

SKILL-11

ID: 2100031023 Name: S

Likith Sai Reddy OUTPUT:

Book ID:

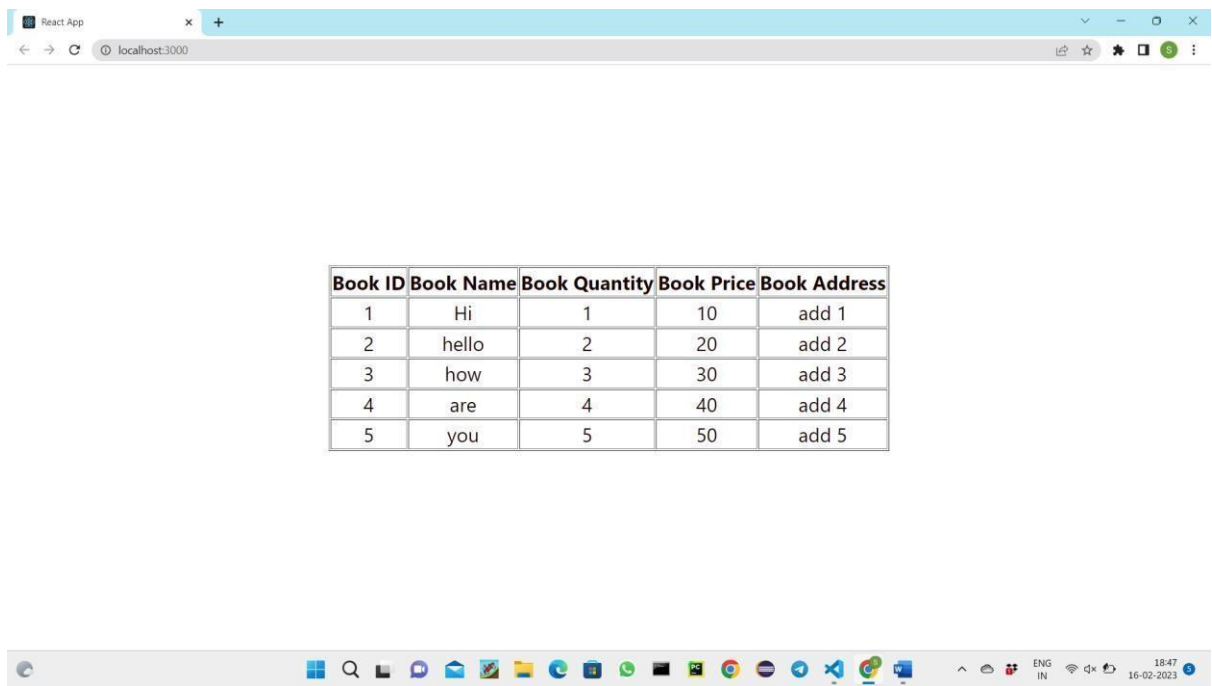
Book Name:

Price

Quantity

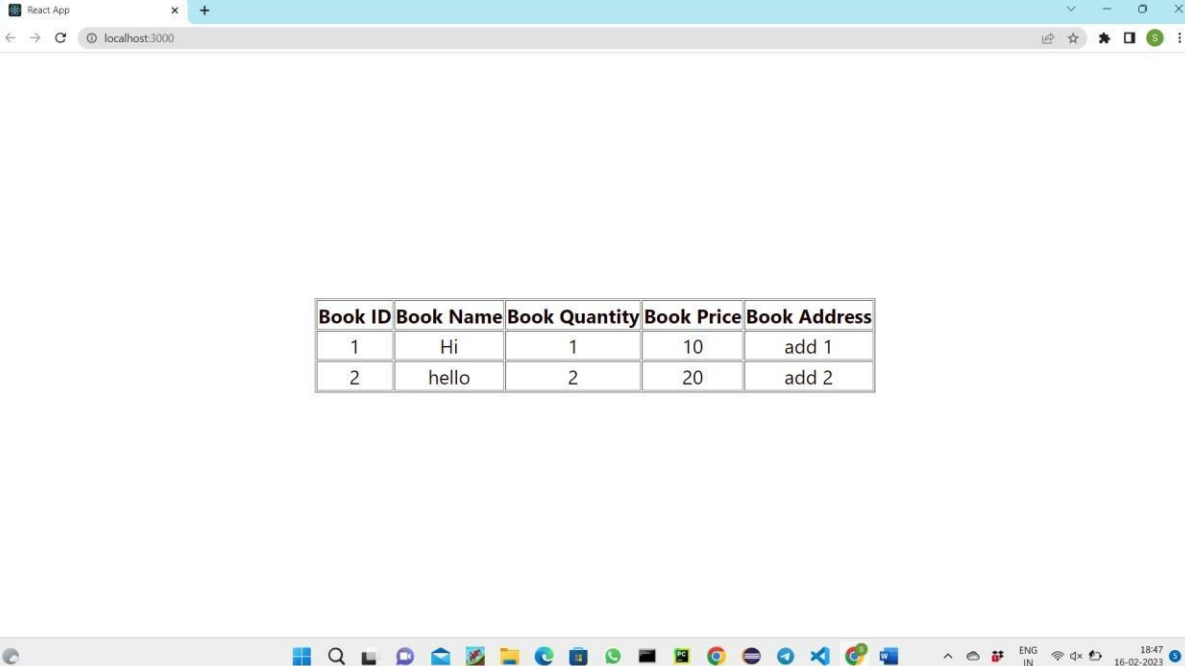
Address

Book ID	Book Name	Book Quantity	Book Price	Book Address
1	Hi	1	10	add 1

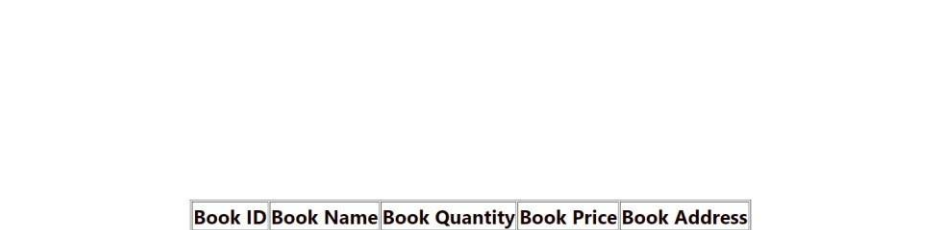


Book ID	Book Name	Book Quantity	Book Price	Book Address
2	hello	2	20	add 2
3	how	3	30	add 3
4	are	4	40	add 4
5	you	5	50	add 5
6	fine	6	60	add 6
7	what	7	70	add 7

Book ID	Book Name	Book Quantity	Book Price	Book Address
4	are	4	40	add 4
5	you	5	50	add 5
6	fine	6	60	add 6
7	what	7	70	add 7



Book ID	Book Name	Book Quantity	Book Price	Book Address
1	Hi	1	10	add 1
2	hello	2	20	add 2



The screenshot shows a web browser window with the address bar displaying 'localhost:3000'. The browser window contains a table with the following data:

Book ID	Book Name	Book Quantity	Book Price	Book Address
1	Hi	1	10	add 1
2	hello	2	20	add 2
3	how	3	30	add 3
4	are	4	40	add 4

React App

localhost:3000

Book ID	Book Name	Book Quantity	Book Price	Book Address
1	Hi	1	10	add 1
4	are	4	40	add 4
5	you	5	50	add 5
6	fine	6	60	add 6
7	what	7	70	add 7

16-02-2023

MSWD FINAL WORKBOOK (3).js

React App

localhost:3000

Book ID:

Book Name:

Price:

Quantity:

Address:

Save Student Data

Display First Book in Database

Display All Books

Display First 5 Books

Display All books where id greater than or equal to 2 Books

Display All books where id greater than 3 Books

Display All books where id less than or equal to 2 Books

Display All books where id less than 5 Books

Display All books where id greater than 3 or book price is equal to 10 Books

Update the book price to 100 where id is equal to 7

1 issue

react-devtools

Download the React DevTools for a better development experience: <https://reactjs.org/link/react-devtools>

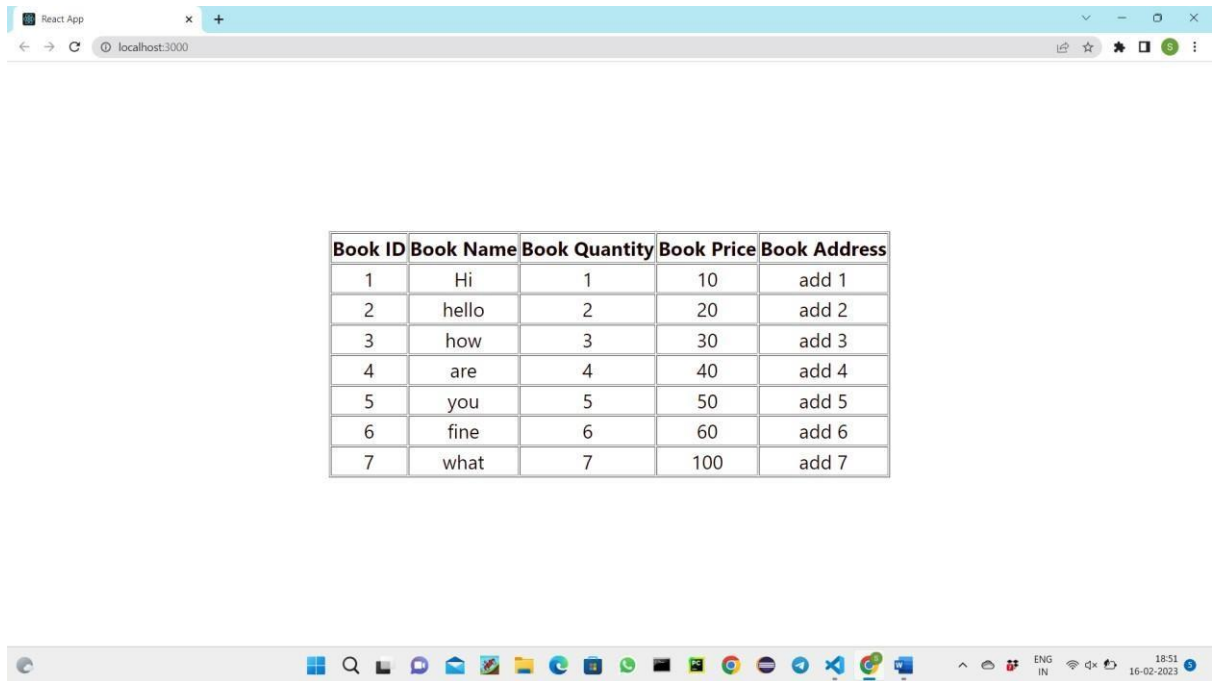
DevTools failed to load source map: Could not load content for chrome-extension://fhoegekfdchfhpcceifdbepaoicaho/sourceMap/chrome.js.criots/iiframe_form_check.map: System error: net::ERR_BLOCKED_BY_CLIENT

DevTools failed to load source map: Could not load content for chrome-extension://fhoegekfdchfhpcceifdbepaoicaho/sourceMap/chrome.js.criots/iiframe_form_detection.map: System error: net::ERR_BLOCKED_BY_CLIENT

Updated Successfully

18-59

16-02-2023



App.js:

```

// import logo from './logo.jpg'; // import klu from './klu.png';
import './App.css'; import Ex61 from './components/Ex61' import Ex63
from './components/Ex63' import { BrowserRouter, Routes, Route } from
'react-router-dom'; //npm i react-router-dom to install react-routerdom
package in client cmd
//BrowserRouter, Routes and Route are used to route from one page to another
like from signup to login etc.
// import { Container, Toolbar, AppBar } from '@material-ui/core'; //npm i
@material-ui/core --force to install @material-ui/core package in client cmd

function App() {
  // function opetation(){
  //   let vl=document.getElementById("Value");
  //   vl.value="1";
  // }
  return(
    <div className="App">

      <div className='App-body'>
        <BrowserRouter>
          <Routes>
            <Route path="/" element={<Ex61 />} />
            { /* <Route path='/ex63' element={<Ex63 />} /> */ }
          </Routes>
        </BrowserRouter>
      </div>
    </div>

  );
} export default
App;

```

App.css:

```

.App {  text-align: center;
}

.App-logo {

```



```
    height: 20vmin;    pointer-events:
none;
}

.App-klu {    height:
50px;    width: 150px;
pointer-events: none;
margin-left: 80px;
border-radius: 30px;
margin-right: 5px;
margin-bottom: 10px;
}

.App-kl{    display:
flex;    flexdirection:
row;    align-items:
center;    text-align:
center;
}

/* .App-header {
backgroundcolor: #282c34;    min-
height: 20vh;    display: flex;
alignitems: center;
justifycontent: center;    font-
size: calc(10px + 2vmin);
color:
white;
} */

.App-Header{    fontsize:
20px;    marginbottom:
10px;
}

.App-para{
fontsize: 15px;
display: flex;
flex-direction: row;
align-items: center;
text-align: center;
}

.App-color{    backgroundcolor:
white;
    /* background-color: #15b1ff; */    borderradius:
30px;
```



```
border :7px solid rgb(116, 116, 197);
padding: 10px;
}
```

```
.App-para1{
display: flex;
align-items: center;
margin-left: 60px;
color: blue;
fontweight: bold;
fontsize: 20px;
}
```

```
.App-para2{
display: flex;
align-items: center;
color: green;
margin-left: 70px;
font-weight: bold;
}
```

```
.App-para3{
display: flex;
align-items: center;
margin-left: 20px;
color: green;
fontweight: bold;
}
```

```
.App-para4{
display: flex;
align-items: center;
margin-left: 130px;
color: red;
fontweight: bold;
}
```

```
.App-body {
backgroundcolor:white; min-
height: 100vh; display: flex;
align-items: center;
justifycontent: center; font-
size: calc(10px + 2vmin);
color:
rgb(21, 1, 1);
}
```

```
.App-idcard {
```



```
/* border:1px solid red; */
```

```

    background-color: white;
padding: 20px;    height: 420px;
width: 350px;    display: block;
align-items: center;
justifycontent: center;    font-size:
calc(10px + 2vmin);    color:
rgb(21, 1, 1);
    /* box-shadow: 10px 10px 5px rgb(60, 206, 255) inset; */    box-shadow: 0
4px 8px 0 rgb(146, 252, 16), 0 6px 20px 0 rgb(16, 186, 233);    borderradius:
30px;    border :7px solid rgb(70, 255, 53); }


.App-inputval{
width:200px;    height:
50px;    margin-bottom:
10px;
}

.App-plus{
height: 50px;
width: 50px;
}
.App-zero{    height:
50px;    width: 50px;
margin-right: 50px;
margin-left: 50px;
}

.App-link {
color: #61dafb;
}

@keyframes App-logo-spin {
from {    transform:
rotate(0deg);
    }    to {    transform:
rotate(360deg);
    }
}

```



Ex61.js:



```

import { RepeatOneSharp } from
"@mui/iconsmaterial"; import axios from
"axios"; import { useState } from "react";
function Ex61()
{
    const [result,setResult] = useState(null);
    const
isDict = dict => {
        return typeof dict === "object" &&
!Array.isArray(dict);
    };
    function handleSubmit(event)
    {
        event.preventDefault();
        const data = new
FormData(event.currentTarget);
        axios.post('http://localhost:8086/insert_book',
        {
            book_id: parseInt(data.get("book_id")),
            book_name:
            data.get("book_name"),
            book_price:
            data.get("book_price"),
            book_quantity:
            data.get("book_quantity"),
            book_address:
            data.get("book_address")
        }).then((resposnse)=>{
            console.log(resposnse.data);
        })
    }
    function showOne() {
        axios.get('http://localhost:8086/show_one_book', {params:{
            }}).then((response)=>{
                console.log(response.data)
                setResult(response.data)
            })
    }
    function showAll() {
        axios.get('http://localhost:8086/show_all_book', {params:{
            }}).then((response)=>{
                console.log(response.data)
                setResult(response.data)
            })
    }
    function
showAlllte5() {
        axios.get('http://localhost:8086/show_all_lte5_book', {params:{

```

```

        })).then((response)=>{

```



```

        console.log(response.data)
    setResult(response.data)
    })
    }
    function showAllgte2() {
    axios.get('http://localhost:8086/show_all_gte2_book', {params:{
        }}).then((response)=>{
    console.log(response.data)
    setResult(response.data)
    })
    }
    function showAllgt3() {
    axios.get('http://localhost:8086/show_all_gt3_book', {params:{
        }}).then((response)=>{
    console.log(response.data)
    setResult(response.data)
    })
    }
    function showAlllte2() {
    axios.get('http://localhost:8086/show_all_lte2_book', {params:{
        }}).then((response)=>{
    console.log(response.data)
    setResult(response.data)
    })
    }
    function showAlllt5() {
    axios.get('http://localhost:8086/show_all_lt5_book', {params:{
        }}).then((response)=>{
    console.log(response.data)
    setResult(response.data)
    })
    }
    function showAllgt3orbookpriceet10() {
    axios.get('http://localhost:8086/show_all_gt3_or_bookprice_equalto10_b    ook',
    {params:{

        }}).then((response)=>{

```

```
console.log(response.data)
```

```
setResult(response.data)
```

```

    })
  }
  function updateid7() {
    console.log("upd")
    axios.patch('http://localhost:8086/show_all_update', {params:{
      id: result
    }}).then((response)=>{
      console.log(response.data)
      // setResult(response.data)
    })
  }
  if(result ==
null) {
    return(
      <div>
        <form onSubmit={handleSubmit}>
          Book ID: <input type="text" name="book_id" />
          <br/>
          Book Name: <input type="text" name="book_name" />
          <br/>
          Price <input type="text" name="book_price" />
          <br/>
          Quantity <input type="text" name="book_quantity" />
          <br/>
          Address <input type="text" name="book_address" />
          <br/><br/>
          <input type="submit" value="Save Student Data" />
        </form>
        <br/>
        <button onClick={()=>showOne()}>Display First Book in
Database</button>
        <br/>
        <button onClick={()=>showAll()}>Display All Books</button>
        <br/>
        <button onClick={()=>showAlllte5()}>Display First 5
Books</button>
        <br/>
        <button onClick={()=>showAllgte2()}>Display All books where id
greater than or equal to 2 Books</button>
        <br/>
        <button onClick={()=>showAllgt3()}>Display All books where id
greater than 3 Books</button>
        <br/>
        <button onClick={()=>showAlllte2()}>Display All books where id
less than or equal to 2 Books</button>
        <br/>
      </div>
    )
  }
}

```

```
<button onClick={()=>showAlllt5()}>Display All books where id
```



```

less than 5 Books</button>
    <br/>
    <button onClick={()=>showAllgt3orbookpriceet10()}>Display All
books where id greater than 3 or book price is equal to 10 Books</button>
    <br/>
    <button onClick={()=>updateid7()}>Update the book price to 100
where id is equal to 7 </button>
    <br/>
  </div>
);    }    else {
if(isDict(result)) {
return (
    <div>
      <table border="1">
        <tr>
          <th> Book ID </th>
          <th> Book Name </th>
          <th> Book Quantity </th>
          <th> Book Price </th>
          <th> Book Address </th>
        </tr>
        <tr>
          <td> {result.book_id} </td>
          <td> {result.book_name} </td>
          <td> {result.book_quantity} </td>
          <td> {result.book_price} </td>
          <td> {result.book_address} </td>
        </tr>
      </table>
    </div>
  );
}
else {
console.log("java");
return (
    <div>
      <table border="1">
        <tr>
          <th> Book ID </th>
          <th> Book Name </th>
          <th> Book Quantity </th>
          <th> Book Price </th>
          <th> Book Address </th>
        </tr>
        <tr>
          <td> {result.map((obj)=>{
return(

```

```

        <tr>
            <td> {obj.book_id} </td>
            <td> {obj.book_name} </td>
            <td> {obj.book_quantity} </td>
            <td> {obj.book_price} </td>
            <td> {obj.book_address} </td>
        </tr>
    </table>
    </div>
);
}
}
} export default Ex61;

```

Server.js:

```
const express = require('express') const cors =
require('cors') const { MongoClient } =
require('mongodb') const { request, response }
= require('express') const app = express()
app.use(cors()) app.use(express.json()) const
uri
=
"mongodb+srv://admin:rocky9886@cluster0.nczlwja.mongodb.net/?retryWrites=true&
w=majority" const client = new MongoClient(uri) client.connect()
//Experiment 6-1 var db =
client.db("ex6")
var col = db.collection("costumer")

app.get('/',(request,response) => {
response.send('This is a Server')
}) app.post('/insert_book', (request,response) =>
{
```

```
col = db.collection("costumer")
```

```
console.log(request.body)
```

```

col.insertOne(request.body)
response.send(request.body)
}) app.get('/show_one_book', (req, res)=>{
col = db.collection("costumer")
console.log(req.query)    async function
find(){    try{    const
result=await col.findOne()
console.log(result)    if(result
== null)
    {
res.send({"Data_Retrieval":"Fail"})
    }    else{
res.send(result)
    }
}    finally{}
}
find().catch(console.dir)
}) app.get('/show_all_book', (req, res)=>{
col = db.collection("costumer")
console.log(req.query)    async function find(){
try{    const result=await
col.find().toArray()
console.log(result)    if(result == null)
    {
console.log("sever");
res.send({"Data_Retrieval":"Fail"})
    }    else{
res.send(result)
    }
}
}
finally{}
}
find().catch(console.dir)
})

```

```

app.get('/show_all_lte5_book', (req, res)=>{

```



```
col = db.collection("costumer")    console.log(req.query)
```

```

async function find(){          try{                const result=await
col.find({book_id:{$lte:5}}).toArray()
console.log(result)              if(result == null)
{
res.send({"Data_Retrieval":"Fail"})
}                                else{
res.send(result)
}
}                                finally{}
}
find().catch(console.dir)
})

app.get('/show_all_gte2_book', (req, res)=>{    col =
db.collection("costumer")    console.log(req.query)    async
function find(){                try{                const result=await
col.find({book_id:{$gte:2}}).toArray()
console.log(result)              if(result == null)
{
res.send({"Data_Retrieval":"Fail"})
}                                else{
res.send(result)
}
}
}
finally{}
}
find().catch(console.dir)
}) app.get('/show_all_gt3_book', (req, res)=>{    col =
db.collection("costumer")    console.log(req.query)
async function find(){          try{                const
result=await col.find({book_id:{$gt:3}}).toArray()
console.log(result)              if(result == null)
{

```




```

        res.send({"Data_Retrieval":"Fail"})
    }
    else{
        res.send(result)
    }
}
finally{}
}
find().catch(console.dir)
}) app.get('/show_all_lte2_book', (req, res)=>{    col =
db.collection("costumer")    console.log(req.query)    async
function find(){        try{            const result=await
col.find({book_id:{$lte:2}}).toArray()
console.log(result)            if(result == null)
        {
res.send({"Data_Retrieval":"Fail"})
        }            else{
res.send(result)
        }
    }
}
finally{}
}
find().catch(console.dir)
})

app.get('/show_all_lt5_book', (req, res)=>{
col = db.collection("costumer")
console.log(req.query)    async function find(){
try{        const result=await
col.find({book_id:{$lt:5}}).toArray()
console.log(result)        if(result ==
null)
        {
res.send({"Data_Retrieval":"Fail"})        }
    else{        res.send(result)
        }
    }
}
finally{}
}

```



```

        find().catch(console.dir)
    }) app.get('/show_all_gt3_or_bookprice_equalto10_book', (req,
res)=>{      col = db.collection("costumer")
console.log(req.query)      async function find(){          try{
const result=await
col.find({$or:[{book_price:"10"},{book_id:{$gt:3}}]}).toArray()
console.log(result)          if(result == null)
            {
res.send({"Data_Retrieval":"Fail"})
            }          else{
res.send(result)
            }
        }
    }
finally{}
}
find().catch(console.dir)
}) app.patch('/show_all_update',(req,
res)=>{      console.log("update")      col
= db.collection("costumer")
console.log(req.body)      const doc = {
        $set: {book_price:"100"}
    }
col.updateOne({book_id:7},doc)
res.send("Updated Successfully")

// console.log(req.query)
// async function find(){
//   try{
//     const result=await col.find().toArray()
//     console.log(result)
//     if(result == null)
//     {
//       console.log("sever");
//       res.send({"Data_Retrieval":"Fail"})
//     }
//     else{
//       res.send(result)
//     }
//   }
// }

```



```
    // finally{}  
    // }  
    // find().catch(console.dir)  
  })  
app.listen(8086) console.log("server started")
```