**MERN Stack Internal Practical Exam – Set D**

**Duration: 1.5 – 2 Hours Total Marks: 30**

**Task 1 – React Task (15 Marks)**

**Build a “Book Library” App using React.**

**Requirements:**

1. Create a component LibraryApp.
2. Allow the user to:

* Add Book details (Title, Author, Genre).
* Display books in a table format.
* Delete a book entry.

1. Use useState() for state management.

**Code:**

import React, { useState } from "react";

function LibraryApp() {

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

*const* [title, setTitle] = useState("");

*const* [author, setAuthor] = useState("");

*const* [genre, setGenre] = useState("");

*const* addBook = () => {

        if (title.trim() === "") {

            alert("Please Enter Title");

*return*;

        }

        if (author.trim() === "") {

            alert("Please Enter Author");

*return*;

        }

        if (genre.trim() === "") {

            alert("Please Enter Genre");

*return*;

        }

*const* newBook = { id: Date.now(), title, author, genre };

        setBooks([**...**books, newBook]);

        setTitle("");

        setAuthor("");

        setGenre("");

    };

*const* deleteBook = (id) => {

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

    };

*return* (

        <div style={{ padding: "20px", maxWidth: "600px", margin: "0 auto" }}>

            <h2>Book Library</h2>

            <div style={{ marginBottom: "10px" }}>

                <div>

                    Title: <input

                        type="text"

                        placeholder="Title"

                        value={title}

                        onChange={(e) => setTitle(e.target.value)}

                        style={{ marginRight: "10px", }}

                    />

                </div>

                <div>

                    Author: <input

                        type="text"

                        placeholder="Author"

                        value={author}

                        onChange={(e) => setAuthor(e.target.value)}

                        style={{ marginRight: "10px", }}

                    />

                </div>

                <div>

                    Genre: <input

                        type="text"

                        placeholder="Genre"

                        value={genre}

                        onChange={(e) => setGenre(e.target.value)}

                        style={{ marginRight: "10px", }}

                    />

                </div>

                <button onClick={addBook}>Add Book</button>

            </div>

            {books.length > 0 ? (

                <table border="1" cellPadding="8" cellSpacing="0" width="100%">

                    <thead>

                        <tr>

                            <th>Title</th>

                            <th>Author</th>

                            <th>Genre</th>

                            <th>Delete</th>

                        </tr>

                    </thead>

                    <tbody>

                        {books.map(({ id, title, author, genre }) => (

                            <tr key={id}>

                                <td>{title}</td>

                                <td>{author}</td>

                                <td>{genre}</td>

                                <td>

                                    <button onClick={() => deleteBook(id)}>Delete</button>

                                </td>

                            </tr>

                        ))}

                    </tbody>

                </table>

            ) : (

                <p>No books added yet.</p>

            )}

        </div>

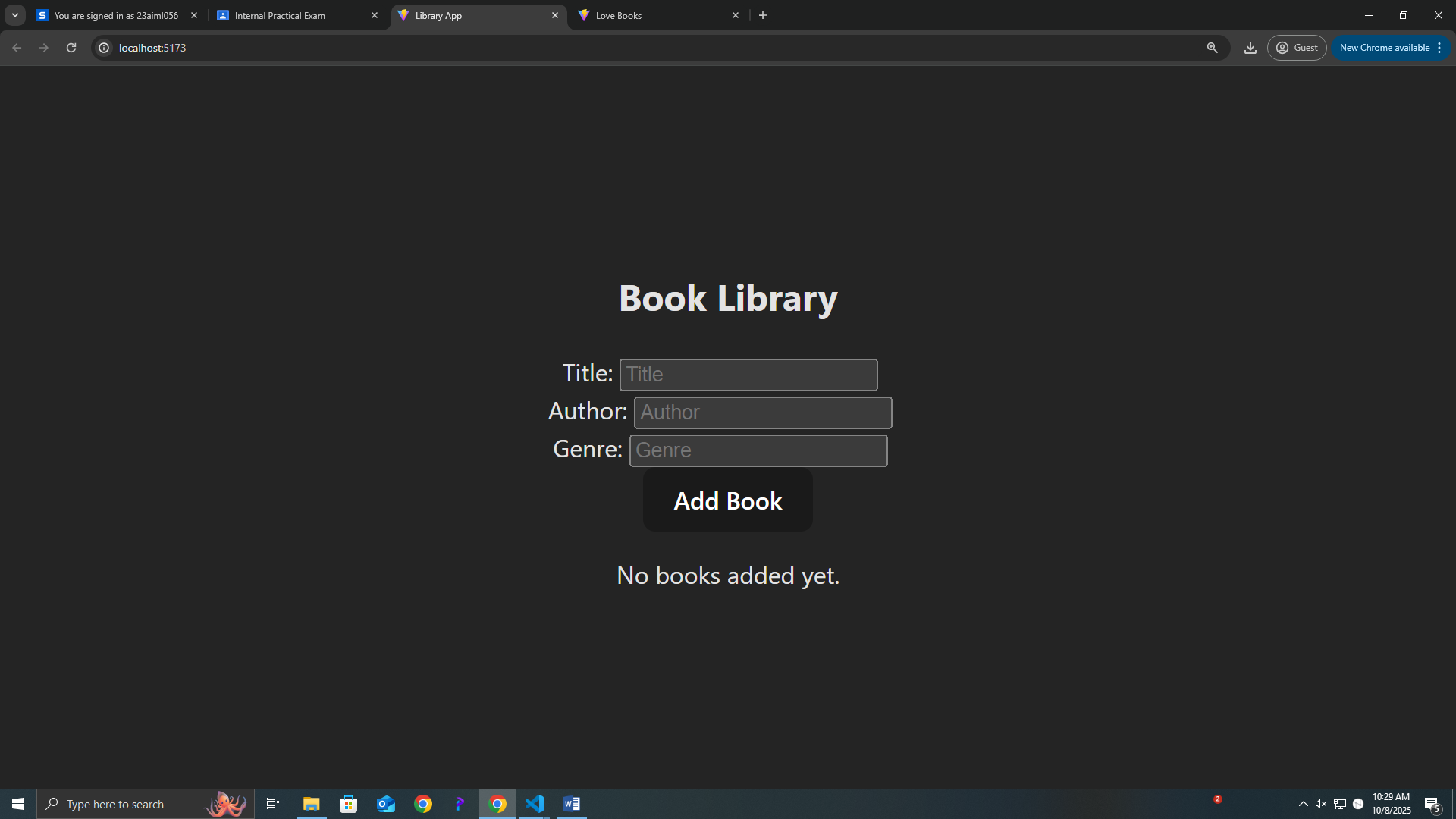
    );

}

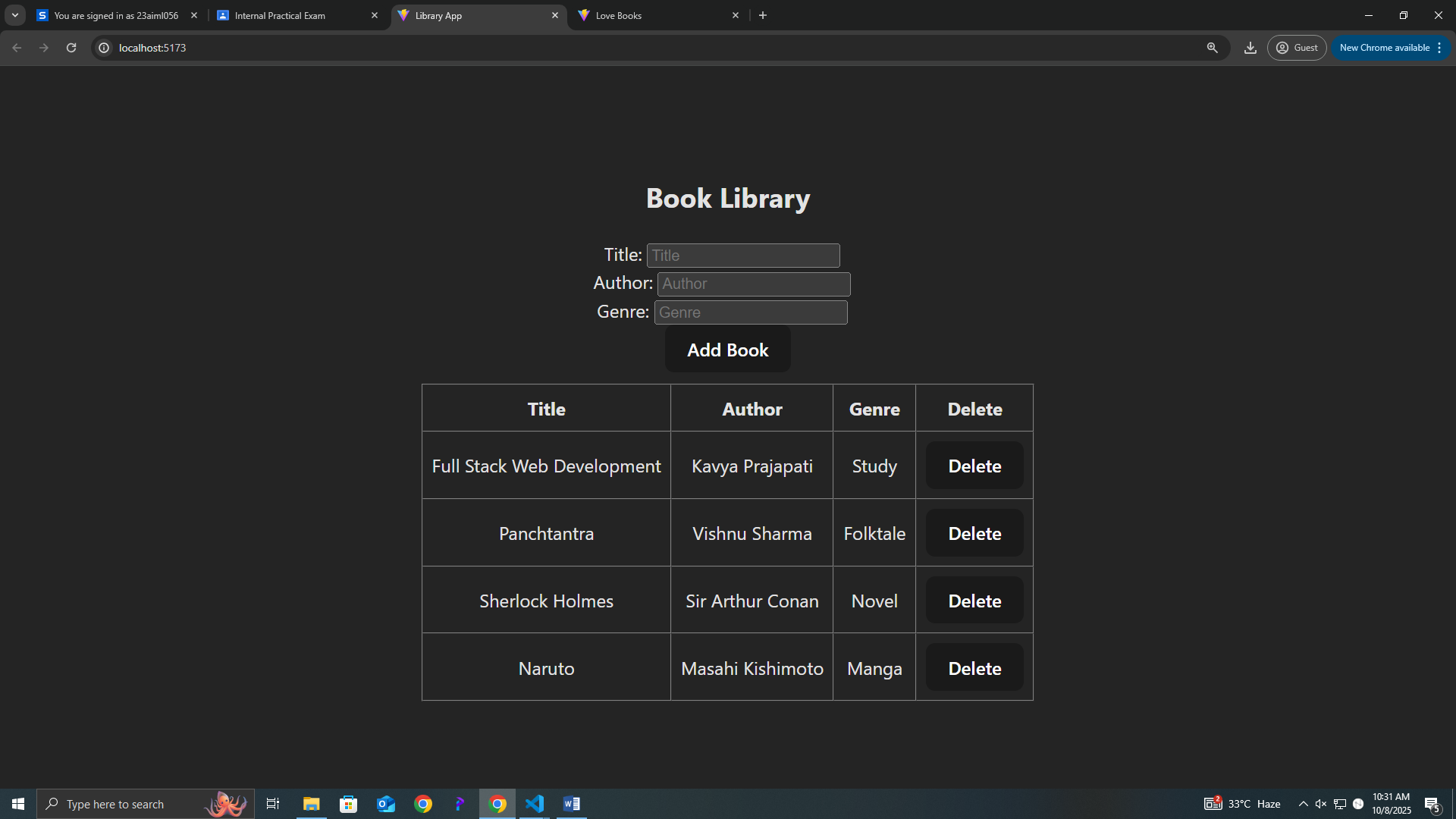
export default LibraryApp;

**Output:**

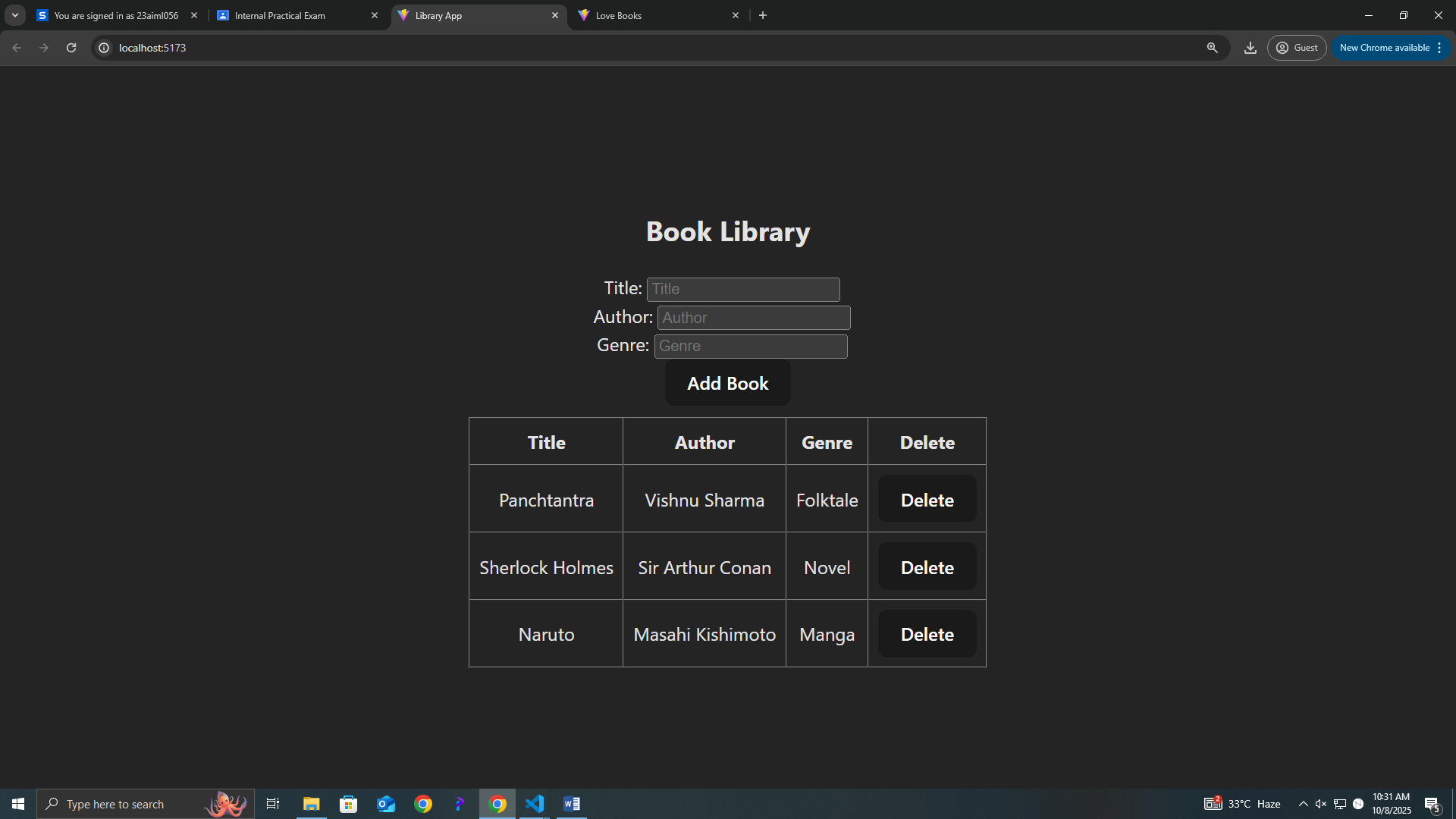
**Initial Page:**



**After Adding some entries:**



**After deleting** “Full Stack Web Development” **Entry:**



**Task 2 (15 Marks) – API Task**

**Fetch and Display Book Data from an API.**

**Requirements:**

1. Use API URL: https://openlibrary.org/subjects/love.json?limit=10
2. Display book details:

* title, author\_name, and first\_publish\_year

1. Use axios or fetch() and handle loading + error states.

**Code:**

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

function LoveBooks() {

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

    useEffect(() => {

        fetch("https://openlibrary.org/subjects/love.json?limit=10")

            .then((res) => {

                if (!res.ok) throw new Error("Network response was not ok");

*return* res.json();

            })

            .then((data) => {

                setBooks(data.works);

                setLoading(false);

            })

            .catch((err) => {

                setError(err.message);

                setLoading(false);

            });

    }, []);

*return* (

        <table>

            <thead>

                <tr>

                    <th>Title</th>

                    <th>Author Name</th>

                    <th>First Publish Year</th>

                </tr>

            </thead>

            <tbody>

                {books.map((book) => (

                    <tr key={book.key}>

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

                        <td>{book.authors && book.authors[0]?.name}</td>

                        <td>{book.first\_publish\_year}</td>

                    </tr>

                ))}

            </tbody>

        </table>

    );

}

export default LoveBooks;

**Output:**

