**CODE :**

**addBook.jsx**

import React, { useState } from "react";

import { Button, Form, InputGroup } from "react-bootstrap";

import { bookService } from "../Services/firebase-services";

const AddBook = () => {

const [formData, setformData] = useState({

title: "",

author: "",

status: "",

});

const handleSubmit = (e) => {

e.preventDefault();

addBook();

};

const addBook = () => {

const newBook = {

title: formData.title,

author: formData.author,

status: formData.status,

};

var ret = bookService.addBook(newBook);

console.log(ret);

};

const handleChange = (e) => {

const { name, value } = e.target;

setformData((prevData) => ({ ...prevData, [name]: value }));

console.log(formData);

};

return (

<div className="p-4 box">

<Form onSubmit={handleSubmit}>

<Form.Group>

<InputGroup>

<InputGroup.Text id="formBookTitle">Title</InputGroup.Text>

<Form.Control

name="title"

type="text"

placeholder="Book Title"

value={formData.title}

onChange={handleChange}

></Form.Control>

</InputGroup>

</Form.Group>

<Form.Group>

<InputGroup>

<InputGroup.Text id="formBookAuthor">Author</InputGroup.Text>

<Form.Control

name="author"

type="text"

placeholder="Book Author"

value={formData.author}

onChange={handleChange}

></Form.Control>

</InputGroup>

</Form.Group>

<Form.Group>

<InputGroup>

<InputGroup.Text id="formBookStatus">Status</InputGroup.Text>

<Form.Select

aria-label="Default select example"

onChange={handleChange}

name="status"

>

<option>Open this select menu</option>

<option value="Available">Available</option>

<option value="Not Available">Not Available</option>

</Form.Select>

</InputGroup>

</Form.Group>

<Button variant="primary" type="submit">

Add

</Button>

</Form>

</div>

);

};

export default AddBook;

**ListBook.jsx**

import React, { useEffect, useState } from "react";

import { bookService } from "../Services/firebase-services";

import "./ListBooks.css";

const BookManagement = () => {

const [books, setBooks] = useState([]);

const [showModal, setShowModal] = useState(false);

const [currentBook, setCurrentBook] = useState({ id: "", title: "", author: "", status: "Available" });

const [newBook, setNewBook] = useState({ title: "", author: "", status: "Available" });

useEffect(() => {

getBooks();

}, []);

const getBooks = async () => {

const data = await bookService.getAllBooks();

const newdata = data.docs.map((doc) => ({ ...doc.data(), id: doc.id }));

setBooks(newdata);

};

const handleDelete = async (id) => {

await bookService.deleteBook(id);

setBooks(books.filter((book) => book.id !== id));

};

const handleUpdateClick = (book) => {

setCurrentBook(book);

setShowModal(true);

};

const handleUpdate = async (e) => {

e.preventDefault();

await bookService.updateBook(currentBook.id, {

title: currentBook.title,

author: currentBook.author,

status: currentBook.status,

});

setShowModal(false);

getBooks();

};

const handleCurrentBookChange = (e) => {

const { name, value } = e.target;

setCurrentBook({ ...currentBook, [name]: value });

};

const handleNewBookChange = (e) => {

const { name, value } = e.target;

setNewBook({ ...newBook, [name]: value });

};

const handleAddBook = async (e) => {

e.preventDefault();

if (!newBook.title || !newBook.author) {

alert("Please fill in all fields");

return;

}

await bookService.addBook(newBook);

setNewBook({ title: "", author: "", status: "Available" });

getBooks();

};

return (

<div className="book-management">

<div className="header">

<h1>Book Management System</h1>

</div>

<div className="container">

<div className="add-book-section">

<h2>Add New Book</h2>

<form onSubmit={handleAddBook} className="add-book-form">

<div className="form-group">

<label htmlFor="title">Title</label>

<input

type="text"

id="title"

name="title"

value={newBook.title}

onChange={handleNewBookChange}

placeholder="Enter book title"

required

/>

</div>

<div className="form-group">

<label htmlFor="author">Author</label>

<input

type="text"

id="author"

name="author"

value={newBook.author}

onChange={handleNewBookChange}

placeholder="Enter author name"

required

/>

</div>

<div className="form-group">

<label htmlFor="status">Status</label>

<select

id="status"

name="status"

value={newBook.status}

onChange={handleNewBookChange}

>

<option value="Available">Available</option>

<option value="Not Available">Not Available</option>

</select>

</div>

<button type="submit" className="btn btn-add">Add Book</button>

</form>

</div>

<div className="book-list-section">

<h2>Book Collection</h2>

<div className="table-container">

<table className="book-table">

<thead>

<tr>

<th>Sr. No.</th>

<th>Title</th>

<th>Author</th>

<th>Status</th>

<th>Actions</th>

</tr>

</thead>

<tbody>

{books.length > 0 ? (

books.map((book, index) => (

<tr key={book.id}>

<td>{index + 1}</td>

<td>{book.title}</td>

<td>{book.author}</td>

<td>

<span className={`status ${book.status === "Available" ? "available" : "not-available"}`}>

{book.status}

</span>

</td>

<td>

<div className="action-buttons">

<button

className="btn btn-update"

onClick={() => handleUpdateClick(book)}

>

Update

</button>

<button

className="btn btn-delete"

onClick={() => handleDelete(book.id)}

>

Delete

</button>

</div>

</td>

</tr>

))

) : (

<tr>

<td colSpan="5" className="no-books">No books available</td>

</tr>

)}

</tbody>

</table>

</div>

</div>

</div>

{/\* Update Modal \*/}

{showModal && (

<div className="modal-overlay">

<div className="modal">

<div className="modal-header">

<h3>Update Book</h3>

<button className="close-btn" onClick={() => setShowModal(false)}>×</button>

</div>

<div className="modal-body">

<form onSubmit={handleUpdate}>

<div className="form-group">

<label htmlFor="update-title">Title</label>

<input

type="text"

id="update-title"

name="title"

value={currentBook.title}

onChange={handleCurrentBookChange}

required

/>

</div>

<div className="form-group">

<label htmlFor="update-author">Author</label>

<input

type="text"

id="update-author"

name="author"

value={currentBook.author}

onChange={handleCurrentBookChange}

required

/>

</div>

<div className="form-group">

<label htmlFor="update-status">Status</label>

<select

id="update-status"

name="status"

value={currentBook.status}

onChange={handleCurrentBookChange}

>

<option value="Available">Available</option>

<option value="Not Available">Not Available</option>

</select>

</div>

<div className="modal-footer">

<button type="button" className="btn btn-cancel" onClick={() => setShowModal(false)}>

Cancel

</button>

<button type="submit" className="btn btn-confirm">

Update

</button>

</div>

</form>

</div>

</div>

</div>

)}

</div>

);

};

export default BookManagement;

**firebaseServices.jsx**

import { db } from "../firebase-config";

import { addDoc, collection, getDocs, doc, deleteDoc, updateDoc } from "firebase/firestore";

const bookCollectionRef = collection(db, "books");

export const bookService = {

getAllBooks: () => {

return getDocs(bookCollectionRef);

},

addBook: (newBook) => {

return addDoc(bookCollectionRef, newBook);

},

deleteBook: async (id) => {

const bookRef = doc(db, "books", id);

await deleteDoc(bookRef);

},

updateBook: async (id, updatedBook) => {

const bookRef = doc(db, "books", id);

await updateDoc(bookRef, updatedBook);

},

};

**firebaseConfig.jsx**

// Import the functions you need from the SDKs you need

import { initializeApp } from "firebase/app";

import { getAnalytics } from "firebase/analytics";

import { getFirestore } from "firebase/firestore";

// TODO: Add SDKs for Firebase products that you want to use

// https://firebase.google.com/docs/web/setup#available-libraries

// Your web app's Firebase configuration

// For Firebase JS SDK v7.20.0 and later, measurementId is optional

const firebaseConfig = {

apiKey: "AIzaSyCIf4cnh9PMEgA53Mi160xk4dmlYpgQH7Q",

authDomain: "mu-test-fire-2025.firebaseapp.com",

projectId: "mu-test-fire-2025",

storageBucket: "mu-test-fire-2025.firebasestorage.app",

messagingSenderId: "331974042670",

appId: "1:331974042670:web:1683e8fc1141394c486bbf",

measurementId: "G-XNFPXBD1YR"

};

// Initialize Firebase

const app = initializeApp(firebaseConfig);

const analytics = getAnalytics(app);

export const db = getFirestore(app);

**OUTPUT**



