**FRONTEND SETUP**

**Login.html**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<title>Login</title>

<style>

body {

margin: 0;

height: 100vh;

display: flex;

justify-content: center;

align-items: center;

background: linear-gradient(135deg, #6a11cb, #2575fc);

font-family: 'Poppins', sans-serif;

}

.login-container {

background-color: white;

padding: 40px 35px;

border-radius: 10px;

box-shadow: 0 10px 20px rgba(0, 0, 0, 0.2);

width: 320px;

text-align: center;

}

h2 {

margin-bottom: 25px;

font-weight: 600;

color: #333;

}

input {

width: 100%;

padding: 10px 12px;

margin-bottom: 15px;

border: 1px solid #ddd;

border-radius: 5px;

outline: none;

font-size: 14px;

background-color: #fffbe8;

}

input:focus {

border-color: #2575fc;

}

button {

width: 100%;

background-color: #2575fc;

color: white;

border: none;

padding: 10px;

font-size: 15px;

border-radius: 5px;

cursor: pointer;

transition: 0.3s;

}

button:hover {

background-color: #1b5fda;

}

p {

margin-top: 15px;

font-size: 13px;

color: #555;

}

a {

color: #2575fc;

text-decoration: none;

font-weight: 500;

}

a:hover {

text-decoration: underline;

}

</style>

</head>

<body>

<div class="login-container">

<h2>Login</h2>

<input type="email" id="email" placeholder="Email" required />

<input type="password" id="password" placeholder="Password" required />

<button onclick="login()">Login</button>

<p>Don't have an account? <a href="#">Sign Up</a></p>

</div>

<script>

function login() {

const email = document.getElementById("email").value.trim();

const password = document.getElementById("password").value.trim();

if (email === "" || password === "") {

alert("Please enter both email and password.");

return;

}

// Dummy check (for frontend only)

if (email && password) {

alert(`Login successful!\nEmail: ${email}`);

}

}

</script>

<script>

Async function login() {

Const username = document.getElementById(“email”).value.trim();

Const password = document.getElementById(“password”).value.trim();

If (!username || !password) {

Alert(“Please fill all fields!”);

Return;

}

Const res = await fetch(<http://localhost:5000/login>, {

Method: “POST”,

Headers: { “Content-Type”: “application/json” },

Body: JSON.stringify({ username, password })

});

Const data = await res.json();

If (res.ok) {

Alert(“Login successful!”);

localStorage.setItem(“token”, data.token);

window.location.href = “dashboard.html”;

} else {

Alert(data.message || “Login failed!”);

}

}

</script>

</body>

</html>

**Dashboard.html**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<title>Library Dashboard</title>

<style>

body {

margin: 0;

font-family: 'Poppins', sans-serif;

display: flex;

height: 100vh;

background: linear-gradient(135deg, #6a11cb, #2575fc);

color: #333;

}

.sidebar {

width: 200px;

background: #243447;

color: white;

padding: 20px;

display: flex;

flex-direction: column;

}

.sidebar h2 {

text-align: center;

margin-bottom: 30px;

}

.sidebar a {

color: white;

text-decoration: none;

margin: 10px 0;

display: block;

padding: 10px;

border-radius: 5px;

transition: 0.3s;

}

.sidebar a:hover {

background: #3b556d;

}

.main {

flex: 1;

padding: 30px;

overflow-y: auto;

}

h2 {

color: white;

margin-bottom: 20px;

}

.cards {

display: grid;

grid-template-columns: repeat(auto-fit, minmax(200px, 1fr));

gap: 20px;

margin-bottom: 30px;

}

.card {

background: white;

border-radius: 8px;

padding: 20px;

text-align: center;

box-shadow: 0 5px 15px rgba(0,0,0,0.1);

}

table {

width: 100%;

border-collapse: collapse;

background: white;

border-radius: 8px;

overflow: hidden;

}

th, td {

padding: 12px;

text-align: left;

border-bottom: 1px solid #ddd;

}

th {

background: #243447;

color: white;

}

</style>

</head>

<body>

<div class="sidebar">

<h2>Library</h2>

<a href="#">Home</a>

<a href="#">Books</a>

<a href="addbook.html">Add Book</a>

<a href="#">Members</a>

<a href="#">Issue/Return</a>

<a href="#">Logout</a>

</div>

<div class="main">

<h2>Dashboard</h2>

<div class="cards">

<div class="card">

<h3>Total Books</h3>

<p>120</p>

</div>

<div class="card">

<h3>Books Issued</h3>

<p>45</p>

</div>

<div class="card">

<h3>Available Books</h3>

<p>75</p>

</div>

<div class="card">

<h3>Total Members</h3>

<p>50</p>

</div>

</div>

<h2>Recent Books</h2>

<table>

<tr>

<th>Book ID</th>

<th>Title</th>

<th>Author</th>

<th>Status</th>

</tr>

<tr>

<td>001</td>

<td>JavaScript Basics</td>

<td>John Doe</td>

<td>Available</td>

</tr>

<tr>

<td>002</td>

<td>Python Guide</td>

<td>Jane Smith</td>

<td>Issued</td>

</tr>

</table>

</div>

<script>

Async function loadDashboard() {

Const res = await fetch(<http://localhost:5000/dashboard>);

Const data = await res.json();

Document.getElementById(“totalBooks”).textContent = data.totalBooks;

Document.getElementById(“booksIssued”).textContent = data.booksIssued;

Document.getElementById(“availableBooks”).textContent = data.availableBooks;

Document.getElementById(“totalMembers”).textContent = data.totalMembers;

}

Async function loadRecentBooks() {

Const res = await fetch(<http://localhost:5000/books>);

Const books = await res.json();

Const table = document.getElementById(“recentBooks”);

Table.innerHTML = books.map(b => `

<tr>

<td>${b.book\_id}</td>

<td>${b.title}</td>

<td>${b.author}</td>

<td>${b.status}</td>

</tr>`).join(‘’);

}

loadDashboard();

loadRecentBooks();

</script>

</body>

</html>

**Addbook.html**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<title>Add New Book</title>

<style>

body {

margin: 0;

font-family: 'Poppins', sans-serif;

display: flex;

height: 100vh;

background: linear-gradient(135deg, #6a11cb, #2575fc);

color: #333;

}

.sidebar {

width: 200px;

background: #243447;

color: white;

padding: 20px;

display: flex;

flex-direction: column;

}

.sidebar h2 {

text-align: center;

margin-bottom: 30px;

}

.sidebar a {

color: white;

text-decoration: none;

margin: 10px 0;

display: block;

padding: 10px;

border-radius: 5px;

transition: 0.3s;

}

.sidebar a:hover {

background: #3b556d;

}

.main {

flex: 1;

padding: 30px;

overflow-y: auto;

}

h2 {

color: white;

margin-bottom: 20px;

}

.form-box {

background: white;

padding: 25px;

border-radius: 8px;

box-shadow: 0 5px 15px rgba(0,0,0,0.1);

width: 350px;

}

label {

display: block;

margin-bottom: 5px;

font-weight: 500;

}

input, select {

width: 100%;

padding: 8px 10px;

margin-bottom: 15px;

border: 1px solid #ccc;

border-radius: 5px;

outline: none;

}

button {

background: #243447;

color: white;

border: none;

padding: 10px;

border-radius: 5px;

width: 100%;

cursor: pointer;

transition: 0.3s;

}

button:hover {

background: #1c293a;

}

</style>

</head>

<body>

<div class="sidebar">

<h2>Library</h2>

<a href="dashboard.html">Home</a>

<a href="#">Books</a>

<a href="addbook.html">Add Book</a>

<a href="#">Members</a>

<a href="#">Issue/Return</a>

<a href="#">Logout</a>

</div>

<div class="main">

<h2>Add New Book</h2>

<div class="form-box">

<label>Book ID:</label>

<input type="text" id="bookId" placeholder="Enter Book ID">

<label>Title:</label>

<input type="text" id="title" placeholder="Enter Book Title">

<label>Author:</label>

<input type="text" id="author" placeholder="Enter Author Name">

<label>Status:</label>

<select id="status">

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

<option value="Issued">Issued</option>

</select>

<button onclick="addBook()">Add Book</button>

</div>

</div>

<script>

function addBook() {

const id = document.getElementById('bookId').value.trim();

const title = document.getElementById('title').value.trim();

const author = document.getElementById('author').value.trim();

const status = document.getElementById('status').value;

if (!id || !title || !author) {

alert('Please fill all fields.');

return;

}

alert(`Book Added!\n\nBook ID: ${id}\nTitle: ${title}\nAuthor: ${author}\nStatus: ${status}`);

}

</script>

<script>

Async function addBook() {

Const book\_id = document.getElementById(“bookId”).value.trim();

Const title = document.getElementById(“title”).value.trim();

Const author = document.getElementById(“author”).value.trim();

If (!book\_id || !title || !author) {

Alert(“Please fill all fields!”);

Return;

}

Const res = await fetch(<http://localhost:5000/add-book>, {

Method: “POST”,

Headers: { “Content-Type”: “application/json” },

Body: JSON.stringify({ book\_id, title, author })

});

Const data = await res.json();

Alert(data.message);

}

</script>

</body>

</html>

**Issues / Return book.html**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<title>Library Book Management - Issue & Return</title>

<style>

body {

font-family: "Poppins", sans-serif;

background: #f5f7fa;

margin: 0;

padding: 0;

}

header {

background: linear-gradient(90deg, #0052cc, #007bff);

color: white;

text-align: center;

padding: 15px 0;

font-size: 22px;

font-weight: 600;

}

.container {

display: flex;

justify-content: space-around;

flex-wrap: wrap;

margin: 30px auto;

max-width: 900px;

gap: 20px;

}

.card {

background: #fff;

box-shadow: 0 3px 10px rgba(0, 0, 0, 0.1);

border-radius: 10px;

padding: 20px;

width: 320px;

}

h3 {

text-align: center;

margin-bottom: 15px;

color: #003366;

}

label {

display: block;

margin: 8px 0 5px;

font-weight: 500;

}

input {

width: 100%;

padding: 8px;

border-radius: 6px;

border: 1px solid #ccc;

}

button {

width: 100%;

background: #007bff;

border: none;

color: white;

padding: 10px;

margin-top: 10px;

border-radius: 6px;

font-size: 16px;

cursor: pointer;

transition: 0.3s;

}

button:hover {

background: #0056b3;

}

table {

width: 90%;

margin: 20px auto;

border-collapse: collapse;

background: white;

border-radius: 10px;

overflow: hidden;

box-shadow: 0 3px 10px rgba(0, 0, 0, 0.1);

}

th, td {

padding: 12px;

text-align: center;

border-bottom: 1px solid #ddd;

}

th {

background: #0052cc;

color: white;

}

.status-returned {

color: green;

font-weight: 600;

}

.status-issued {

color: orange;

font-weight: 600;

}

</style>

</head>

<body>

<header>Library Book Management - Issue & Return</header>

<div class="container">

<!-- Issue Book -->

<div class="card">

<h3>Issue Book</h3>

<label>Student Name</label>

<input type="text" id="studentName" placeholder="Enter student name">

<label>Book ID</label>

<input type="text" id="bookId" placeholder="Enter book ID">

<label>Issue Date</label>

<input type="date" id="issueDate">

<button onclick="issueBook()">Issue Book</button>

</div>

<!-- Return Book -->

<div class="card">

<h3>Return Book</h3>

<label>Book ID</label>

<input type="text" id="returnBookId" placeholder="Enter book ID">

<label>Return Date</label>

<input type="date" id="returnDate">

<button onclick="returnBook()">Return Book</button>

</div>

</div>

<section>

<h3 style="text-align:center; margin-top:40px;">Library Book Report</h3>

<table id="reportTable">

<thead>

<tr>

<th>Book ID</th>

<th>Book Name</th>

<th>Student Name</th>

<th>Issue Date</th>

<th>Return Date</th>

<th>Status</th>

</tr>

</thead>

<tbody id="reportBody">

<tr>

<td>B001</td>

<td>Introduction to Python</td>

<td>John Doe</td>

<td>2025-10-20</td>

<td>2025-10-23</td>

<td class="status-returned">Returned</td>

</tr>

<tr>

<td>B002</td>

<td>HTML & CSS Basics</td>

<td>Jane Smith</td>

<td>2025-10-21</td>

<td>-</td>

<td class="status-issued">Issued</td>

</tr>

</tbody>

</table>

</section>

<script>

const reportBody = document.getElementById('reportBody');

function issueBook() {

const name = document.getElementById('studentName').value;

const id = document.getElementById('bookId').value;

const date = document.getElementById('issueDate').value;

if (!name || !id || !date) return alert("Please fill all fields.");

const row = document.createElement('tr');

row.innerHTML = `

<td>${id}</td>

<td>Book Title</td>

<td>${name}</td>

<td>${date}</td>

<td>-</td>

<td class="status-issued">Issued</td>

`;

reportBody.appendChild(row);

alert("Book issued successfully!");

}

function returnBook() {

const id = document.getElementById('returnBookId').value;

const date = document.getElementById('returnDate').value;

if (!id || !date) return alert("Please fill all fields.");

const rows = reportBody.querySelectorAll('tr');

let found = false;

rows.forEach(row => {

if (row.children[0].textContent === id && row.children[5].textContent === "Issued") {

row.children[4].textContent = date;

row.children[5].textContent = "Returned";

row.children[5].className = "status-returned";

found = true;

}

});

alert(found ? "Book returned successfully!" : "Book ID not found or already returned.");

}

</script>

<script>

Async function issueBook() {

Const student\_name = document.getElementById(“studentName”).value.trim();

Const book\_id = document.getElementById(“issueBookId”).value.trim();

Const issue\_date = document.getElementById(“issueDate”).value;

If (!student\_name || !book\_id || !issue\_date) {

Alert(“Please fill all fields!”);

Return;

}

Const res = await fetch(<http://localhost:5000/issue>, {

Method: “POST”,

Headers: { “Content-Type”: “application/json” },

Body: JSON.stringify({ student\_name, book\_id, issue\_date })

});

Const data = await res.json();

Alert(data.message);

}

Async function returnBook() {

Const book\_id = document.getElementById(“returnBookId”).value.trim();

Const return\_date = document.getElementById(“returnDate”).value;

If (!book\_id || !return\_date) {

Alert(“Please fill all fields!”);

Return;

}

Const res = await fetch(<http://localhost:5000/return>, {

Method: “POST”,

Headers: { “Content-Type”: “application/json” },

Body: JSON.stringify({ book\_id, return\_date })

});

Const data = await res.json();

Alert(data.message);

}

</script>

</body>

</html>

**BACKEND SETUP**

**Initialise node.js**

Npm init -y

Npm install express mysql2 cors body-parser bcryptjs jsonwebtoken

**Create db.js**

// backend/db.js

Const mysql = require(‘mysql2’);

Const db = mysql.createConnection({

Host: ‘localhost’,

User: ‘root’, // change if needed

Password: ‘’, // add your MySQL password if any

Database: ‘library\_db’

});

Db.connect(err => {

If (err) throw err;

Console.log(‘✅ MySQL Connected’);

});

Module.exports = db;

**Create database**

**(run in MySQL terminal)**

CREATE DATABASE library\_db;

USE library\_db;

CREATE TABLE admins (

id INT AUTO\_INCREMENT PRIMARY KEY,

username VARCHAR(50),

password VARCHAR(255)

);

INSERT INTO admins (username, password) VALUES ('admin', '12345');

CREATE TABLE books (

id INT AUTO\_INCREMENT PRIMARY KEY,

book\_id VARCHAR(20),

title VARCHAR(100),

author VARCHAR(100),

status VARCHAR(20) DEFAULT 'Available'

);

CREATE TABLE members (

id INT AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(100),

email VARCHAR(100)

);

CREATE TABLE issue\_return (

id INT AUTO\_INCREMENT PRIMARY KEY,

book\_id VARCHAR(20),

student\_name VARCHAR(100),

issue\_date DATE,

return\_date DATE,

status VARCHAR(20)

);

**Create server.js**

// backend/server.js

Const express = require(‘express’);

Const cors = require(‘cors’);

Const bodyParser = require(‘body-parser’);

Const db = require(‘./db’);

Const app = express();

App.use(cors());

App.use(bodyParser.json());

// ================= LOGIN =================

App.post(‘/login’, (req, res) => {

Const { username, password } = req.body;

Db.query(‘SELECT \* FROM admins WHERE username=?’, [username], (err, results) => {

If (err) return res.status(500).send(err);

If (results.length === 0) return res.status(401).send({ message: ‘User not found’ });

Const admin = results[0];

If (admin.password === password) {

Res.send({ message: ‘Login successful’ });

} else {

Res.status(401).send({ message: ‘Invalid password’ });

}

});

});

// ================= DASHBOARD =================

App.get(‘/dashboard’, (req, res) => {

Const sql = `

SELECT

(SELECT COUNT(\*) FROM books) AS totalBooks,

(SELECT COUNT(\*) FROM issue\_return WHERE status=’Issued’) AS booksIssued,

(SELECT COUNT(\*) FROM books WHERE status=’Available’) AS availableBooks,

(SELECT COUNT(\*) FROM members) AS totalMembers

`;

Db.query(sql, (err, result) => {

If (err) return res.status(500).send(err);

Res.send(result[0]);

});

});

// ================= ADD BOOK =================

App.post(‘/add-book’, (req, res) => {

Const { book\_id, title, author } = req.body;

Db.query(‘INSERT INTO books (book\_id, title, author) VALUES (?, ?, ?)’,

[book\_id, title, author],

(err) => {

If (err) return res.status(500).send(err);

Res.send({ message: ‘Book added successfully’ });

}

);

});

// ================= BOOK LIST =================

App.get(‘/books’, (req, res) => {

Db.query(‘SELECT \* FROM books’, (err, results) => {

If (err) return res.status(500).send(err);

Res.send(results);

});

});

// ================= ISSUE BOOK =================

App.post(‘/issue’, (req, res) => {

Const { book\_id, student\_name, issue\_date } = req.body;

Db.query(

‘INSERT INTO issue\_return (book\_id, student\_name, issue\_date, status) VALUES (?, ?, ?, ?)’,

[book\_id, student\_name, issue\_date, ‘Issued’],

(err) => {

If (err) return res.status(500).send(err);

Db.query(‘UPDATE books SET status=”Issued” WHERE book\_id=?’, [book\_id]);

Res.send({ message: ‘Book issued successfully’ });

}

);

});

// ================= RETURN BOOK =================

App.post(‘/return’, (req, res) => {

Const { book\_id, return\_date } = req.body;

Db.query(

‘UPDATE issue\_return SET return\_date=?, status=”Returned” WHERE book\_id=? AND status=”Issued”’,

[return\_date, book\_id],

(err) => {

If (err) return res.status(500).send(err);

Db.query(‘UPDATE books SET status=”Available” WHERE book\_id=?’, [book\_id]);

Res.send({ message: ‘Book returned successfully’ });

}

);

});

// ================= REPORT =================

App.get(‘/report’, (req, res) => {

Db.query(‘SELECT \* FROM issue\_return’, (err, results) => {

If (err) return res.status(500).send(err);

Res.send(results);

});

});

App.listen(5000, () => console.log(‘🚀 Server running on <http://localhost:5000’)>);

**Run the system**

cd backend

node server.js

**You’ll see ;**

✅ MySQL Connected

🚀 Server running on http://localhost:5000