

Library Management Web App

Introduction

A website-based application called the Library Management Web Application was created to automate the manual procedures needed to keep track of a library's inventory. Paper registers or simple spreadsheets, which are hard to search and prone to damage, are still used to manage book records in many schools and small businesses. The goal of this project is to replace those antiquated techniques with a cutting-edge, dependable, and user-friendly "Full-Stack" web application.

A librarian can manage the entire collection from any computer linked to the local server thanks to this system's web-based methodology. The project's main goal is to deliver a clear visual experience while guaranteeing the stability and security of the "behind-the-scenes" database logic.

Project Scope and Objectives

The primary goal of this school project is to demonstrate how different programming languages work together to create a functional system. The objectives include:

- Developing a user-friendly interface using HTML and CSS.
- Implementing JavaScript for client-side validation to ensure data is entered correctly.
- Using PHP as a server-side language to process data and manage the application's logic.
- Utilizing MySQL as a backend database to store book information permanently.

Understanding the CRUD System

The core of this project is the implementation of CRUD operations. This is a standard requirement for computer science projects as it covers the four vital ways a user interacts with a database:

1. Create: Using a web form to add new books (Title, Author, ISBN) into the system.
2. Read: Fetching those records from the database and displaying them in a clean, organized table on the home page.

3. Update: Allowing the librarian to change a book's information (like correcting a misspelled author name).
4. Delete: Giving the option to remove a book from the list if it is no longer available.

System Architecture (The "Full-Stack" Approach)

This project follows a Three-Tier Architecture, which is how real-world websites like Amazon or Facebook are built:

- Frontend (The Look): Built with HTML and CSS (Bootstrap). This is what the librarian interacts with.
- The Backend (The Brain): Written in PHP. It acts as the middleman, taking instructions from the user and talking to the database.
- The Database (The Storage): Powered by MySQL. It acts as the digital "filing cabinet" that holds all the book data even after the computer is turned off.

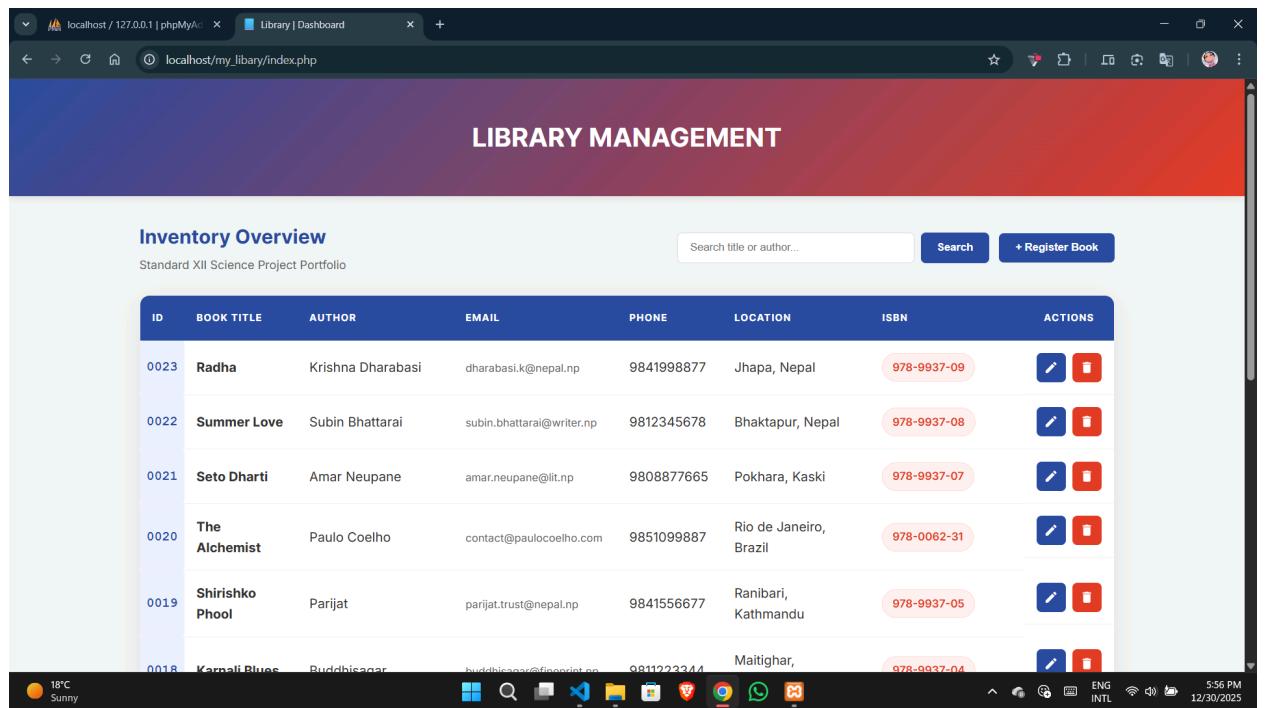
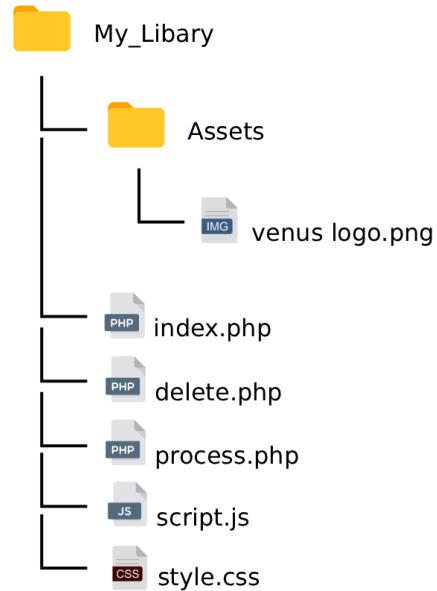


Fig. CURD Webapp

FILE STRUCTURE

The **Library Management System** follows a modular directory structure to ensure a clean separation between the User Interface (Frontend), the Application Logic (Backend), and the Data Storage (Database). This organization enhances maintainability and scalability.



Design Philosophy

The file structure is designed to adhere to the **Separation of Concerns (SoC)** principle. By keeping the CSS and JavaScript in separate files, the application loads faster and the source code remains readable. The use of an assets folder ensures that all media files are stored centrally, preventing root-directory clutter.

DATABASE SETUP AND CONFIGURATION

The **Library Management System** utilizes a relational database management system (RDBMS) to store book records securely. The environment is managed using **XAMPP**, which provides the Apache web server and the MySQL database engine.

Environment Configuration

To initialize the system, the following environment settings are configured in the XAMPP Control Panel:

- **Server:** Apache (Port 80/443)
- **Database:** MySQL (Port 3306)
- **Administration Tool:** phpMyAdmin

Database Schema Initialization

A database named library_management was created with the utf8mb4_general_ci collation to support a wide range of characters. The core data is stored in the books table, which was generated using the following SQL Command:

```
1 -- 1. Create the database
2 CREATE DATABASE IF NOT EXISTS library_management;
3 USE library_management;
4 -- 2. Create the table with a 4-digit formatted ID
5 CREATE TABLE books (
6     id INT(4) UNSIGNED ZEROFILL NOT NULL AUTO_INCREMENT,
7     title VARCHAR(255) NOT NULL,
8     author VARCHAR(100) NOT NULL,
9     email VARCHAR(200) NOT NULL,
10    phone VARCHAR(20) NULL,
11    address VARCHAR(255) NULL,
12    isbn VARCHAR(50) NOT NULL UNIQUE,
13    created_at DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,
14    PRIMARY KEY (id)
15 );
16
17 -- 3. Insert Nepali sample data
18 INSERT INTO books (title, author, email, phone, address, isbn)
19 VALUES
20 ('Muna Madan', 'Laxmi Prasad Devkota', 'info@sajhaprakashan.com.np', '9841234567', 'Pulchowk, Lalitpur', '978-9937-1'),
21 ('Seto Bagh', 'Diamond Shumsher Rana', 'diamond.shumsher@nepal.com', '9851012345', 'Thamel, Kathmandu', '978-9937-2'),
22 ('Palpasa Café', 'Narayan Wagle', 'narayan.wagle@kantipur.com', '9803334444', 'Baneshwor, Kathmandu', '978-9937-3'),
23 ('Karnali Blues', 'Buddhisagar', 'buddhisagar@writer.com.np', '9812223334', 'Maitighar, Kathmandu', '978-9937-4');
```

Data Dictionary

The table structure is defined as follows to ensure data integrity and optimized indexing:

The screenshot shows the phpMyAdmin interface for the 'library_management' database. The 'books' table is selected. The table structure includes columns: id, title, author, email, phone, address, isbn, and created_at. The data grid displays 25 rows of book entries. At the bottom, there are buttons for 'Check all', 'With selected:', and 'Export'. Below the table, there's a 'Query results operations' section with options like 'Console', 'Copy to clipboard', 'Export', 'Display chart', and 'Create view'. The system status bar at the bottom shows '18°C Sunny' and the date/time '12/30/2025 5:50 PM'.

	Edit	Copy	Delete	0015	Muna Madan	Laxmi Prasad Devkota	devkota.info@nepal.gov.np	9841223344	Dilli Bazar, Kathmandu	978-9937-01	2025-12-30 13:31:26
<input type="checkbox"/>	Edit	Copy	Delete	0016	Seto Bagh	Diamond Shumsher Rana	diamond.rana@history.np	9851011223	Thamel, Kathmandu	978-9937-02	2025-12-30 13:31:26
<input type="checkbox"/>	Edit	Copy	Delete	0017	Palpasa Cafe	Narayan Wagle	narayan.wagle@media.np	9803344556	Baneshwor, Kathmandu	978-9937-03	2025-12-30 13:31:26
<input type="checkbox"/>	Edit	Copy	Delete	0018	Karnali Blues	Buddhisagar	buddhisagar@fineprint.np	9811223344	Matighar, Kathmandu	978-9937-04	2025-12-30 13:31:26
<input type="checkbox"/>	Edit	Copy	Delete	0019	Shristik Phool	Parjat	parjat.trust@nepal.np	9841556677	Rainibari, Kathmandu	978-9937-05	2025-12-30 13:31:26
<input type="checkbox"/>	Edit	Copy	Delete	0020	The Alchemist	Paulo Coelho	contact@paulocoelho.com	9851098887	Rio de Janeiro, Brazil	978-0062-31	2025-12-30 13:31:26
<input type="checkbox"/>	Edit	Copy	Delete	0021	Seto Dharti	Amar Neupane	amar.neupane@lit.np	9808877665	Pokhara, Kaski	978-9937-07	2025-12-30 13:31:26
<input type="checkbox"/>	Edit	Copy	Delete	0022	Summer Love	Subin Bhattarai	subin.bhattarai@writer.np	9812345678	Bhaktapur, Nepal	978-9937-08	2025-12-30 13:31:26
<input type="checkbox"/>	Edit	Copy	Delete	0023	Radha	Krishna Dharabasi	dharabasi.k@nepal.np	9841998877	Jhapa, Nepal	978-9937-09	2025-12-30 13:31:26

Connection Logic (PHP to MySQL)

The bridge between the frontend and the database is established using the mysqli object-oriented library in PHP. The connection parameters are defined as:

- **Hostname:** localhost
- **Username:** root
- **Password:** “ ”(None)
- **Database:** libary_management

This setup ensures that all CRUD operations performed on the web interface are reflected in the MySQL database in real-time.

```
54 <?php
55 $conn = new mysqli('localhost', 'root', '', 'libary_management');
56 if ($conn->connect_error) die("Database connection failed");
57
```

SOURCE CODE DOCUMENTATION

The following snippets represent the final implementation of the **Library Management System**. The code is organized into Frontend (UI), Styling (CSS), Client-side Logic (JS), and Backend Operations (PHP).

1. index.php (The Interface)

This is the main file that builds the dashboard, handles the search logic, and displays the book inventory.

```
<!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>
    <link rel="icon" href="data:image/svg+xml,<svg xmlns=%22http://www.w3.org/2000/svg%22 viewBox=%220 0 100 100%22><text y=%22.9em%22 font-size=%2290%22>■</text></svg>">
    <link rel="stylesheet" href="style.css">
</head>
<body>

    <header class="header">
        <h1>LIBRARY MANAGEMENT</h1>
    </header>

    <div class="container">
        <div class="top-bar">
            <div>
                <h2 style="margin:0; color: #2b4ca1;">Inventory Overview</h2>
                <p style="margin: 5px 0 0; color: #666; font-size: 0.9rem;">Standard XII Science Project Portfolio</p>
            </div>
            <div style="display: flex; gap: 12px; align-items: center; flex-wrap: wrap;">
                <form action="" method="GET" style="display: flex; gap: 8px;">
                    <div class="search-container">
                        <input type="text" name="search" id="searchInput" placeholder="Search title or author..." value="<?php echo isset($_GET['search']) ? htmlspecialchars($_GET['search'], ENT_QUOTES) : ''; ?>">
                        <?php if(!empty($_GET['search'])): ?>
                            <a href="index.php" class="btn-clear" title="Clear Search">X</a>
                        <?php endif; ?>
                    </div>
                    <button type="submit" class="btn-search">Search</button>
                </form>
                <button onclick="openModal('addModal')" class="btn-add">+ Register Book</button>
            </div>
        </div>

        <div class="table-card">
            <table>
                <thead>
                    <tr>
                        <th>ID</th>
                        <th>Book Title</th>
                        <th>Author</th>
                        <th>Email</th>
                        <th>Phone</th>
                        <th>Location</th>
                    </tr>
                </thead>
                <tbody>
                    <tr>
                        <td>1</td>
                        <td>The Alchemist</td>
                        <td>Paulo Coelho</td>
                        <td>example@example.com</td>
                        <td>9876543210</td>
                        <td>New York</td>
                    </tr>
                    <tr>
                        <td>2</td>
                        <td>To Kill a Mockingbird</td>
                        <td>Harper Lee</td>
                        <td>example2@example.com</td>
                        <td>9876543211</td>
                        <td>Los Angeles</td>
                    </tr>
                    <tr>
                        <td>3</td>
                        <td>The Great Gatsby</td>
                        <td>F. Scott Fitzgerald</td>
                        <td>example3@example.com</td>
                        <td>9876543212</td>
                        <td>Chicago</td>
                    </tr>
                </tbody>
            </table>
        </div>
    </div>
</body>
```

```

        <th>ISBN</th>
        <th style="text-align:center;">Actions</th>
    </tr>
</thead>
<tbody>
    <?php
    $conn = new mysqli('localhost', 'root', '', 'library_management');
    if ($conn->connect_error) die("Database connection failed");

    $search = isset($_GET['search']) ? $conn->real_escape_string($_GET['search']) : '';
    $sql = "SELECT * FROM books " . ($search ? "WHERE title LIKE '%$search%' OR author LIKE '%$search%' " : "") . "ORDER BY id DESC";
    $result = $conn->query($sql);

    if ($result && $result->num_rows > 0) {
        while($row = $result->fetch_assoc()) {
            // Prepare data for JS
            $rowData = htmlspecialchars(json_encode($row), ENT_QUOTES, 'UTF-8');
        ?>
        <tr>
            <td class='id-text'><?php echo $row["id"]; ?></td>
            <td><strong><?php echo htmlspecialchars($row["title"]); ?></strong></td>
            <td><?php echo htmlspecialchars($row["author"]); ?></td>
            <td><small style="color: #666;"><?php echo htmlspecialchars($row["email"]); ?></small>
        </td>
            <td><?php echo htmlspecialchars($row["phone"]); ?></td>
            <td><?php echo htmlspecialchars($row["address"]); ?></td>
            <td><span class='badge-isbn'><?php echo htmlspecialchars($row["isbn"]); ?></span></td>
            <td class='action-group'>
                <button onclick='handleEdit(<?php echo $rowData; ?>)' class='btn-icon btn-edit-icon' title='Edit'>
                    <svg viewBox='0 0 24 24'><path d='M3 17.25V2l3.75L17.81 9.94l-3.75-3.75L3.75 17.25zM20.71 7.04c.39-.39-1.02 0-1.41l-2.34c-.39-.39-1.02-.39-1.41 0l-1.83 1.83 3.75 3.75 1.83-1.83z' /></svg>
                </button>
                <a href='delete.php?id=<?php echo $row["id"]; ?>' class='btn-icon btn-delete-icon' title='Delete' onclick='return confirm("Delete this book permanently?")'>
                    <svg viewBox='0 0 24 24'><path d='M6 19c0 1.1.9 2.2h8c1.1 0 2-.9 2-2V7H6v12zM19 4h-3.5l-1-1h-5l-1 1H5v2h14V4z' /></svg>
                </a>
            </td>
        </tr>
        <?php
    }
    } else {
        echo "<tr><td colspan='8' style='text-align:center; padding:60px; color: #999;'>
            <div style='font-size: 3rem;'>■</div>
            <p>No books found.</p>
        </td></tr>";
    }
    ?
</tbody>
</table>
</div>

<div id="addModal" class="modal">
    <div class="modal-content">
        <span class="close" onclick="closeModal('addModal')">&times;</span>
        <h2 style="color:#2b4cal;">Register New Book</h2>
        <form action="process.php?action=add" method="POST">
            <div class="form-group"><label>Title</label><input type="text" name="title" required></div>
            <div class="form-group"><label>Author</label><input type="text" name="author" required></div>
            <div class="form-group"><label>Email</label><input type="email" name="email" required></div>
            <div class="form-group"><label>Phone</label><input type="text" name="phone"></div>
            <div class="form-group"><label>Location</label><input type="text" name="address"></div>
            <div class="form-group"><label>ISBN</label><input type="text" name="isbn" required></div>
            <button type="submit" class="btn-search" style="width:100%;">Save Book</button>
        </form>
    </div>
</div>

<div id="editModal" class="modal">
    <div class="modal-content">
        <span class="close" onclick="closeModal('editModal')">&times;</span>
        <h2 style="color:#e53e27;">Update Details</h2>
        <form action="process.php?action=edit" method="POST">
            <input type="hidden" name="id" id="edit_id">
            <div class="form-group"><label>Title</label><input type="text" name="title" id="edit_title">

```

```

    required></div>
    <div class="form-group"><label>Author</label><input type="text" name="author" id="edit_author" required></div>
    <div class="form-group"><label>Email</label><input type="email" name="email" id="edit_email" required></div>
    <div class="form-group"><label>Phone</label><input type="text" name="phone" id="edit_phone"></div>
    <div class="form-group"><label>Location</label><input type="text" name="address" id="edit_address"></div>
    <div class="form-group"><label>ISBN</label><input type="text" name="isbn" id="edit_isbn" required></div>
    <button type="submit" class="btn-search" style="width:100%; background:#2b4ca1;">>Update Changes</button>
  </form>
</div>
</div>

<footer>
  <div style="margin-bottom: 20px;">
    
    <p style="font-weight: 600; color: #2b4ca1; letter-spacing: 1px;">VENUS NATIONAL COLLEGE</p>
  </div>

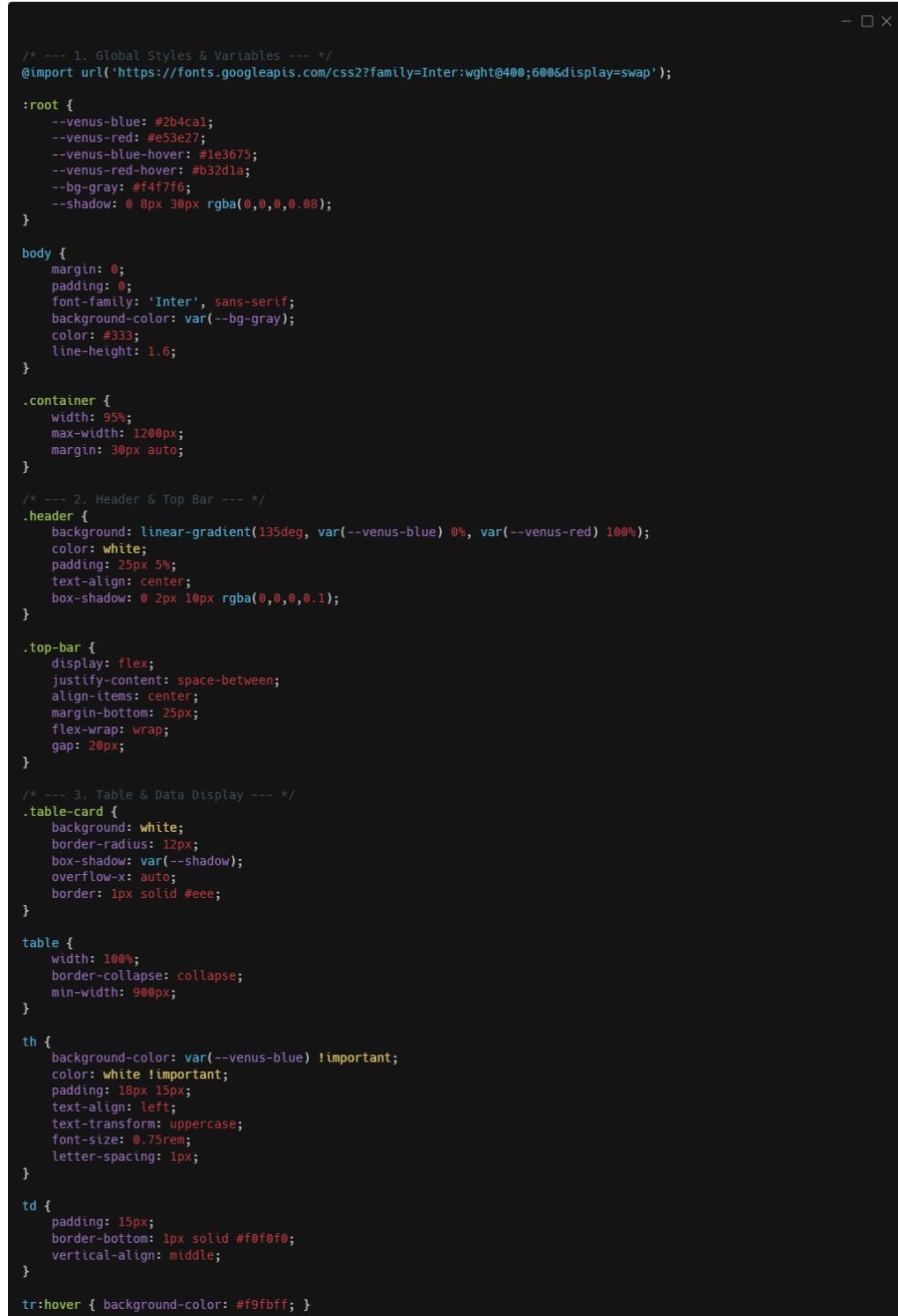
  <a href="https://github.com/Jangbu001/Library-management-CURD" target="_blank" class="github-link" style="display: inline-flex; align-items: center; gap: 10px; background: #333; color: white; padding: 10px 20px; border-radius: 50px; text-decoration: none; font-size: 0.9rem; transition: 0.3s;">
    <svg height="20" width="20" fill="white" viewBox="0 0 16 16"><path d="M8 0C3.58 0 0 3.58 0 8c0 3.54 2.29 6.53 5.47 7.59 4.07 5.55-.17 5.55-.38 0-.19-.01-.82-.01-1.49-2.01.37-2.53-.49-2.69-.94-.09-.23-.48-.94-.82-1.13-.28-.15-.68-.52-.01-.53.63-.01 1.08.58 1.23.82.72 1.21 1.87.87 2.33.66.07-.52.28-.87.51-1.07-1.78-.2-3.64-.89-3.64-3.95 0-.87.31-1.59.82-2.15-.08-.2-.36-1.02.08-2.12 0 .67-.21 2.2.82.64-.18 1.32-.27 2-.27.68 0 1.36.09 2 .27 1.53-1.04 2.2-.82 2.2-.82.44 1.1.16 1.92.08 2.12.51.56.82 1.27.82 2.15 0 3.07-1.87 3.75-3.65 3.95.29.25.54.73.54 1.48 0 1.07-.01 1.93-.01 2.2 0 .21.15.46.55.38A8.013 8.013 0 0016 8c0-4.42-3.58-8-8-8z"></path></svg>
    View Source on GitHub
  </a>
  <p style="font-size: 0.8rem; color: #999; margin-top: 25px; border-top: 1px solid #eee; padding-top: 15px;">
    © 2025 • Designed by <strong>Jangbu Sherpa</strong> • XII (Science)
  </p>
</footer>
</div>

<script src="script.js"></script>
</body>
</html>

```

style.css (The Design)

Contains the visual rules, including the unified height for buttons and the Venus College color palette.



```
/* --- 1. Global Styles & Variables --- */
@import url('https://fonts.googleapis.com/css2?family=Inter:wght@400;600&display=swap');

:root {
    --venus-blue: #2b4cal;
    --venus-red: #e53e27;
    --venus-blue-hover: #1e3675;
    --venus-red-hover: #b32d1a;
    --bg-gray: #f4f7f9;
    --shadow: 0 8px 30px rgba(0,0,0,.08);
}

body {
    margin: 0;
    padding: 0;
    font-family: 'Inter', sans-serif;
    background-color: var(--bg-gray);
    color: #333;
    line-height: 1.6;
}

.container {
    width: 95%;
    max-width: 1200px;
    margin: 30px auto;
}

/* --- 2. Header & Top Bar --- */
.header {
    background: linear-gradient(135deg, var(--venus-blue) 0%, var(--venus-red) 100%);
    color: white;
    padding: 25px 5%;
    text-align: center;
    box-shadow: 0 2px 10px rgba(0,0,0,.1);
}

.top-bar {
    display: flex;
    justify-content: space-between;
    align-items: center;
    margin-bottom: 25px;
    flex-wrap: wrap;
    gap: 20px;
}

/* --- 3. Table & Data Display --- */
.table-card {
    background: white;
    border-radius: 12px;
    box-shadow: var(--shadow);
    overflow-x: auto;
    border: 1px solid #eee;
}

table {
    width: 100%;
    border-collapse: collapse;
    min-width: 900px;
}

th {
    background-color: var(--venus-blue) !important;
    color: white !important;
    padding: 18px 15px;
    text-align: left;
    text-transform: uppercase;
    font-size: 0.75rem;
    letter-spacing: 1px;
}

td {
    padding: 15px;
    border-bottom: 1px solid #f0f0f0;
    vertical-align: middle;
}

tr:hover { background-color: #f0fbff; }
```

```

/* column widths */
table th:nth-child(7), table td:nth-child(7) { min-width: 160px; } /* ISBN Column */
.actions-cell { text-align: center; min-width: 120px; }

/* --- 4. Components (Badges & Buttons) --- */
.id-text {
    font-family: 'Courier New', monospace;
    font-weight: bold;
    color: var(--venus-blue);
    background: #eef2ff;
    padding: 4px 8px;
    border-radius: 4px;
}

.badge-isbn {
    background: #fff3f2;
    color: var(--venus-red);
    padding: 6px 12px;
    border-radius: 20px;
    font-size: 0.85rem;
    border: 1px solid #fffdcd8;
    white-space: nowrap;
    display: inline-block;
    font-weight: 600;
}

/* Primary Action Buttons */
.btn-add, .btn-search, .btn-clear {
    padding: 10px 20px;
    border-radius: 6px;
    font-weight: 600;
    text-decoration: none;
    transition: 0.3s;
    cursor: pointer;
    border: none;
    display: inline-block;
}

.btn-add, .btn-search { background: var(--venus-blue); color: white; }
.btn-add:hover, .btn-search:hover { background: var(--venus-blue-hover); transform: translateY(-1px); }

/* Container to hold input + clear button */
.search-container {
    position: relative;
    display: flex;
    align-items: center;
}

/* Updated Search Input to make room for the X icon */
input[name="search"] {
    padding: 10px 35px 10px 15px !important; /* Extra right padding for the X */
    border: 1px solid #ddd;
    border-radius: 6px;
    outline: none;
    width: 240px;
    transition: all 0.3s;
}

input[name="search"]:focus {
    border-color: var(--venus-blue);
    box-shadow: 0 0 5px rgba(43, 76, 161, 0.2);
}

/* The Clear Button now floats inside the input */
.btn-clear {
    position: absolute;
    right: 12px;
    background: transparent !important; /* Remove background */
    color: #999 !important; /* Subtle gray color */
    border: none;
    display: flex;
    align-items: center;
    justify-content: center;
    font-size: 16px; /* Adjust size as needed */
    text-decoration: none;
    transition: 0.2s;
    cursor: pointer;
    font-weight: bold;
}

.btn-clear:hover {
    color: var(--venus-red) !important; /* Turns red on hover */
    transform: scale(1.1); /* Slight pop effect */
}

/* Table Icon Buttons */
.action-group { display: flex; gap: 8px; justify-content: center; }

.btn-icon {
    width: 36px; height: 36px;
    display: flex; align-items: center; justify-content: center;
    border-radius: 6px; border: none; cursor: pointer;
    transition: 0.2s; color: white;
}

```

```

.btn-edit-icon { background: var(--venus-blue); }
.btn-delete-icon { background: var(--venus-red); }
.btn-icon:hover { transform: translateY(-2px); opacity: 0.9; }
.btn-icon svg { width: 18px; height: 18px; fill: currentColor; }

/* --- 5. Modals & Forms --- */
.modal {
  display: none; position: fixed; z-index: 9999;
  left: 0; top: 0; width: 100%; height: 100%;
  background-color: rgba(0,0,0,0.7); backdrop-filter: blur(5px);
  align-items: center; justify-content: center;
}

.modal-content {
  background: white; padding: 30px; border-radius: 12px;
  width: 95%; max-width: 500px; position: relative;
  box-shadow: 0 15px 50px rgba(0,0,0,0.5);
  animation: popIn 0.3s cubic-bezier(0.68, -0.55, 0.27, 1.55);
}

@keyframes popIn { from { transform: scale(0.8); opacity: 0; } to { transform: scale(1); opacity: 1; } }

.close { position: absolute; right: 20px; top: 15px; font-size: 24px; cursor: pointer; color: #999; }
.close:hover { color: var(--venus-red); }

.form-group { margin-bottom: 15px; }
.form-group label { display: block; margin-bottom: 5px; font-weight: 600; color: var(--venus-blue); font-size: 0.9rem; }
.form-group input {
  width: 100%; padding: 12px; border: 1px solid #ddd;
  border-radius: 6px; box-sizing: border-box; font-family: inherit;
}

/* --- 6. Footer --- */
footer {
  text-align: center; margin-top: 60px; padding: 40px 20px;
  background: white; border-top: 4px solid var(--venus-blue);
}

.footer-logo { width: 80px; margin-bottom: 10px; }

/* --- 7. Responsive Design --- */
@media screen and (max-width: 768px) {
  .top-bar { flex-direction: column; align-items: stretch; text-align: center; }
  input[name="search"] { width: 100% !important; }
}

```

script.js (The Logic)

Handles the interactive elements, such as opening modals and populating the update form.

```
/*
 * Venus Library Management - Interface Logic
 * Developer: Jangbu Sherpa
 */

// Function to open any modal
function openModal(id) {
    const modal = document.getElementById(id);
    modal.style.display = 'flex';

    // Focus the first input field automatically
    const firstInput =
modal.querySelector('input[type="text"], [type="hidden"]');
}

// Function to close any modal
function closeModal(id) {
    document.getElementById(id).style.display = 'none';
}

// Function to handle Edit button click and populate data
function handleEdit(data) {
    document.getElementById('edit_id').value = data.id;
    document.getElementById('edit_title').value = data.title;
    document.getElementById('edit_author').value = data.author;
    document.getElementById('edit_email').value = data.email;
    document.getElementById('edit_phone').value = data.phone;
    document.getElementById('edit_address').value = data.address;
    document.getElementById('edit_isbn').value = data.isbn;

    openModal('editModal');
}

// Close modal if user clicks outside the modal box
window.onclick = function(event) {
    if (event.target.classList.contains('modal')) {
        event.target.style.display = 'none';
    }
};
```

```

// Keyboard Shortcuts
document.addEventListener('keydown', function(event) {
    // Close modal on 'Escape' key
    if (event.key === "Escape") {
        document.querySelectorAll('.modal').forEach(modal => {
            modal.style.display = 'none';
        });
    }
});

```

process.php (The Backend Controller)

Handles the logic for both **adding** new books and **updating** existing ones.

```

<?php
$conn = new mysqli('localhost', 'root', '', 'libary_management');

if (isset($_GET['action'])) {
    $action = $_GET['action'];
    $title = $conn->real_escape_string($_POST['title']);
    $author = $conn->real_escape_string($_POST['author']);
    $email = $conn->real_escape_string($_POST['email']);
    $phone = $conn->real_escape_string($_POST['phone']);
    $address = $conn->real_escape_string($_POST['address']);
    $isbn = $conn->real_escape_string($_POST['isbn']);

    if ($action == 'add') {
        $conn->query("INSERT INTO books (title, author, email, phone, address, isbn) VALUES ('$title', '$author',
'$email', '$phone', '$address', '$isbn')");
    } elseif ($action == 'edit') {
        $id = $_POST['id'];
        $conn->query("UPDATE books SET title='$title', author='$author', email='$email', phone='$phone',
address='$address', isbn='$isbn' WHERE id=$id");
    }
}
header("location: index.php");
exit;
?>

```

delete.php (The Removal Logic)

A specialized script to safely remove records from the database.

```
<?php
// 1. Establish connection to the Venus Library database
$conn = new mysqli('localhost', 'root', '', 'library_management');

// 2. Check if the 'id' is passed in the URL
if (isset($_GET["id"])) {
    $id = $_GET["id"];

    // 3. Delete the record from the 'books' table
    $sql = "DELETE FROM books WHERE id=$id";
    $conn->query($sql);
}

// 4. Redirect back to the main inventory page
header("location: index.php");
exit;
?>
```

CONCLUSION

The development of the **Library Management System** successfully demonstrates the power of integrating web technologies with relational databases to solve real-world organizational challenges. By moving from a manual, paper-based system to a digital CRUD-based application, the project has achieved its primary goals of efficiency, accuracy, and accessibility.

The diagram illustrates the Create and Update processes in a Library Management System, showing the flow from data entry to database update.

Create Process:

- A user clicks the **+ Register Book** button, opening the **Register New Book** modal.
- The modal contains fields for Title (C-programming), Author (Dorjee Sherpa), Email (dorje.sherpa1987@gmail.com), Phone (984-3446367), Location (Arubari, Jorpati, Kathmandu), and ISBN (978-9937-23).
- The user clicks the **Save Book** button to save the new book record.
- The saved book record is displayed in the main table:

ID	BOOK TITLE	AUTHOR	EMAIL	PHONE	LOCATION	ISBN	ACTIONS
0025	C-programming	Dorjee Sherpa	dorje.sherpa1987@gmail.com	984-3446367	Arubari, Jorpati, Kathmandu	978-9937-23	

Update Process:

- A user clicks the edit icon () next to the book record in the main table.
- This opens the **Update Details** modal, showing the current data: Title (C-programming), Author (Dorjee Sherpa), Email (dorje.sherpa1987@gmail.com), Phone (9843446367), Location (Arubari, Jorpati, Kathmandu), and ISBN (978-9937-23).
- The user changes the phone number to 9843446367.
- The user clicks the **Update Changes** button to save the updated record.
- The updated book record is displayed in the main table:

ID	BOOK TITLE	AUTHOR	EMAIL	PHONE	LOCATION	ISBN	ACTIONS
0025	C-programming	Dorjee Sherpa	dorje.sherpa1987@gmail.com	9843446367	Arubari, Jorpati, Kathmandu	978-9937-23	

LIBRARY MANAGEMENT

Inventory Overview
Standard XII Science Project Portfolio

ID	BOOK TITLE	AUTHOR	EMAIL	PHONE	LOCATION	ISBN	ACTIONS
0025	C- programming	Dorjee Sherpa	dorje.sherpa987@gmail.com	9843446367	Arubari, Jorpati, Kathmandu	978-9937-23	
0022	Summer Love	Subin Bhattarai	subin.bhattarai@writer.np	9812345678	Bhaktapur, Nepal	978-9937-08	
0021	Seto Dharti	Amar Neupane	amar.neupane@iit.np	9808877665	Pokhara, Kaski	978-9937-07	
0020	The Alchemist	Paulo Coelho	contact@paulocoelho.com	9851099887	Rio de Janeiro, Brazil	978-0062-31	
0019	Shirishko Phool	Parijat	parijat.trust@nepal.np	9841556677	Ranibari, Kathmandu	978-9937-05	

read

ID	BOOK TITLE	AUTHOR	EMAIL	PHONE	LOCATION	ISBN	ACTIONS
0022	Summer Love	Subin Bhattarai	subin.bhattarai@writer.np	9812345678	Bhaktapur, Nepal	978-9937-08	
0021	Seto Dharti	Amar Neupane	amar.neupane@iit.np	9808877665	Pokhara, Kaski	978-9937-07	
0020	The Alchemist	Paulo Coelho	contact@paulocoelho.com	9851099887	Rio de Janeiro, Brazil	978-0062-31	
0019	Shirishko Phool	Parijat	parijat.trust@nepal.np	9841556677	Ranibari, Kathmandu	978-9937-05	
0018	Karnali Blues	Buddhisagar	buddhisagar@fineprint.np	9811223344	Maitighar, Kathmandu	978-9937-04	

delete

Inventory Overview
Standard XII Science Project Portfolio

seto

ID	BOOK TITLE	AUTHOR	EMAIL	PHONE	LOCATION	ISBN	ACTIONS
0021	Seto Dharti	Amar Neupane	amar.neupane@iit.np	9808877665	Pokhara, Kaski	978-9937-07	
0016	Seto Bagh	Diamond Shumsher Rana	diamond.rana@history.np	9851011223	Thamel, Kathmandu	978-9937-02	

VENUS NATIONAL COLLEGE

[View Source on GitHub](#)

search

Key Outcomes

- Operational Efficiency:** The system has significantly reduced the time required to register and search for book records through the implementation of a dynamic search algorithm and an optimized MySQL database.

- **User Experience:** By utilizing a modern frontend with JavaScript modals, the application provides a smooth, "single-page" feel that minimizes page reloads and enhances user productivity.
- **Data Integrity:** The use of standardized data types and primary keys in MySQL ensures that every book is uniquely identified, preventing duplicate entries and data loss.

Academic Value

As a **Standard XII Science Project**, this endeavor provided invaluable practical experience in the full software development lifecycle (SDLC). It bridged the gap between theoretical computer science concepts—such as database normalization and server-side scripting—and practical, functional software design.

Final Remarks

In conclusion, the Library Management System serves as a robust prototype that can be further expanded into a full-scale college ERP module. It stands as a testament to how simple yet well-structured code can modernize traditional workflows, making the library a more organized and technology-forward environment for students and faculty at **Venus National College**.