



## BookStore Application – React UI

JLCBookStore

[Home](#) [Login](#) [Register](#)

Books @ Java Learning Center

[Add New Book](#)

Book Id	Book Name	Author	Price	Category	Publications	Edit /Delete	
106	Master Spring	Srinivas	Rs.12500	Java	JLC	<a href="#">Edit Book</a>	<a href="#">Delete Book</a>
107	Master Hibernate	Srinivas	Rs.12500	Web	JLC	<a href="#">Edit Book</a>	<a href="#">Delete Book</a>
108	Master Docker	Srinivas	Rs.12500	Web	JLC	<a href="#">Edit Book</a>	<a href="#">Delete Book</a>
109	Master Kubernetes	Srinivas	Rs.12500	Java	JLC	<a href="#">Edit Book</a>	<a href="#">Delete Book</a>
201	Learn Java Script	Srinivas	Rs.5000	Web	JLC	<a href="#">Edit Book</a>	<a href="#">Delete Book</a>

[Previous Records](#)

[Next Records](#)

JLCBookStore

[Home](#) [Login](#) [Register](#)

### Add Book Form

Book Name	<input type="text"/>
	Provide Book Name
Author	<input type="text"/>
	Provide Author
Price	<input type="text"/>
	Provide Price
Category	<input type="text"/>
	Provide Category
Publications	<input type="text"/>
	Provide Publications

[Add New Book](#)



### Edit Book Form

Book ID	101
Book Name	<input type="text" value="Master Java8"/>
Author	<input type="text" value="Srinivas D"/>
Price	<input type="text" value="12500"/>
Category	<input type="text" value="Java"/>
Publications	<input type="text" value="JLCINDIA"/>

JLCBookStore [Home](#) [Login](#) [Register](#)

Books @ Java Learning Center

Book Id	Book Name	Author	Price	Category	Publications	Edit /Delete
101	Master Java8	Srinivas D	Rs.12500	Java	JLCINDIA	<input type="button" value="Edit Book"/> <input type="button" value="Delete Book"/>
102	Master Angular9	Srinivas	Rs.12500	Web	JLC	<input type="button" value="Edit Book"/> <input type="button" value="Delete Book"/>
103	Master React16	Srinivas	Rs.12500	Web	JLC	<input type="button" value="Edit Book"/> <input type="button" value="Delete Book"/>
104	Master Boot2	Srinivas	Rs.12500	Java	JLC	<input type="button" value="Edit Book"/> <input type="button" value="Delete Book"/>
105	Master MicroServices	Srinivas	Rs.12500	Java	JLC	<input type="button" value="Edit Book"/> <input type="button" value="Delete Book"/>

Delete Confirmation

Are you sure to delete Book : 101

## **Developing Bookstore Application:**

### **Working Steps :**

#### **A)Setup the Base Project**

##### **1. Create the React Project**

```
create-react-app bsapp --scripts-version 1.1.5
```

##### **2. Enable Bootstrap in React App.**

a)Install JQuery

```
npm install jquery --save
```

b)Install bootstrap

```
npm install bootstrap --save
```

c)import the bootstrap.min.css in App.js

```
import "bootstrap/dist/css/bootstrap.min.css";
```

##### **3. Update Index.js**

```
import React from 'react';
```

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

```
import './index.css';
```

```
import App from './App';
```

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

##### **4. Update App.js**

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

```
import "bootstrap/dist/css/bootstrap.min.css";
```

```
class App extends Component {
```

```
  render() {
```

```
    return (
```

```
      <div className="App">
```

```
        <h1 className="text-center"> JLC BookStore </h1>
```

```
      </div>
```

```
    );
```

```
  }
```

```
}
```

```
export default App;
```

## 5. Run the App

npm start

## 6. Install Packages required for Your Project.

```
npm install react-router-dom --save  
npm install axios --save  
npm install classes --save
```

## 7. Add the Required CSS classes in Index.css

As given in BookLet

## 8. Write the Header component.

JLCHeader.js

As given in BookLet

## 9. Write the Footer component.

JLCFooter.js

As given in BookLet

## 10. Write the Body component.

JLCBody.js

As given in BookLet

## 11. Update App.js

As given in BookLet

## 12. Run the App

npm start

## **B) Requirement: Display List of Books**

### **13. Update JLCBody.js**

```
import BooksList from "./BooksList";

<div>
<Route path="/" component={BooksList} exact/>

</div>
```

### **14. Develop the BooksList Component**

BooksList.js

As given in BookLet

## **C) Requirement: Add Book**

### **15. Add the add-book route in JLCBody.js**

```
<div>

<Route path="/" component={BooksList} exact/>
<Route path="/add-book/:bid" component={AddBook} exact/>

</div>
```

### **16. Develop the AddBook Component**

AddBook.js

As given in BookLet

## **D) Requirement: Edit Book**

### **17. Add the edit-book route in JLCBody.js**

```
<div>
<Route path="/" component={BooksList} exact/>
<Route path="/add-book/:bid" component={AddBook} exact/>
<Route path="/edit-book/:mybookId" component={EditBook} exact/>
</div>
```

### **18. Develop the AddBook Component**

EditBook.js

As given in BookLet

## **BookStore ApplicationV.10**

### **Files Required**

1. index.html	2. index.js
3. index.css	4. App.js
5. JLCHeader.js	6. JLCFooter.js
7. JLCBody.js	8. MyTextInput.js
9. BookList.js	10. AddBook.js
11. EditBook.js	

#### **1. index.html**

#### **2. index.js**

```
import React from 'react';
import ReactDOM from 'react-dom';
import './index.css';
import App from './App';
```

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

#### **3. index.css**

```
body {
  margin: 0;
  padding: 0;
  font-family: sans-serif;
}
.mymargin1{
  margin-right: 10px;
}
.mytext-large {
  font-family: Cambria, Cochin, Georgia, Times, "Times New Roman", serif;
  font-size: 25px;
}
.mytext-extra-large {
  font-size: 28px;
}
.mytext-center {
  text-align: center;
}
.mywhite {
  color: white;
}
.myfloat-right{
  float: right;
}
```



```
.myfloat-left{  
  float: left;  
}
```

#### 4. App.js

```
import React, { Component } from 'react';  
  
import "bootstrap/dist/css/bootstrap.min.css";  
import JLCHeader from './JLCHeader';  
import JLCBody from './JLCBody';  
import JLCFooter from './JLCFooter';  
import { BrowserRouter as MyRouter } from 'react-router-dom';  
  
class App extends Component {  
  render() {  
    return (  
      <MyRouter>  
        <div className="card">  
          <JLCHeader/>  
          <JLCBody/>  
          <JLCFooter/>  
        </div>  
      </MyRouter>  
    );  
  }  
}  
  
export default App;
```

#### 5. JLCHeader.js

```
import React from "react";  
import { NavLink } from "react-router-dom";  
  
const JLCHeader = () => {  
  return (  
    <div>  
      <header>  
        <nav className="navbar navbar-expand-lg navbar-light bg-dark">  
          <h3 className="text-center mywhite">JLCBookStore</h3>  
  
          <button className="navbar-toggler" type="button" data-toggle="collapse"  
            data-target="#navbarNav" aria-controls="navbarNav" aria-expanded="false"  
            aria-label="Toggle navigation">  
            <span className="navbar-toggler-icon"></span>  
          </button>  
        </nav>  
      </header>  
    </div>  
  );  
}
```



```
<div className="collapse navbar-collapse justify-content-end" id="navbarNav">
<ul className="nav justify-content-end">
<li className="nav-item active">
<h4>
<NavLink to="/" className="nav-link mywhite" exact> Home </NavLink>
</h4>
</li>
<li className="nav-item active">
<h4>
<NavLink to="/login" className="nav-link mywhite" exact> Login </NavLink>
</h4>
</li>
<li className="nav-item active">
<h4>
<NavLink to="/register" className="nav-link mywhite" exact> Register </NavLink>
</h4>
</li>
</ul>
</div>
</nav>
</header>
</div>
);
};
export default JLCHeader;
```

## 6. JLCFooter.js

```
import React from "react";

const JLCFooter = () => {
  return (
    <div>
      <br /> <br /> <br />
      <footer className="bg-dark mytext-center mytext-large">
        <span className="mytext-large mywhite">&copy; 2020 JLCBookStore</span>{" "}
        <br />
      </footer>
    </div>
  );
};
export default JLCFooter;
```





## 7. JLCBody.js

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

import { Route } from "react-router-dom";
import AddBook from "./AddBook";
import BooksList from "./BooksList";
import EditBook from "./EditBook";

class JLCBody extends Component {
  render() {
    return (
      <div>
        <Route path="/" component={BooksList} exact/>
        <Route path="/add-book" component={AddBook} exact/>
        <Route path="/edit-book/:mybookId" component={EditBook} exact/>
      </div>
    );
  }
}

export default JLCBody;
```

## 8. MyTextInput.js

```
import React from 'react'
import propTypes from 'prop-types';
import classNames from 'classnames';
const MyTextInput =
({mylabel,mytype,myname,myvalue,myplaceholder,myOnChange,myerror})=> {

  return (
    <div className="form-group">
      <label htmlFor={myname} > {mylabel} </label>
      <input type={mytype}
        name={myname}
        className={classNames("form-control form-control-lg",{"is-invalid":myerror})}
        placeholder={myplaceholder}
        value={myvalue}
        onChange={myOnChange}  />
    </div>
  );
}
```



```
{myerror && <div className="invalid-feedback"> {myerror}</div> }
</div>
)
}
MyTextInput.propTypes = {
  mylabel : propTypes.string.isRequired,
  myname : propTypes.string.isRequired,
  myvalue : propTypes.string.isRequired,
  myOnChange : propTypes.func.isRequired,
}
MyTextInput.defaultProps = {
  mytype:"text",
  myplaceholder:"",
  myerror:""
}
export default MyTextInput;
```

## 9. **BookList.js**

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

import axios from "axios";
import { NavLink } from "react-router-dom";
import { Modal } from "react-bootstrap";
import Container from "react-bootstrap/Container";
import Button from "react-bootstrap/Button";

class BookList extends Component {
  state = {
    mybooks: [],
    showConfirm: false,
    mybookIdToDelete: "",
  };

  componentDidMount() {
    console.log("componentDidMount");
    this.getAllBooks();
  }
}
```



```
getAllBooks = () => {
  const URL = "https://coursecube.com/jlc-bookstore/mybooks";
  console.log(URL);
  axios.get(URL).then((myresponse) => {
    console.log(myresponse.data);
    // console.log(myresponse.headers);
    this.setState({
      mybooks: myresponse.data,
    });
  });
};

showDeleteConfirm = (mybookId) => {
  console.log("showDeleteConfirm", mybookId);
  this.setState({
    showConfirm: true,
    mybookIdToDelete: mybookId,
  });
};

hideDeleteConfirm = () => {
  console.log("hideDeleteConfirm");
  this.setState({
    showConfirm: false,
  });
};

handleDelete = () => {
  console.log("handleDelete", this.state.mybookIdToDelete);

  const URL = `https://coursecube.com/jlc-bookstore/deleteBook/${this.state.mybookIdToDelete}`;
  axios.delete(URL)
    .then((myresponse) => {
      console.log(1, myresponse.data);
      this.hideDeleteConfirm();
      this.getAllBooks();
      this.props.history.push("/");
    })
    .catch((myerror) => {
      console.log(2, myerror);
    });
};
```



```
render() {
  const mybooksList = this.state.mybooks.map((mybook) => {
    return (
      <tr key={mybook.bookId}>
        <td> {mybook.bookId} </td>
        <td> {mybook.bookName} </td>
        <td> {mybook.author} </td>
        <td> {mybook.price} </td>
        <td> {mybook.category} </td>
        <td> {mybook.publications} </td>

        <td>
          <NavLink to={"/edit-book/" + mybook.bookId}
            className="btn btn-primary mytext-large mymargin1" exact >
            Edit Book
          </NavLink>

          <button type="button" data-toggle="modal"
            data-target="#BookDeleteConfirmation"
            onClick={this.showDeleteConfin.bind(this, mybook.bookId)}
            className="btn btn-danger mytext-large mymargin1" >
            Delete Book
          </button>
        </td>
      </tr>
    );
  });

  return (
    <Container className="span-viewport d-flex flex-column align-items-center">
      <br />
      <div className="container">
        <div className="myfloat-left">
          <span className="text-center mytext-extra-large">
            Books @ Java Learning Center
          </span>
        </div>
        <div className="myfloat-right">
          <NavLink to={"/add-book"}
            className="btn btn-primary mytext-large mymargin1" exact >
            Add Book
          </NavLink>
        </div>
      </div>
    </div>
  );
}
```



```
<table className="table">
  <thead>
    <tr>
      <th> Book ID</th>
      <th> Book Name</th>
      <th> Author</th>
      <th> Price</th>
      <th> Category</th>
      <th> Publications</th>
      <th> </th>
    </tr>
  </thead>
  <tbody>{mybooksList}</tbody>
</table>

<Modal show={this.state.showConfirm} onHide={this.hideDeleteConfirm}>
  <Modal.Header closeButton>
    <Modal.Title>Confirmation</Modal.Title>
  </Modal.Header>
  <Modal.Body>
    <p>Are you sure to delete?</p>
    <b>{this.state.mybookIdToDelete}</b>
  </Modal.Body>
  <Modal.Footer>
    <Button variant="secondary" onClick={this.hideDeleteConfirm}>
      Close
    </Button>
    <Button variant="danger" onClick={this.handleDelete}>
      Delete
    </Button>
  </Modal.Footer>
</Modal>
</Container>
);
}
}
export default BooksList;
```



## 10. AddBook.js

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

import 'bootstrap/dist/css/bootstrap.min.css';
import MyTextInput from './MyTextInput';
import axios from 'axios';

class AddBook extends Component {

  state = {
    bookName:"",
    author:"",
    price:0,
    category:"",
    publications:"",
    errors :{}
  }

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

  onSubmitHandler = (event) => {

    event.preventDefault();
    const {bookName,author,price,category,publications} = this.state;

    console.log("onSubmitHandler");
    console.log(this.state);
    //Do the Validations

    if(bookName === ""){
      this.setState({errors : {
        bookName: "Book Name is Required"
      }});
      return ;
    }
    if(author === ""){
      this.setState({errors : {
        author: "Author is Required"
      }});
      return ;
    }
  }
}
```



```
if(price === ""){
  this.setState({errors : {
    price: "Price is Required"
  }});
  return ;
}
if(category === ""){
  this.setState({errors : {
    category: "Category is Required"
  }});
  return ;
}
if(publications === ""){
  this.setState({errors : {
    pub: "Publications is Required"
  }});
  return ;
}

//Make Call to Server

const URL="https://coursecube.com/jlc-bookstore/addBook";
axios.post(URL,this.state)
.then(myresponse => {
  console.log(1,myresponse.data);
  //Form Reset
  this.setState({
    mybook : { },
  });
  this.props.history.push("/");
}).catch(myerror => {
  console.log(2,myerror);
});
}

render(){
  const {bookName,author,price,category,publications} = this.state;

  return (
    <div className="card-body container col-md-6">
      <h2 className="text-center"> Add Book Form </h2>
      <form onSubmit={this.onSubmitHandler}>
```

```
<MyTextInput myname="bookName"
  mylabel="Book Name"
  myvalue={bookName}
  myerror={this.state.errors.bookName}
  myOnChange={this.onChangeHandler}/>

<MyTextInput myname="author"
  mylabel="Author"
  myvalue={author}
  myerror={this.state.errors.author}
  myOnChange={this.onChangeHandler}/>

<MyTextInput myname="price"
  mylabel="Price"
  myvalue={price}
  myerror={this.state.errors.price}
  myOnChange={this.onChangeHandler}/>

<MyTextInput myname="category"
  mylabel="Category"
  myvalue={category}
  myerror={this.state.errors.category}
  myOnChange={this.onChangeHandler}/>

<MyTextInput myname="publications"
  mylabel="Publications"
  myvalue={publications}
  myerror={this.state.errors.publications}
  myOnChange={this.onChangeHandler}/>

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





## 11. EditBook.js

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

import "bootstrap/dist/css/bootstrap.min.css";
import MyTextInput from "./MyTextInput";
import axios from "axios";
import classNames from "classnames";

class EditBook extends Component {
  state = {
    bookId: 0,
    bookName: "",
    author: "",
    price: 0,
    category: "",
    publications: "",
    errors: {},
  };

  componentDidMount() {
    console.log("componentDidMount", this.props);
    this.setState({
      bookId: this.props.match.params.mybookId,
    });
    const URL = `https://coursecube.com/jlc-bookstore/mybook/${this.props.match.params.mybookId}`;
    axios.get(URL).then((myresponse) => {
      console.log(myresponse.data);
      this.setState({
        bookId: myresponse.data.bookId,
        bookName: myresponse.data.bookName,
        author: myresponse.data.author,
        price: myresponse.data.price,
        category: myresponse.data.category,
        publications: myresponse.data.publications,
      });
    });
  }
}
```



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

```
onSubmitHandler = (event) => {  
  event.preventDefault();  
  const { bookName, author, price, category, publications } = this.state;  
  console.log("onSubmitHandler");  
  console.log(this.state);  
  //Do the Validations
```

```
  if (bookName === "") {  
    this.setState({  
      errors: {  
        bookName: "Book Name is Required",  
      },  
    });  
    return;  
  }
```

```
  if (author === "") {  
    this.setState({  
      errors: {  
        author: "Author is Required",  
      },  
    });  
    return;  
  }
```

```
  if (price === "") {  
    this.setState({  
      errors: {  
        price: "Price is Required",  
      },  
    });  
  }
```



```
        return;
    }

    if (category === "") {
        this.setState({
            errors: {
                category: "Category is Required",
            },
        });
        return;
    }

    if (publications === "") {
        this.setState({
            errors: {
                pub: "Publications is Required",
            },
        });
        return;
    }

    //Make Call to Server
    const URL = "https://coursecube.com/jlc-bookstore/updateBook";
    axios.put(URL, this.state)
        .then((myresponse) => {
            console.log(1, myresponse.data);
            //Form Reset
            this.setState({
                bookId: 0,
                bookName: "",
                author: "",
                price: 0,
                category: "",
                publications: "",
                errors: {},
            });
            this.props.history.push("/");
        })
    }
```



```
.catch((myerror) => {  
  console.log(2, myerror);  
});  
};
```

```
render() {  
  const { bookId, bookName, author, price, category, publications,} = this.state;  
  return (  
    <div className="card-body container col-md-6">  
      <h2 className="text-center"> Edit Book Form </h2>  
      <form onSubmit={this.onSubmitHandler}>  
        <div className="form-group">  
          <label htmlFor="bookId"> Book Id </label>  
          <input type="text" name="bookId" value={bookId} readOnly  
            className={classnames("form-control form-control-lg")} />  
        </div>  
        <MyTextInput  
          myname="bookName"  
          mylabel="Book Name"  
          myvalue={bookName}  
          myerror={this.state.errors.bookName}  
          myOnChange={this.onChangeHandler} />  
  
        <MyTextInput  
          myname="author"  
          mylabel="Author"  
          myvalue={author}  
          myerror={this.state.errors.author}  
          myOnChange={this.onChangeHandler} />  
  
        <MyTextInput  
          myname="price"  
          mylabel="Price"  
          myvalue={price}  
          myerror={this.state.errors.price}  
          myOnChange={this.onChangeHandler} />  
      </div>  
    );  
}
```



```
<MyTextInput
  myname="category"
  mylabel="Category"
  myvalue={category}
  myerror={this.state.errors.category}
  myOnChange={this.onChangeHandler} />

<MyTextInput
  myname="publications"
  mylabel="Publications"
  myvalue={publications}
  myerror={this.state.errors.publications}
  myOnChange={this.onChangeHandler} />

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

export default EditBook;
```