
  this.setState({successMessage:'Your visit is successfully booked for ${this.state.bookVi
})
})
.catch(()=>{
  this.setState({errorMessage:'Something went wrong'})
})
}

/* To Be Modified */
render() {
  return (
    <div className="row">
      <div className="col-md-6 offset-md-3">
        <br />
        <div className="card">
          <div className="card-header text-center bg-success">
            <h3><label>Farm visit</label></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 bookVisitForm state
              3. All the form fields should have proper name same as bookVisitForm state
              4. city field should be disabled by default
              5. customerName field should be of type text
              6. customerName error message displaying tag should have id as customerName
              7. visitDate field should be of type date
            */}
          </div>
        </div>
      </div>
    </div>
  )
}
```



```
js Locations.js JS FarmVisit.js X
nents > JS FarmVisit.js > 🏠 FarmVisit > 🔑 handleChange > 🌐 newState > 📝 bookVisitForm
  10. phoneNumber error message displaying tag should have id as phoneNumberE
  11. Book Visit button should have id as bookVisitBtn
  12. It should be disabled until buttonActive state property becomes true
  13. display errorMessage and successMessage on form submission appropriately
  14. both errorMessage and successMessage should not appear together
 */
<form>
  <div className="form-group">
    <input type="text" name="city" value={this.state.bookVisitForm.city}/>
  </div>
  <div className="form-group">
    <label htmlFor="customerName">Customer Name</label>
    <input type="text" id="customerName" name="customerName"
      value={this.state.bookVisitForm.customerName} onChange={this.handleChange}>
  </div>
  <div id='customerNameError'>
    {this.state.bookVisitFormErrorMessage.customerName}
  </div>
  <div className="form-group">
    <label htmlFor="visitDate">Visit Date</label>
    <input type="date" name="visitDate" id="visitDate" value={this.state.bookVis
      onChange={this.handleChange}>
  </div>
  <div id="visitDateError">
    {this.state.bookVisitFormErrorMessage.visitDate}
  </div>
  <div className="form-group">
    <label htmlFor="phoneNumber">Phone Number</label>
    <input type="number" name="phoneNumber" id="phoneNumber"
      value={this.state.bookVisitForm.phoneNumber}>
  </div>
  <div id="phoneNumberError" style={{color: red}}>
    {this.state.bookVisitFormErrorMessage.phoneNumber}
  </div>
</form>
```



pp.js JS Locations.js JS FarmVisit.js X

Components > JS FarmVisit.js > handleChange > newState > bookVisitForm

```
<div className="form-group">
  <label htmlFor="visitDate">Visit Date</label>
  <input type="date" name="visitDate" id="visitDate" value={this.state.bookVi
  onChange={this.handleChange}}/>
</div>
<div id="visitDateError">
  {this.state.bookVisitFormErrorMessage.visitDate}
</div>
<div className="form-group">
  <label htmlFor="phoneNumber">Phone Number</label>
  <input type="number" name="phoneNumber" id="phoneNumber"
  value={this.state.bookVisitFormErrorMessage.phoneNumber}></input>
</div>
<div id="phoneNumberError">
  {this.state.bookVisitFormErrorMessage.phoneNumber}
</div>
<div>
  <button type="submit" id="bookVisitBtn"
  disabled={!this.state.bookVisitFormValid.buttonActive}>Book Visit</button>
</div>
{this.state.successMessage?(
  <div id="successMessage">{this.state.successMessage}
  </div>):("")}
{this.state.errorMessage?(
  <div id="errorMessage">{this.state.errorMessage}
  </div>):("")}

</form>
</div>
```

