React Forms

- You need to Design Forms to handle the User Input.
- Forms (User Input) can be Handle in Two Ways
  1. Controlled Components
  2. Uncontrolled Components

1) Controlled Components

- An input form element whose value is controlled by React in this way is called a “controlled component”.
- With a controlled component, the input’s value is always driven by the React state.

2) Uncontrolled Components

- In most cases, It is recommend using controlled components to implement forms.
- In a controlled component, form data is handled by a React component.
- The alternative is uncontrolled components, where form data is handled by the DOM itself.
- To write an uncontrolled component, instead of writing an event handler for every state update, you can use a ref to get form values from the DOM.

Lab21 : Forms Example using Controlled Components

Lab21: Files Required:

1. index.html	2. index.js
3. index.css	4. App.js
5. LoginForm.js	

1. index.html

2. index.js

3. index.css

4. App.js

```
import React, { Component } from 'react';

import 'bootstrap/dist/css/bootstrap.min.css';
import LoginForm from './LoginForm';

class App extends Component {

  render() {

  return (
   <div className="card">
   <div className="card-header">
     <h2 className="text-center">Welcome to Java Learning Center</h2>
   </div>
```

```
    <div className="card-body">
      <LoginForm/>
    </div>
    </div>
   );
  }
}

export default App;
```

5. LoginForm.js

```
import React, { Component } from 'react';

class LoginForm extends Component {

    state = {
      username : '',
      password : ''
    }

    mychangeHandler = (myevent) => {
      myevent.preventDefault();
      console.log("mychangeHandler");
      this.setState({
        [myevent.target.name]: myevent.target.value,
      })
    };

    myclickHandler = (myevent) => {
      myevent.preventDefault();
      console.log("myclickHandler");
      console.log(this.state);
    };

     mysubmitHanlder = (myevent) => {
      myevent.preventDefault();
      console.log("mysubmitHanlder");
      console.log(this.state);
      //Do the Validations
      //Make API Calls
      //Reset the Form
    };
```

```
  render() {
   return (
  <div className="container col-md-6">
 <h2 className="text-center">Login Form</h2>
 <form onSubmit={this.mysubmitHanlder}>
  <table className="table table-striped table-bordered table-light">
    <tbody>
   <tr>
    <td><label htmlFor="username"> Username </label></td>
    <td>
     <input   type="text"
      id="username"
      name="username"
      className="form-control form-control-lg"
      placeholder="Enter Username..."
      value={this.state.username}
      onChange={this.mychangeHandler}  />      </td>
   </tr>
   <tr>
    <td><label htmlFor="password"> Password </label></td>
    <td>
     <input    type="password"
      id="password"
      name="password"
      className="form-control form-control-lg"
      placeholder="Enter Password..."
      value={this.state.password}
      onChange={this.mychangeHandler}  />      </td>
   </tr>
   <tr>
    <td colSpan="2" align="center">
     <button     type="submit"
      onClick={this.myclickHandler}
      className="btn btn-primary btn-lg" >   <h2>Login Now</h2> </button>
    </td>
   </tr>
   </tbody>
  </table>
 </form>
</div>
   );
 }
}

export default LoginForm;
```

Lab22 : Forms Example using UN-Controlled Components

Lab22: Files Required:

1. index.html	2. index.js
3. index.css	4. App.js
5. LoginForm.js	

1. index.html

2. index.js

3. index.css

4. App.js

```
import React, { Component } from 'react';
import 'bootstrap/dist/css/bootstrap.min.css';
import LoginForm from './LoginForm';

class App extends Component {

  render() {
   return (
   <div className="card">
   <div className="card-header">
     <h2 className="text-center">Welcome to Java Learning Center</h2>
   </div>
  <div className="card-body">
     <LoginForm/>
   </div>
   </div>
  );
 }
}

export default App;
```

5. LoginForm.js

```
import React, { Component } from 'react';

class LoginForm extends Component {

 static defaultProps = {
   username: "Srinivas",
   password: "Hello123",
 };
```

```
constructor(props) {
 super(props);

 this.usernameInput = React.createRef(); //300
 this.passwordInput = React.createRef(); //500
}

  myclickHandler = (myevent) => {
   myevent.preventDefault();
   console.log("myclickHandler");
   console.log("UN : ",this.usernameInput.current.value);
   console.log("PW : ",this.passwordInput.current.value);
  };

  mysubmitHanlder = (myevent) => {
   myevent.preventDefault();
   console.log("mysubmitHanlder");
   console.log("UN : ",this.usernameInput.current.value);
   console.log("PW : ",this.passwordInput.current.value);

   //Do the Validations
   //Make API Calls
   //Reset the Form
  };

 render() {
  return (
 <div className="container col-md-6">
<h2 className="text-center">Login Form</h2>
 <form onSubmit={this.mysubmitHanlder}>
 <table className="table table-striped table-bordered table-light">
   <tbody>
  <tr>
   <td><label htmlFor="username"> Username </label></td>
   <td>
    <input   type="text"
     id="username"
     name="username"
     className="form-control form-control-lg"
     placeholder="Enter Username..."
     defaultValue={this.props.username}
     ref={this.usernameInput}   />

   </td>
  </tr>
```

```
   <tr>
    <td><label htmlFor="password"> Password </label></td>
    <td>
     <input   type="text"
      id="password"
      name="password"
      className="form-control form-control-lg"
      placeholder="Enter Password..."
      defaultValue={this.props.password}
      ref={this.passwordInput}   />
     </td>
   </tr>

   <tr>
    <td colSpan="2" align="center">
     <button    type="submit"
      onClick={this.myclickHandler}
      className="btn btn-primary btn-lg"  >
      <h2>Login Now</h2>
     </button>
    </td>
   </tr>
   </tbody>
  </table>
  Welcome to {this.props.username}
 </form>
</div>
   );
 }
}

export default LoginForm;
```

Lab23 : Forms Example using Controlled Components

Lab23: Files Required:

1. index.html	2. index.js
3. index.css	4. App.js
5. AddBook.js	

1. index.html

2. index.css

3. index.js

```
import React from 'react';
import ReactDOM from 'react-dom';

import 'bootstrap/dist/css/bootstrap.min.css';
import './index.css';

import App from './App';

ReactDOM.render(<App />, document.getElementById('myroot'));
```

4. App.js

```
import React, { Component } from 'react';

import 'bootstrap/dist/css/bootstrap.min.css';
import AddBook from './AddBook';

class App extends Component {

  render() {
   return (
  <div className="card">
  <div className="card-header">
    <h2 className="text-center">Welcome to Java Learning Center</h2>
  </div>
  <div className="card-body">
    <AddBook/>
  </div>
  </div>

  );
 }
}

export default App;
```

5. AddBook.js

```
import React, { Component } from 'react';
import classnames from 'classnames';

class AddBook extends Component {

state = {
bookId : '',
bookName : '',
author : '',
price : '',
pub : '',
myerrors : {  }
}

mychangeHandler = (myevent) => {
myevent.preventDefault();
console.log("mychangeHandler");
this.setState({
[myevent.target.name]: myevent.target.value,
})
};

mysubmitHanlder = (myevent) => {
myevent.preventDefault();
console.log("mysubmitHanlder");
//1.Do the Validations
if(this.validateBookForm()){
//2.Make API Calls
console.log("Before Reset",this.state);

//3.Reset the Form
this.setState({
bookId : '',
bookName : '',
author : '',
price : '',
pub : '',
myerrors : { }
});

console.log("After Reset",this.state);
}

};

validateBookForm(){
console.log("validateBookForm");
const {bookId,bookName,author,price,pub} = this.state;

let isValid = true;
let errors= { }; //Errors here temporarly


if(bookId===null || bookId.length===0){
isValid=false;
errors["mybookId"] ="BookId is Required";//
}else if(bookId.length<=4){
isValid=false;
errors["mybookId"] ="BookId minlength is 5";//
}else if(bookId.length>=8){
isValid=false;
errors["mybookId"] ="BookId maxlength is 7";//
}

if(bookName===null || bookName.length===0){
isValid=false;
errors["mybookName"] ="BookName is Required";//
}else if(bookName.length<=5){
isValid=false;
errors["mybookName"] ="BookName minlength is 5";//
}else if(bookName.length>=11){
isValid=false;
errors["mybookName"] ="BookName maxlength is 7";//
}
```

```
if(author===null || author.length===0){
isValid=false;
errors["myauthor"] ="Author is Required";//
}else if(author.length<=5){
isValid=false;
errors["myauthor"] ="Author minlength is 5";//
}else if(author.length>=11){
isValid=false;
errors["myauthor"] ="Author maxlength is 7";//
}

if(price===null || price.length===0){
isValid=false;
errors["myprice"] ="Price is Required";//
}

if(pub===null || pub.length===0){
isValid=false;
errors["mypub"] ="Pub is Required";//
}

this.setState({
myerrors : errors
});

return isValid;

} //end of Function

render() {

const {bookId,bookName,author,price,pub} = this.state;

return (
<div className="container col-md-6">
<h2 className="text-center">AddBook Form</h2>
<form onSubmit={this.mysubmitHanlder}>
```

```
<div className="form-group">
<label htmlFor="bookId" > Book Id </label>
        <input type="text"
                name="bookId"
                className={classnames("form-control form-control-lg",
                        {"is-invalid":this.state.myerrors.mybookId,"
                        is-valid":!this.state.myerrors.mybookId})}
                placeholder="Enter BookId..."
                value={bookId}
                onChange={this.mychangeHandler}/>
<div className="text-danger"> {this.state.myerrors.mybookId} </div>
</div>

<div className="form-group">
<label htmlFor="bookName" > Book Name </label>
        <input type="text"
                name="bookName"
                className={classnames("form-control form-control-lg",
                                {"is-invalid":this.state.myerrors.mybookName,"
                                is-valid":!this.state.myerrors.mybookName})}
                placeholder="Enter Book Name..."
                value={bookName}
                onChange={this.mychangeHandler}/>
<div className="text-danger"> {this.state.myerrors.mybookName} </div>
</div>

<div className="form-group">
<label htmlFor="author" > Author </label>
        <input type="text"
                name="author"
                className={classnames("form-control form-control-lg",
                        {"is-invalid":this.state.myerrors.myauthor,
                        "is-valid":!this.state.myerrors.myauthor})}
                placeholder="Enter Author..."
                value={author}
                onChange={this.mychangeHandler}/>
<div className="text-danger"> {this.state.myerrors.myauthor} </div>
</div>
```

```
<div className="form-group">
<label htmlFor="price" > Price </label>
        <input type="text"
                name="price"
                className={classnames("form-control form-control-lg",
                        {"is-invalid":this.state.myerrors.myprice,
                        "is-valid":!this.state.myerrors.myprice})}
                placeholder="Enter Price..."
                value={price}
                onChange={this.mychangeHandler}/>
<div className="text-danger"> {this.state.myerrors.myprice} </div>
</div>

<div className="form-group">
<label htmlFor="pub" > Publiccations </label>
        <input type="text"
                name="pub"
                className={classnames("form-control form-control-lg",
                        {"is-invalid":this.state.myerrors.mypub,
                        "is-valid":!this.state.myerrors.mypub})}
                placeholder="Enter Pub..."
                value={pub}
                onChange={this.mychangeHandler}/>
<div className="text-danger"> {this.state.myerrors.mypub} </div>
</div>

<input type="submit"  value="Add Book"   className="btn btn-primary btn-lg"/>
</form>
</div>
);
}
}

export default AddBook;
```

Lab24 : Forms Example using Controlled Components

Lab24: Files Required:

1. index.html	2. index.js
3. index.css	4. App.js
5. AddBook.js	6. MyTextInput1.js
7. MyTextInput.js	

1. index.html

2. index.css

3. index.js

```
import React from 'react';
import ReactDOM from 'react-dom';

import 'bootstrap/dist/css/bootstrap.min.css';
import './index.css';

import App from './App';

ReactDOM.render(<App />, document.getElementById('myroot'));
```

4. App.js

```
import React, { Component } from 'react';

import 'bootstrap/dist/css/bootstrap.min.css';
import AddBook from './books/AddBook';

class App extends Component {

  render() {
   return (
  <div className="card">
  <div className="card-header">
    <h2 className="text-center">Welcome to Java Learning Center</h2>
  </div>
  <div className="card-body">
     <AddBook/>
  </div>
  </div>

  );
 }
}

export default App;
```

5. AddBook.js

```
import React, { Component } from 'react';

import MyTextInput from '../mycomponents/MyTextInput';
import MyTextInput1 from '../mycomponents/MyTextInput1';

class AddBook extends Component {
   state = {
     bookId : '',
     bookName : '',
     author : '',
     price : '',
     pub : '',
     myerrors : {   }
  }

  mychangeHandler = (myevent) => {
    myevent.preventDefault();
    console.log("mychangeHandler");
    this.setState({
      [myevent.target.name]: myevent.target.value,
    })
   };

   mysubmitHanlder = (myevent) => {
    myevent.preventDefault();
    console.log("mysubmitHanlder");
   //1.Do the Validations
   if(this.validateBookForm()){
    //2.Make API Calls
    console.log("Before Reset",this.state);

    //3.Reset the Form
     this.setState({
      bookId : '',
      bookName : '',
      author : '',
      price : '',
      pub : '',
      myerrors : { }
   });

   console.log("After Reset",this.state);
  }
 };

   validateBookForm(){
      console.log("validateBookForm");
      const {bookId,bookName,author,price,pub} = this.state;

      let isValid = true;
      let errors= { }; //Errors here temporarly


 if(bookId===null || bookId.length===0){
  isValid=false;
  errors["mybookId"] ="BookId is Required";//
 }else if(bookId.length<=4){
  isValid=false;
  errors["mybookId"] ="BookId minlength is 5";//
 }else if(bookId.length>=8){
  isValid=false;
  errors["mybookId"] ="BookId maxlength is 7";//
  }

 if(bookName===null || bookName.length===0){
  isValid=false;
  errors["mybookName"] ="BookName is Required";//
 }else if(bookName.length<=5){
  isValid=false;
  errors["mybookName"] ="BookName minlength is 5";//
 }else if(bookName.length>=11){
  isValid=false;
  errors["mybookName"] ="BookName maxlength is 7";//
 }
```

```
if(author===null || author.length===0){
  isValid=false;
  errors["myauthor"] ="Author is Required";//
}else if(author.length<=5){
  isValid=false;
  errors["myauthor"] ="Author minlength is 5";//
}else if(author.length>=11){
  isValid=false;
  errors["myauthor"] ="Author maxlength is 7";//
}

if(price===null || price.length===0){
  isValid=false;
  errors["myprice"] ="Price is Required";//
}

if(pub===null || pub.length===0){
  isValid=false;
  errors["mypub"] ="Pub is Required";//
}

this.setState({
  myerrors : errors
});

return isValid;

} //end of Function

 render() {

 const {bookId,bookName,author,price,pub} = this.state;
 return (
  <div className="container col-md-6">
<h2 className="text-center">AddBook Form</h2>

<form onSubmit={this.mysubmitHanlder}>
```

```
<MyTextInput
    myname="bookId"
    mylabel="Book Id"
    myvalue={bookId}
    myonChange={this.mychangeHandler}
    myerror={this.state.myerrors.mybookId} />

 <MyTextInput
    mytype="password"
    myname="bookName"
    mylabel="Book Name"
    myplacehodler="Enter Book Name"
    myvalue={bookName}
    myonChange={this.mychangeHandler}
    myerror={this.state.myerrors.mybookName} />

 <MyTextInput
    mytype="text"
    myname="author"
    mylabel="Author"
    myplacehodler="Enter Author"
    myvalue={author}
    myonChange={this.mychangeHandler}
    myerror={this.state.myerrors.myauthor} />

 <MyTextInput1
    mytype="text"
    myname="price"
    mylabel="Price"
    myplacehodler="Enter Price"
    myvalue={price}
    myonChange={this.mychangeHandler}
    myerror={this.state.myerrors.myprice} />

 <MyTextInput1
    mytype="text"
    myname="pub"
    mylabel="Publications"
```

```
        myplacehodler="Enter Pub"
        myvalue={pub}
        myonChange={this.mychangeHandler}
        myerror={this.state.myerrors.mypub} />

        <input type="submit"   value="Add Book"   className="btn btn-primary btn-lg"/>
        </form>
       </div>
  );
 }
}
export default AddBook;
```

6. MyTextInput1.js

```
import React from 'react';

import classnames from 'classnames';
import propTypes from 'prop-types';

const MyTextInput1 = (
   {myname,mylabel,mytype,myvalue,myplaceholder,myonChange,myerror}
   ) => {
return(
   <div className="container">
     <div className="form-group">
       <label htmlFor= {myname} > {mylabel} </label>
       <input type= {mytype}
            name= {myname}
            placeholder= {myplaceholder}
            value= {myvalue}
            onChange= {myonChange}
            className={  classnames("form-control form-control-lg",
              {"is-invalid":myerror, "is-valid":!myerror} )    }
              />
       <div className="text-danger"> {myerror} </div>
       </div>
   </div>
)  }
```

```
MyTextInput1.propTypes = {
  mylabel : propTypes.string.isRequired,
  myname : propTypes.string.isRequired,
  myvalue : propTypes.string.isRequired,
  myonChange : propTypes.func.isRequired,
}

MyTextInput1.defaultProps = {
  mytext:'text',
  myerror:'',
}
export default MyTextInput1;
```

7. MyTextInput.js

```
import React from 'react';

import classnames from 'classnames';
import propTypes from 'prop-types';

const MyTextInput = (props) => {
return(
  <div className="container">
    <div className="form-group">
      <label htmlFor= {props.myname} > {props.mylabel} </label>
      <input type= {props.mytype}
          name= {props.myname}
          placeholder= {props.myplaceholder}
          value= {props.myvalue}
          onChange= {props.myonChange}
          className={ classnames("form-control form-control-lg",
            {"is-invalid":props.myerror,  "is-valid":!props.myerror} )  }
         />
      <div className="text-danger"> {props.myerror} </div>
      </div>
  </div>
)
}
```

```
MyTextInput.propTypes = {
   mylabel : propTypes.string.isRequired,
   myname : propTypes.string.isRequired,
   myvalue : propTypes.string.isRequired,
   myonChange : propTypes.func.isRequired,
}

MyTextInput.defaultProps = {
   mytext:'text',
   myerror:'',
}

export default MyTextInput;
```

Lab25 : Customizing and Using Form Elements

Lab25: Files Required:

1. index.html	2. index.js
3. index.css	4. App.js
5. MyButton.js	
6. MyTextInput.js	7. TextInputDemo.js
8. MyTextArea.js	9. TextAreaDemo.js
10. MySelectBox.js	11. SelectBoxDemo.js
12. MyRadioButton.js	13. RadioButtonDemo.js
14. MyCheckBox.js	15. CheckBoxDemo.js

1. index.html

2. index.js

3. index.css

4. App.js

```
import React, { Component } from 'react';

import 'bootstrap/dist/css/bootstrap.min.css';
import TextInputDemo from './mydemos/TextInputDemo';
import TextAreaDemo from './mydemos/TextAreaDemo';
import SelectBoxDemo from './mydemos/SelectBoxDemo';
import RadioButtonDemo from './mydemos/RadioButtonDemo';
import CheckBoxDemo from './mydemos/CheckBoxDemo';

class App extends Component {

  render() {

  return (
   <div className="card">
   <div className="card-header">
     <h2 className="text-center">Welcome to Java Learning Center</h2>
   </div>
   <div className="card-body">
     <TextInputDemo />
     <TextAreaDemo />
     <SelectBoxDemo />
     <RadioButtonDemo />
     <CheckBoxDemo/>
   </div>
   </div>
  );
 }
}

export default App;
```

5. MyButton.js

```
import React from 'react';

import propTypes from 'prop-types';

const MyButton = (props) => {
return(
   <div className="container">
     <div className="form-group">
       <input type={props.mytype}
           value={props.myvalue}
           className="btn btn-primary btn-lg"
           />
       </div>
   </div>
)
}

MyButton.propTypes = {
   myvalue : propTypes.string.isRequired,
}

MyButton.defaultProps = {
   mytype:'submit',
}

export default MyButton;
```

6. MyTextInput.js

```
import React from 'react';
import propTypes from 'prop-types';

const MyTextInput = (props) => {
return(
   <div className="container">
     <div className="form-group">
       <label htmlFor={props.myname} > {props.mylabel} </label>
       <input type={props.mytype}
           name={props.myname}
           placeholder={props.myplaceholder}
           value={props.myvalue}
           onChange={props.myOnChange}
           className="form-control form-control-lg"
           />
       </div>
   </div>
)
}

MyTextInput.propTypes = {
  mylabel : propTypes.string.isRequired,
  myname : propTypes.string.isRequired,
  myvalue : propTypes.string.isRequired,
  myOnChange : propTypes.func.isRequired,
}

MyTextInput.defaultProps = {
  mytype:'text',
}

export default MyTextInput;
```

7. TextInputDemo.js

```
import React, { Component } from "react";
import MyButton from "../mycomponents/MyButton";
import MyTextInput from "../mycomponents/MyTextInput";

class TextInputDemo extends Component {
 state = {
  username: "sd",
  password: "sd"
 };

 onChangeHandler = (event) => {
  console.log("onChangeHandler");
  this.setState({
   [event.target.name]: event.target.value,
  });
 };

 onSubmitHandler = (event) => {
  event.preventDefault();
  console.log("onSubmitHandler");
  const { username,password } = this.state;
  console.log(username);
  console.log(password);
  //Do the Validations
  //Make Call to Server
  //Form Reset
  this.setState({
    username: "",
    password: ""
  });
 };

 render() {
  const { username,password } = this.state;

  return (
   <div className="card-body container col-md-6">
    <h3 className="text-center"> InputTextDemo </h3>
    <form onSubmit={this.onSubmitHandler}>

     <MyTextInput
      myname="username"
      mylabel="Username"
      myvalue={username}
      myOnChange={this.onChangeHandler}/>

     <MyTextInput
      mytype="password"
      myname="password"
      mylabel="Password"
      myvalue={password}
      myOnChange={this.onChangeHandler}/>

    <MyButton myvalue="Submit Now"/>
    </form>
   </div>
  );
 }
}
export default TextInputDemo;
```

8. MyTextArea.js

```
import React from "react";
import propTypes from "prop-types";

const MyTextArea = (props) => {
  return (
    <div className="form-group">
      <h3>    <label> {props.mylabel} </label>   </h3>
      <textarea
        value={props.myvalue}
        onChange={props.myOnChange}
        placeholder={props.myplaceholder}
        rows={props.myrows}
        cols={props.mycols}
        className="form-control form-control-lg"
      />
    </div>
  );
};
MyTextArea.propTypes = {
  mylabel: propTypes.string.isRequired,
  myvalue: propTypes.string.isRequired,
  myOnChange: propTypes.func.isRequired,
};
MyTextArea.defaultProps = {
  myplaceholder: "",
  myrows: 5,
  mycols:25
};
export default MyTextArea;
```

9. TextAreaDemo.js

```
import React, { Component } from "react";
import MyButton from "../mycomponents/MyButton";
import MyTextArea from "../mycomponents/MyTextArea";

class TextAreaDemo extends Component {
  constructor(props) {
    super(props);
    this.state = {
      remarks: "",
    };
    this.onChangeHandler = this.onChangeHandler.bind(this);
    this.onSubmitHandler = this.onSubmitHandler.bind(this);
  }

  onChangeHandler = (event) => {
    console.log("onChangeHandler");
    this.setState({
      remarks: event.target.value,
    });
  };

  onSubmitHandler = (event) => {
    event.preventDefault();
    console.log("onSubmitHandler");
    console.info(this.state.remarks);
    //Do the Validations
    //Make Call to Server
    //Form Reset
    this.setState({
      remarks: "",
    });
  };
  render() {
    const { remarks } = this.state;
    return (
      <div className="card-body container col-md-6">
        <h3 className="text-center"> TextAreaDemo </h3>
        <form onSubmit={this.onSubmitHandler}>
          <MyTextArea
            mylabel="Remarks"
            myvalue={remarks}
            myOnChange={this.onChangeHandler}/>

        <MyButton myvalue="Submit Now"/>
        </form>
      </div>
    );
  }    }

export default TextAreaDemo;
```

10. MySelectBox.js

```
import React from "react";
import propTypes from "prop-types";

const MySelectBox = (props) => {
 return (
  <div className="form-group">
   <h3>   <label> {props.mylabel} </label>
    <select  className="form-control form-control-lg"
           name={props.myname}
          onChange={props.myOnChange}    >
     {props.myoptions.map((myoption, myindex) => {
      return (
       <option key={myindex} value={myoption}>
        {myoption}
       </option>
      );
     })}
    </select>
   </h3>
  </div>
 );
};
MySelectBox.propTypes = {
 mylabel: propTypes.string.isRequired,
 myname: propTypes.string.isRequired,
 myoptions: propTypes.array.isRequired,
 myOnChange: propTypes.func.isRequired,
};

export default MySelectBox;
```

11. SelectBoxDemo.js

```
import React from "react";

import MyButton from "../mycomponents/MyButton";
import MySelectBox from "../mycomponents/MySelectBox";

class SelectBoxDemo extends React.Component {
 constructor() {
  super();
```

```
    this.state = {
      timings: null,
    };
    this.onChangeHandler = this.onChangeHandler.bind(this);
    this.onSubmitHandler = this.onSubmitHandler.bind(this);
  }

  onChangeHandler(event) {
    console.log("onChangeHandler");
    this.setState({
      timings: event.target.value,
    });
  }

  onSubmitHandler = (event) => {
    event.preventDefault();
    console.log("onSubmitHandler");
    console.info(this.state.timings);
    //Do the Validations
    //Make Call to Server

    //Form Reset
    this.setState({
      timings: null,
    });
  };

  render() {
    const tmingsOptions = ["Select Timings", "WeekDays", "WeekEnds"];
    return (
      <div className="card-body container col-md-6">
        <h3 className="text-center"> SelectBoxDemo </h3>
        <form onSubmit={this.onSubmitHandler}>

          <MySelectBox
            mylabel="Tmings"
            myname="timings"
            myOnChange={this.onChangeHandler}
            myoptions={tmingsOptions}/>

            <MyButton myvalue="Submit Now"/>
        </form>
      </div>
    );
  }
}

export default SelectBoxDemo;
```

12. MyRadioButton.js

```
import React from "react";
import propTypes from "prop-types";

const MyRadioButton = (props) => {
  return (
    <span>
      <h3>
        <input
          type={props.mytype}
          name={props.myname}
          checked={props.mychecked}
          value={props.myvalue}
          onChange={props.myOnChange}
        />
        &nbsp; &nbsp;
        <label> {props.mylabel} </label>
      </h3>
    </span>
  );
};

MyRadioButton.propTypes = {
  mylabel: propTypes.string.isRequired,
  myname: propTypes.string.isRequired,
  myvalue: propTypes.string.isRequired,
  mychecked: propTypes.bool.isRequired,
  myOnChange: propTypes.func.isRequired,
};
MyRadioButton.defaultProps = {
  mytype: "radio",
};
export default MyRadioButton;
```

13. RadioButtonDemo.js

```
import React, { Component } from "react";
import MyButton from "../mycomponents/MyButton";

import MyRadioButton from "../mycomponents/MyRadioButton";

class RadioButtonDemo extends Component {

  state = {
   gender: "",
  };

 onChangeHandler = (event) => {
  console.log("onChangeHandler");

  this.setState({
   gender: event.target.value,
  });
 };

 onSubmitHandler = (event) => {
  event.preventDefault();
  console.log("onSubmitHandler");
  console.log(this.state.gender);
 };

 render() {
  return (
   <div className="card-body container col-md-6">
    <h3 className="text-center"> RadioButton Demo </h3>
    <form onSubmit={this.onSubmitHandler}>
     <div className="radio">
      <MyRadioButton
       mylabel="Male"
       myvalue="Male"
       myname="gender"
       mychecked={this.state.gender === "Male"}
       myOnChange={this.onChangeHandler}
      />
```

```
         <MyRadioButton
          mylabel="Female"
          myvalue="Female"
          myname="gender"
          mychecked={this.state.gender === "Female"}
          myOnChange={this.onChangeHandler}     />
        </div>

        <MyButton myvalue="Submit Now"/>
      </form>
     </div>
   );
  }
}
export default RadioButtonDemo;
```

14. MyCheckBox.js

```
import React from "react";
import propTypes from "prop-types";

const MyCheckBox = (props) => {
  return (
   <span className="form-check">
    <h3>
     <input
      type={props.mytype}
      value={props.myvalue}
      onChange={props.myOnChange}
      checked={props.mychecked}
      className="form-check-input"/>
     &nbsp; &nbsp;
     <label> {props.mylabel} </label>
    </h3>
   </span>
  );
};
MyCheckBox.propTypes = {
  mylabel: propTypes.string.isRequired,
  myvalue: propTypes.string.isRequired,
  mychecked: propTypes.bool.isRequired,
  myOnChange: propTypes.func.isRequired,
};
MyCheckBox.defaultProps = {
  mytype: "checkbox",
};
export default MyCheckBox;
```

15. CheckBoxDemo.js

```
import React from "react";
import MyButton from "../mycomponents/MyButton";
import MyCheckBox from "../mycomponents/MyCheckBox";

class CheckBoxDemo extends React.Component {
    state = {
    mycourses: [
     { id: 1, value: "React JS", isChecked: true },
     { id: 2, value: "Angular ", isChecked: false },
     { id: 3, value: "Spring Boot", isChecked: true },
     { id: 4, value: "MicroServices", isChecked: false },
     { id: 5, value: "DevOps", isChecked: false },
    ],
   };

 onChangeHandler = (event) => {
   console.log("onChangeHandler");

   let mycourses = this.state.mycourses;
   mycourses.forEach((mycourse) => {
    if (mycourse.value === event.target.value)
      mycourse.isChecked = event.target.checked;
   });
   this.setState({ mycourses: mycourses });
 };

 onSubmitHandler = (event) => {
   event.preventDefault();
   console.log("onSubmitHandler");
   console.log(this.state.mycourses);
   let courseList = [];
   this.state.mycourses.forEach((mycourse) => {
    if (mycourse.isChecked === true) {
      courseList.push(mycourse.value);
    }
   });
   console.log("Selected Courses : ", courseList);

   //reset
   this.setState({
    mycourses: [
      { id: 1, value: "React JS", isChecked: false },
      { id: 2, value: "Angular ", isChecked: false },
      { id: 3, value: "Spring Boot", isChecked: false },
      { id: 4, value: "MicroServices", isChecked: false },
      { id: 5, value: "Java FSD", isChecked: false },
    ],
   });
 };
 render() {
   return (
    <div className="card-body container col-md-6">
      <h3 className="text-center"> CheckBoxDemo </h3>
      <form onSubmit={this.onSubmitHandler}>
        <div className="checkbox">
          {this.state.mycourses.map((mycourse) => {
            return (
              <MyCheckBox
                key={mycourse.id}
                mylabel={mycourse.value}
                myvalue={mycourse.value}
                mychecked={mycourse.isChecked}
                myOnChange={this.onChangeHandler}/>
            );
          })}
        </div>
          <MyButton myvalue="Submit Now"/>
        </form>
    </div>
   );
 }
}

export default CheckBoxDemo;
```