

WEB TECHNOLOGY LAB

LAB-8

Design and develop a responsive website for an online book store using REACT, Springboot and MySQL having 1) Home Page 2) Login Page 3) Catalogue Page: 4) Registration Page: (database)

Shruti Ajaykumar Singh

TY-CD-D2

Roll No – 25

PRN - 12010587

Batch – 2

CODE:

Book.java

```
package com.java.books.domain;

import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;

@Entity
public class Book {

    @Id
    @GeneratedValue(strategy=GenerationType.AUTO)
    private Long id;
    private String title;
    private String author;
    private String coverPhotoURL;
    private Long isbnNumber;
    private Double price;
    private String language;

    public Long getId() {
        return id;
    }

    public void setId(Long id) {
        this.id = id;
    }

    public String getTitle() {
        return title;
    }
}
```

```
}  
public void setTitle(String title) {  
    this.title = title;  
}  
public String getAuthor() {  
    return author;  
}  
public void setAuthor(String author) {  
    this.author = author;  
}  
public String getCoverPhotoURL() {  
    return coverPhotoURL;  
}  
public void setCoverPhotoURL(String coverPhotoURL) {  
    this.coverPhotoURL = coverPhotoURL;  
}  
public Long getIsbnNumber() {  
    return isbnNumber;  
}  
public void setIsbnNumber(Long isbnNumber) {  
    this.isbnNumber = isbnNumber;  
}  
public Double getPrice() {  
    return price;  
}  
public void setPrice(Double price) {  
    this.price = price;  
}  
public String getLanguage() {  
    return language;  
}  
public void setLanguage(String language) {  
    this.language = language;  
}  
}
```

BookRepository.java

```
package com.java.books.repository;

import org.springframework.data.repository.CrudRepository;

import com.java.books.domain.Book;

public interface BookRepository extends CrudRepository<Book, Long>{

}
```

BookResourceImpl.java

```
package com.java.books.resource;

import java.util.Collection;
import java.util.Optional;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.HttpStatus;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.CrossOrigin;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

import com.java.books.domain.Book;
import com.java.books.repository.BookRepository;

@RestController
@RequestMapping("/books")
@CrossOrigin(origins="http://localhost:3000")
public class BookResourceImpl implements Resource<Book> {
```

```
@Autowired
private BookRepository bookRepo;

@Override
public ResponseEntity<Optional<Book>> findById(Long id) {
    return new ResponseEntity<Optional<Book>>((Optional<Book>) bookRepo.findById(id), HttpStatus.OK);
}

@Override
public ResponseEntity<Collection<Book>> findAll() {
    return new ResponseEntity<Collection<Book>>((Collection<Book>) bookRepo.findAll(), HttpStatus.OK);
}

@Override
public ResponseEntity<Book> save(@RequestBody Book book){
    return new ResponseEntity<>(bookRepo.save(book), HttpStatus.CREATED);
}

@Override
public ResponseEntity<Book> update(Book book) {
    return new ResponseEntity<>(bookRepo.save(book), HttpStatus.OK); // only difference is status code
}

@Override
public ResponseEntity<Book> deleteById(Long id) {
    if(bookRepo.existsById(id)) {
        bookRepo.deleteById(id);
        return new ResponseEntity<Book>(HttpStatus.OK);
    }
    return new ResponseEntity<Book>(HttpStatus.NOT_FOUND);
}
}
```

Resource.java

```
package com.java.books.resource;

import java.util.Collection;
import java.util.Optional;

import org.springframework.http.MediaType;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.DeleteMapping;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.PutMapping;
import org.springframework.web.bind.annotation.RequestBody;

public interface Resource<T> {

    @GetMapping
    public ResponseEntity<Collection<T>> findAll();

    @GetMapping("/{id}")
    public ResponseEntity<Optional<T>> findById(@PathVariable Long id);

    @PostMapping(consumes = MediaType.APPLICATION_JSON_VALUE)
    public ResponseEntity<T> save(@RequestBody T t);

    @PutMapping(consumes = MediaType.APPLICATION_JSON_VALUE)
    public ResponseEntity<T> update(@RequestBody T t);

    @DeleteMapping("/{id}")
    public ResponseEntity<T> deleteById(@PathVariable Long id);
}
```

BookApplication.java

```
package com.java.books;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication
public class BooksApplication {

    public static void main(String[] args) {
        SpringApplication.run(BooksApplication.class, args);
    }

}
```

App.js

```
import React from 'react';
import './App.css';
import './components/NavigationBar'
import NavigationBar from './components/NavigationBar';
import { Container, Row, Col } from 'react-bootstrap'
import { BrowserRouter as Router, Switch, Route } from 'react-router-dom'

import Welcome from './components/Welcome'
import Footer from './components/Footer'
import BookList from './components/BookList'
import Book from './components/Book'
import UserList from './components/UserList';

function App() {
    const marginTop = {
        marginTop: "20px"
    }
}
```

```

const heading = "Welcome to Book Shop";
const quote = "Good friends, a sleepy conscience: this is the ideal life.";
const footer = "Mark Twain";

return (
  <Router>
    <NavigationBar />
    <Container>
      <Row>
        <Col lg={12} style={marginTop}>
          <Switch>
            <Route path="/" exact component={() => <Welcome heading={heading} quote={quote} footer={footer}/>}/>
            <Route path='/add' exact component={Book}/>
            <Route path='/edit/:id' exact component={Book}/>
            <Route path='/list' exact component={BookList}/>
            <Route path="/users" exact component={UserList}/>
          </Switch>
        </Col>
      </Row>

      </Container>
      <Footer />
    </Router>

  );
}

export default App;

```

App.css

```

.App {
  text-align: center;
}

```



```
.App-logo {
  height: 40vmin;
  pointer-events: none;
}

@media (prefers-reduced-motion: no-preference) {
  .App-logo {
    animation: App-logo-spin infinite 20s linear;
  }
}

.App-header {
  background-color: #282c34;
  min-height: 100vh;
  display: flex;
  flex-direction: column;
  align-items: center;
  justify-content: center;
  font-size: calc(10px + 2vmin);
  color: white;
}

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

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

Book.js

```
import React, { Component } from 'react'
import { Card, Form, Button, Col } from 'react-bootstrap';
import { FontAwesomeIcon } from '@fortawesome/react-fontawesome'
import { faPlusSquare, faSave, faUndo, faList, faEdit } from '@fortawesome/free-solid-svg-icons'
import axios from 'axios'
import MyToast from './MyToast'

class Book extends Component {
  constructor(props) {
    super(props)
    this.state = this.initialState;
    this.bookChange = this.bookChange.bind(this);
    this.submitBook = this.submitBook.bind(this);
  }

  initialState = {
    id: '',
    title: '',
    author: '',
    coverPhotoURL: '',
    isbnNumber: '',
    language: '',
    price: '',
  }

  componentDidMount() {
    const bookId = +this.props.match.params.id;
    if (bookId) {
      this.findBookById(bookId);
    }
  }

  findBookById = (bookId) => {
```

```
fetch("http://localhost:8080/api/v1/books/" + bookId)
  .then(response => response.json())
  .then((book) => {
    this.setState({
      id: book.id,
      title: book.title,
      author: book.author,
      coverPhotoURL: book.coverPhotoURL,
      isbnNumber: book.isbnNumber,
      price: book.price,
      language: book.language
    });
  }).catch((error) => {
    console.error("Error - " + error);
  });
}

submitBook = event => {
  event.preventDefault();
  const book = {
    title: this.state.title,
    author: this.state.author,
    isbnNumber: this.state.isbnNumber,
    coverPhotoURL: this.state.coverPhotoURL,
    language: this.state.language,
    price: this.state.price
  };

  const headers = new Headers();
  headers.append('Content-Type', 'application/json');

  fetch("http://localhost:8080/api/v1/books", {
    method: 'POST',
    body: JSON.stringify(book),
    headers
  })
  .then(response => response.json())
```

```

.then((book) =>{
  if(book !== null){
    this.setState({"show" : "true", "method" : "true"});
    setTimeout(() => this.setState({"show" : "false"}), 3000);
  }else{
    this.setState({"show" : "false"});
  }
});
}

```

```

updateBook = event => {
  event.preventDefault();
  const book = {
    id: this.state.id,
    title: this.state.title,
    author: this.state.author,
    coverPhotoURL: this.state.coverPhotoURL,
    isbnNumber: this.state.isbnNumber,
    price: this.state.price,
    language: this.state.language
  };
  const headers = new Headers();
  headers.append('Content-Type', 'application/json');
  fetch("http://localhost:8081/api/v1/books", {
    method: 'PUT',
    body: JSON.stringify(book),
    headers
  })
  .then(response => response.json())
  .then((book) => {
    if(book) {
      this.setState({"show":true, "method":"put"});
      setTimeout(() => this.setState({"show":false}), 3000);
      setTimeout(() => this.bookList(), 3000);
    } else {
      this.setState({"show":false});
    }
  })
}

```

```

    }
  });
  this.setState(this.initialState);
};
resetBook = () => {
  this.setState(() => this.initialState);
}

bookChange = event => {
  this.setState({
    [event.target.name]: event.target.value
  });
}

bookList = () => {
  return this.props.history.push("/list");
}

render() {
  const { title, author, isbnNumber, coverPhotoURL, language, price } = this.state;
  return (
    <div>
      <div style={{ "display": this.state.show ? "block" : "none" }}>
        <MyToast show={this.state.show} message={this.state.method === "put" ? "Book updated successfully" : "Book saved successfully"} type={"success"} />
      </div>
      <Card className={"border border-dark bg-dark text-white"}>
        <Card.Header><FontAwesomeIcon icon={this.state.id ? faEdit : faPlusSquare} /> {this.state.id ? "Update book" : "Add book "}
      </Card.Header>
      <Form onReset={this.resetBook} onSubmit={this.state.id ? this.updateBook : this.submitBook} id='bookFormId'>
        <Card.Body>
          <Form.Row>
            <Form.Group as={Col} controlId='formGridTitle'>
              <Form.Label>Title</Form.Label>
              <Form.Control required autoComplete="off"

```

```

        type="text" name='title'
        value={title}
        onChange={this.bookChange}
        className={'bg-dark text-white'}
        placeholder="Enter book title" />
    </Form.Group>
    <Form.Group as={Col} controlId='formGridAuthor'>
        <Form.Label>Author</Form.Label>
        <Form.Control required autoComplete="off"
            type="text" name='author'
            value={author}
            onChange={this.bookChange}
            className={'bg-dark text-white'}
            placeholder="Enter Author name" />
    </Form.Group>
</Form.Row>
<Form.Row>
    <Form.Group as={Col} controlId='formGridISBNNumber'>
        <Form.Label>ISBN Number</Form.Label>
        <Form.Control required autoComplete="off"
            type="text" name='isbnNumber'
            value={isbnNumber}
            onChange={this.bookChange}
            className={'bg-dark text-white'}
            placeholder="Enter ISBN number" />
    </Form.Group>
    <Form.Group as={Col} controlId='formGridcoverPhotoURL'>
        <Form.Label>Cover pic URL</Form.Label>
        <Form.Control required autoComplete="off"
            type="text" name='coverPhotoURL'
            value={coverPhotoURL}
            onChange={this.bookChange}
            className={'bg-dark text-white'}
            placeholder="Enter cover pic URL" />
    </Form.Group>
</Form.Row>

```

```

    <Form.Row>
      <Form.Group as={Col} controlId='formGridLanguage'>
        <Form.Label>Language</Form.Label>
        <Form.Control required autoComplete="off"
          type="text" name='language'
          value={language}
          onChange={this.bookChange}
          className={'bg-dark text-white'}
          placeholder="Enter Language" />
      </Form.Group>
      <Form.Group as={Col} controlId='formGridPrice'>
        <Form.Label>Price</Form.Label>
        <Form.Control required autoComplete="off"
          type="text" name='price'
          value={price}
          onChange={this.bookChange}
          className={'bg-dark text-white'}
          placeholder="Enter Price" />
      </Form.Group>
    </Form.Row>
  </Card.Body>
  <Card.Footer style={{ 'textAlign': 'right' }}>
    <Button variant="primary" type="submit">
      <FontAwesomeIcon icon={faSave} /> {this.state.id ? "Update" : "Save "}
    </Button>{' '}
    <Button variant="info" type="reset">
      <FontAwesomeIcon icon={faUndo} /> Reset
    </Button>
    {' '}
    <Button variant="info" type="button" onClick={this.bookList.bind()}>
      <FontAwesomeIcon icon={faList} /> Book List
    </Button>
  </Card.Footer>
</Form>
</Card>
</div>

```

```
    )  
  }  
}
```

```
export default Book
```

BookList.js

```
import React, { Component } from 'react'  
import { Card, Table, Image, ButtonGroup, Button } from 'react-bootstrap';  
import { FontAwesomeIcon } from '@fortawesome/react-fontawesome'  
import { faList, faEdit, faTrash } from '@fortawesome/free-solid-svg-icons'  
import { Link } from 'react-router-dom'  
import axios from 'axios';  
import MyToast from './MyToast'
```

```
class BookList extends Component {  
  constructor(props) {  
    super(props)  
  
    this.state = {  
      books: []  
    }  
  }  
  
  componentDidMount() {  
    this.findAllBooks();  
  }  
  
  findAllBooks(){  
    fetch("http://localhost:8080/api/v1/books")  
      .then(response => response.json())  
      .then((data) => {
```



```

        <th>ISBN Number</th>
        <th>Price</th>
        <th>Language</th>
        <th>Action</th>
    </tr>
</thead>
<tbody>
    {
        this.state.books.length === 0 ?
        <tr align='center'>
            <td colSpan='6'> Books Available</td>
        </tr> :
        this.state.books.map((book) => (
            < tr key={book.id}>
                <td>
                    <Image src={book.coverPicURL} roundedCircle width="25" height="25"/>
                    {' '} {book.title}
                </td>
                <td>{book.author}</td>
                <td>{book.isbnNumber}</td>
                <td>{book.language}</td>
                <td>{book.price}</td>
                <td>
                    <ButtonGroup>
                        <Link to={"edit/"+book.id} className="btn btn-sm btn-outline-
primary"><FontAwesomeIcon icon={faEdit} /></Link>{' '}
                        <Button size="sm" variant="outline-danger" onClick={this.deleteBook.bind(this,
book.id)}><FontAwesomeIcon icon={faTrash} /></Button>
                    </ButtonGroup>
                </td>
            </tr>

        ))
    }
</tbody>

```

```

        </Table>
      </Card.Body>
    </Card >
  </div>

  )
}
}

```

```
export default BookList
```

Welcome.js

```

import React from 'react'
import {Jumbotron} from 'react-bootstrap'

function Welcome(props) {
  return (
    <Jumbotron className="bg-dark text-white">
      <h1>{props.heading}</h1>
      <blockquote className="blockquote mb-0">
        <p>
          {props.quote}
        </p>
        <footer className="blockquote-footer">
          {props.footer}
        </footer>
      </blockquote>
    </Jumbotron>
  );
}

export default Welcome

```

NavigationBar.js

```
import React from 'react'
import { Navbar, Nav } from 'react-bootstrap'
import { Link } from 'react-router-dom'

function NavigationBar() {
  return (
    <Navbar bg="dark" variant="dark">
      <Link to="" className="navbar-brand">
        
        Book Shop
      </Link>
      <Nav className="mr-auto">
        <Link to={"add"} className="nav-link">Add Book</Link>
        <Link to={"list"} className="nav-link">Book List</Link>
        <Link to={"users"} className="nav-link">User List</Link>
      </Nav>
    </Navbar>
  )
}

export default NavigationBar
```

UserList.js

```
import React, {Component} from 'react';
import {Card, Table, InputGroup, FormControl, Button} from 'react-bootstrap';
import {FontAwesomeIcon} from '@fortawesome/react-fontawesome';
import {faUsers, faStepBackward, faFastBackward, faStepForward, faFastForward} from '@fortawesome/free-solid-svg-icons';
import axios from 'axios'
```

```
class UserList extends Component {
  constructor(props) {
    super(props);
    this.state = {
      users : [],
      currentPage : 1,
      usersPerPage : 5
    };
  }

  componentDidMount() {
    this.findAllRandomUsers();
  }

  changePage = event => {
    this.setState({
      [event.target.name]: parseInt(event.target.value)
    });
  };

  firstPage = () => {
    if(this.state.currentPage > 1) {
      this.setState({
        currentPage: 1
      });
    }
  };

  prevPage = () => {
    if(this.state.currentPage > 1) {
      this.setState({
        currentPage: this.state.currentPage - 1
      });
    }
  };
};
```

```
lastPage = () => {
  if(this.state.currentPage < Math.ceil(this.state.users.length / this.state.usersPerPage)) {
    this.setState({
      currentPage: Math.ceil(this.state.users.length / this.state.usersPerPage)
    });
  }
};

nextPage = () => {
  if(this.state.currentPage < Math.ceil(this.state.users.length / this.state.usersPerPage)) {
    this.setState({
      currentPage: this.state.currentPage + 1
    });
  }
};

findAllRandomUsers() {
  axios.get("https://randomapi.com/api/6de6abfedb24f889e0b5f675edc50deb?fmt=raw&sole")
    .then(response => response.data)
    .then((data) => {
      this.setState({users: data});
    });
};

render() {
  const {users, currentPage, usersPerPage} = this.state;
  const lastIndex = currentPage * usersPerPage;
  const firstIndex = lastIndex - usersPerPage;
  const currentUsers = users.slice(firstIndex, 1);
  const totalPages = users.length / usersPerPage;

  const pageNumCss = {
    width: "45px",
    border: "1px solid #17A2B8",
    color: "#17A2B8",
    textAlign: "center",
  }
}
```



```

</Card.Body>
{users.length > 0 ?
<Card.Footer>
  <div style={{ "float": "left" }}>
    Showing page {currentPage} of {totalPages}
  </div>
  <div style={{ "float": "right" }}>
    <InputGroup>
      <InputGroup.Prepend>
        <Button type="button" variant="outline-info" disabled={currentPage === 1 ? true : false}
          onClick={this.firstPage}>
          <FontAwesomeIcon icon={faFastBackward} /> First
        </Button>
        <Button type="button" variant="outline-info" disabled={currentPage === 1 ? true : false}
          onClick={this.prevPage}>
          <FontAwesomeIcon icon={faStepBackward} /> Prev
        </Button>
      </InputGroup.Prepend>
      <FormControl style={pageNumCss} className={"bg-dark"} name="currentPage" value={currentPage}
        onChange={this.changePage}/>
      <InputGroup.Append>
        <Button type="button" variant="outline-info" disabled={currentPage === totalPages ? true :
false}
          onClick={this.nextPage}>
          <FontAwesomeIcon icon={faStepForward} /> Next
        </Button>
        <Button type="button" variant="outline-info" disabled={currentPage === totalPages ? true :
false}
          onClick={this.lastPage}>
          <FontAwesomeIcon icon={faFastForward} /> Last
        </Button>
      </InputGroup.Append>
    </InputGroup>
  </div>
</Card.Footer>
:

```



```

        null}
      </Card>
    </div>
  )
}
}
export default UserList

```

NavigationBar.js

```

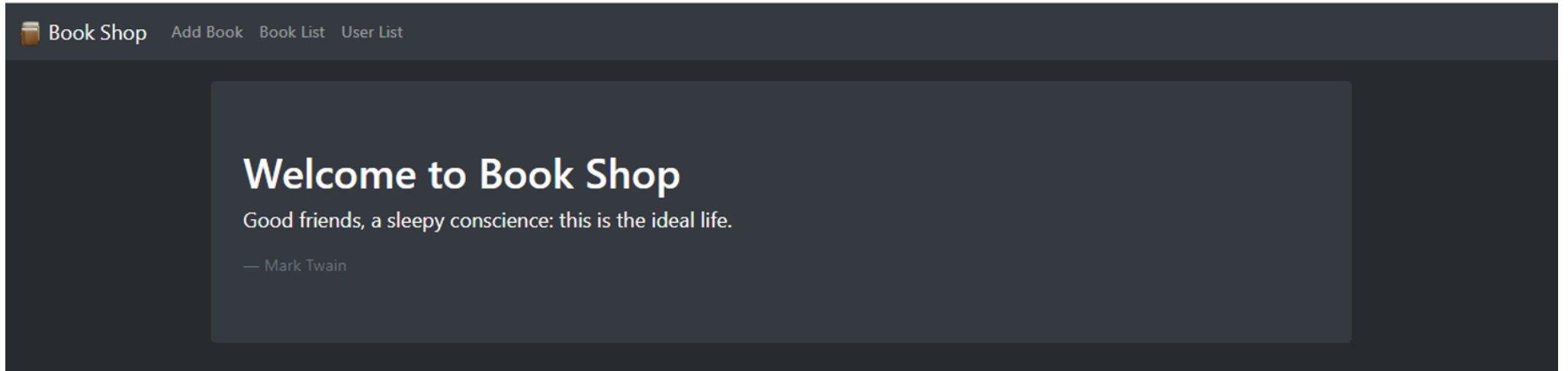
import React from 'react'
import { Navbar, Nav } from 'react-bootstrap'
import { Link } from 'react-router-dom'

function NavigationBar() {
  return (
    <Navbar bg="dark" variant="dark">
      <Link to="/" className="navbar-brand">
        
        Book Shop
      </Link>
      <Nav className="mr-auto">
        <Link to="/add" className="nav-link">Add Book</Link>
        <Link to="/list" className="nav-link">Book List</Link>
        <Link to="/users" className="nav-link">User List</Link>
      </Nav>
    </Navbar>
  )
}
export default NavigationBar


```

OUTPUT:

Home Page –



Add New Book –

 Book Shop

Add BookBook ListUser List

+ Add book

TitleEasy Vegetarian Slow Cooker CookbookAuthorRockridge Press

ISBN Number1234567894Cover pic URLhttps://images-na.ssl-images-amazon.com/images/S/compressed.photo

LanguageEnglishPrice700

SaveResetBook List



Success

Book sav successfully

+ Add book

Title

Easy Vegetarian Slow Cooker Cookbook

Author

Rockridge Press

ISBN Number

1234567894

Cover pic URL

https://images-na.ssl-images-amazon.com/images/S/compressed.photo

Language

English

Price


700

Save


Reset




Book List

Book Catalogue –


 Book Shop


[Add Book](#) [Book List](#) [User List](#)

 Book List

Title	Author	ISBN Number	Price	Language	Action
 Easy Vegetarian Slow Cooker Cookbook	Rockridge Press	1234567894	English	700	 

Update Book Information –

 Book Shop Add Book Book List User List

 Update book


Title	Author
<input type="text" value="Easy Vegetarian Slow Cooker Cookbook"/>	<input type="text" value="Rockridge Press"/>
ISBN Number	Cover pic URL
<input type="text" value="1234567894"/>	<input type="text" value="https://images-na.ssl-images-amazon.com/images/S/compressed.photo"/>
Language	Price
<input type="text" value="English"/>	<input type="text" value="900"/>

Update

Reset

Book List



User List –

 Book Shop Add Book Book List User List

User List

Name	Email	Address	Created	Balance
Flo Maggio	Flo.Maggio@waldo.name	95277 Predovic Hill	July 29, 2013	\$1,145.97
Casper Zboncak	Casper.Zboncak@roslyn.biz	73845 Amanda Via	August 8, 2013	\$7,343.39
Cade Kling	Cade.Kling@ramona.biz	3925 Marshall Squares	September 16, 2022	\$6,925.19
Modesta Haley	skybluefrog03@gmail.com	85442 Gibson Green	July 16, 2017	\$1,252.21
Rosemary Thiel	Rosemary.Thiel@darlene.org	2374 Electa Spur	September 11, 2015	\$4,684.46

Showing page 1 of 15

 First  Prev **1**  Next  Last