

# Library Management System

## Complete User Manual for Students

---

### Table of Contents

1. [Project Overview](#)
  2. [System Requirements](#)
  3. [Installation Guide](#)
  4. [Running the Application](#)
  5. [First-Time Login](#)
  6. [How to Use the System](#)
  7. [Common Problems & Solutions](#)
  8. [Quick Reference](#)
- 

### Project Overview {#overview}

The **Library Management System** is a web-based application designed to help manage books, readers, and borrow records in a library. Instead of using paper registers and manual tracking, this system digitizes the entire library process.

#### What This System Does:

- Keep track of all library books and their availability
- Manage reader/student information and accounts
- Record when books are borrowed and returned
- Track overdue books automatically
- Search for books and readers quickly
- View complete borrowing history

#### Technology Used:

- Python programming language
  - Flask web framework
  - MySQL database
  - HTML, CSS, and JavaScript for the web interface
-

# System Requirements

Before starting, ensure your computer has these programs installed:

## 1. Python (Version 3.8 or higher)

- **Download:** <https://www.python.org/downloads/>
- **Important:** During installation, check the box "Add Python to PATH"
- **Verify Installation:** Open Command Prompt and type `python --version`

## 2. MySQL (Version 8.0 or higher)

- **Download:** <https://dev.mysql.com/downloads/installer/>
- **Important:** Remember your root password during installation
- **Recommended Tool:** MySQL Workbench (user-friendly GUI)

## 3. Git (Optional but Recommended)

- **Download:** <https://git-scm.com/downloads>
- **Use:** For downloading the project easily

## 4. Code Editor (Optional)

- **Recommended:** Visual Studio Code (<https://code.visualstudio.com/>)
  - **Use:** To edit configuration files if needed
- 

# Installation Guide

## Step 1: Download the Project

### Manual Download

1. Download the project as a ZIP file
2. Extract it to a folder on your computer
3. Remember the folder location

## Step 2: Set Up the Database

### Open MySQL Workbench:

1. Launch MySQL Workbench
2. Click on your local MySQL connection
3. Enter your password if prompted

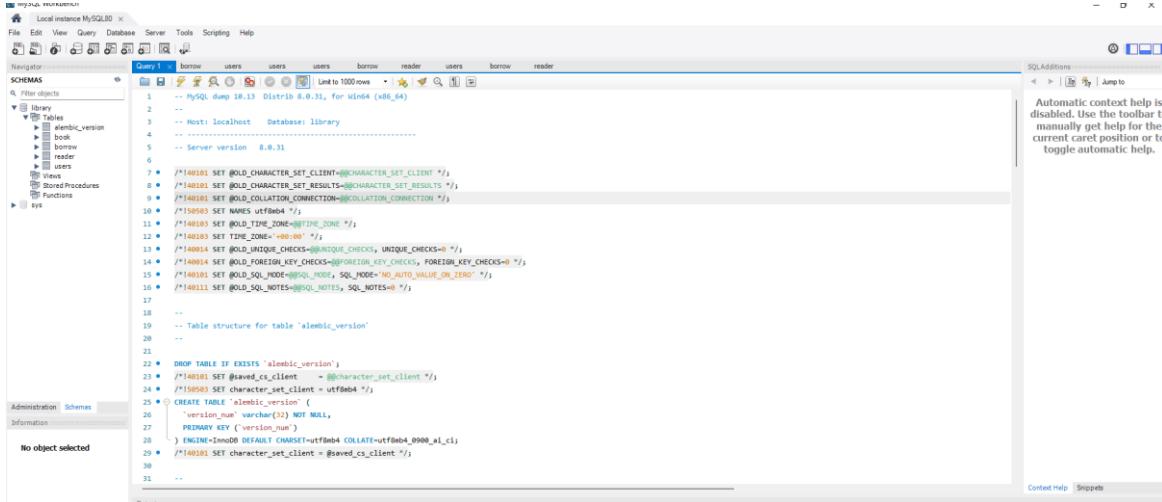
4. Click "File" → "Open SQL Script"
5. Navigate to the `data.sql` file in your project folder
6. Click the  (lightning bolt) icon to execute the script
7. Watch as the tables appear in the left window under "SCHEMAS"

#### Verify the database was created:

- You should see: `books`, `reader`, `borrow`, and `users` tables in the left panel.

### REFER THE YOUTUBE VIDEO FOR BETTER CLARITY....

[https://www.google.com/search?q=mysql+set+up+video+in+tamil&oq=mysql+set+up+video&gs\\_lcp=EgZjaHJvbWUqCQgCECEYChigATIGCAAQRg5MgkIARAhGAoYoAEyCQgCECEYChigATIJCAMQIRgKGKABMgcIBBAhGJ8FMgcIBRAhGJ8FMgcIBhAhGJ8FMgcIBxAhGJ8FMgcICBAhGI8CogELMTIzNDk4ajBqMTWoAgiwAgHxBUIo3d-7WPI-&sourceid=chrome&ie=UTF-8#fpstate=ive&vld=cid:1591c919,vid:UGryKljGAHs,st:o](https://www.google.com/search?q=mysql+set+up+video+in+tamil&oq=mysql+set+up+video&gs_lcp=EgZjaHJvbWUqCQgCECEYChigATIGCAAQRg5MgkIARAhGAoYoAEyCQgCECEYChigATIJCAMQIRgKGKABMgcIBBAhGJ8FMgcIBRAhGJ8FMgcIBhAhGJ8FMgcIBxAhGJ8FMgcICBAhGI8CogELMTIzNDk4ajBqMTWoAgiwAgHxBUIo3d-7WPI-&sourceid=chrome&ie=UTF-8#fpstate=ive&vld=cid:1591c919,vid:UGryKljGAHs,st:o)



```

1 -- MySQL dump 10.13 Distrib 8.0.31, for Win64 (x86_64)
2
3 -- Host: localhost Database: library
4
5 -- Server version 8.0.31
6
7 /*!40101 SET @OLD_CHARACTER_SET_CLIENT=@CHARACTER_SET_CLIENT */;
8 /*!40101 SET @OLD_CHARACTER_SET_RESULTS=@CHARACTER_SET_RESULTS */;
9 /*!40101 SET @OLD_COLLATION_CONNECTION=@COLLATION_CONNECTION */;
10 /*!50503 SET NAMES utf8mb4 */;
11 /*!40101 SET @OLD_TIME_ZONE=@TIME_ZONE */;
12 /*!40101 SET TIME_ZONE='+00:00' */;
13 /*!40014 SET @OLD_UNIQUE_CHECKS=@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
14 /*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0 */;
15 /*!40101 SET @OLD_SQL_MODE=@SQL_MODE, SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
16 /*!40111 SET @OLD_SQL_NOTES=@SQL_NOTES, SQL_NOTES=0 */;
17
18
19 -- Table structure for table 'alembic_version'
20 --
21
22 /*!40101 TABLE IF EXISTS `alembic_version` */;
23 /*!40101 SET @saved_cs_client = @@character_set_client */;
24 /*!40101 SET character_set_client = utf8mb4 */;
25 /*!40000 CREATE TABLE `alembic_version` (
26     `version_num` varchar(32) NOT NULL,
27     PRIMARY KEY (`version_num`)
28 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci */;
29 /*!40101 SET character_set_client = @saved_cs_client */;
30
31 */

```

### Step 3: Configure the Database Connection

1. Open the project folder on your computer
2. Find and open `librarysys.py` file (use Notepad or VS Code)
3. Locate line 18, which looks like:

```
app.config['SQLALCHEMY_DATABASE_URI'] =
'mysql://root:1234567890@localhost/library'
```

4. Replace `1234567890` with YOUR MySQL root password
5. **Save the file** (Ctrl+S)

**Example:** If your password is mypassword123:  
app.config['SQLALCHEMY\_DATABASE\_URI'] =  
'mysql://root:mypassword123@localhost/library'

## Step 4: Install Python Packages

### On Windows (PowerShell):

1. Open PowerShell in your project folder
2. Create virtual environment:  
python -m venv .venv
3. Activate it:  
.\\venv\\Scripts\\Activate.ps1
4. Install packages:  
pip install -r requirements.txt

### On Mac/Linux:

1. Open Terminal in your project folder
2. Create virtual environment:  
python -m venv .venv
3. Activate it:  
source .venv/bin/activate
4. Install packages:  
pip install -r requirements.txt

**Note:** After activation, you should see (.venv) at the start of your terminal line.

---

## Running the Application {#running}

### Starting the Application

#### In PowerShell/Terminal (with .venv activated):

1. Ensure virtual environment is active (you see (.venv) in terminal)
2. Set the Flask app:  
\$env:FLASK\_APP = [librarysys.py](#)
3. Start the application:  
flask run
4. You should see:  
Running on <http://127.0.0.1:5000>

### Accessing the Application

1. Open your web browser (Chrome, Firefox, Edge, etc.)
2. Type in the address bar: <http://127.0.0.1:5000>

3. Press Enter
4. You should see the **Library Management System login page**

**Important:** Use `http://` (not `https://`) and ensure Flask is running in your terminal.

## Stopping the Application

In your terminal, press `Ctrl+C` to stop the application.

---

## First-Time Login {#login}

### Default Admin Login

#### First login credentials:

- **Email:** smirthi@gmail.com
- **Password:** password

#### Steps:

1. On the login page, click "**Librarian Login**"
2. Enter email: `smirthi@gmail.com`
3. Enter password: `password`
4. Click "**Login**" button
5. You should see the **Librarian Dashboard**

**Success!** You're now inside the system.

---

## How to Use the System {#usage}

### Navigation Menu

Once logged in, you'll see a navigation bar with these options:

Option	What It Does
<b>Books</b>	View all books in the library
<b>Add Book</b>	Add a new book to the system
<b>Readers</b>	View and manage all library members
<b>New Borrow</b>	Record when someone borrows a book
<b>Active Borrows</b>	See which books are currently borrowed

<b>History</b>	View all past transactions and records
<b>Logout</b>	Exit the system safely

---

## Task 1: Adding a New Book

### Steps:

1. Click "**Add Book**" in the navigation menu
2. Fill in the form:
  - o **Title:** Book name (e.g., "Harry Potter and the Philosopher's Stone")
  - o **Author:** Author name (e.g., "J.K. Rowling")
  - o **Synopsis:** Short description of the book
3. Click "**Submit**" button
4. Success message appears, book is added to the system

**Result:** Book appears in the "Books" list with status "Available"

---

## Task 2: Managing Readers/Students

### Adding a New Reader

#### Option A: Student Self-Registration

1. Send students the login link: <http://127.0.0.1:5000>
2. Click "**Reader Registration**" or "**Register**"
3. Students fill in their details (Name, Email, Password, etc.)
4. Click "**Sign Up**"
5. Account created, student can now login

#### Option B: Librarian Adds Reader

1. Click "**Readers**" in navigation menu
2. Click "**Add New Reader**" button
3. Fill in student information
4. Click "**Submit**"
5. Reader account created

### Viewing All Readers

1. Click "**Readers**" in navigation menu
2. See list of all registered students

3. Search for a specific student using the search bar

## Editing Student Information

1. Go to "**Readers**" section
  2. Find the student name
  3. Click on their profile or "**Edit**" button
  4. Update information
  5. Click "**Save**"
- 

## Task 3: Recording a Book Borrow

### When a student borrows a book:

1. Click "**New Borrow**" in navigation menu
2. You'll see a form asking for:
  - o **Book ID:** Find this by going to "Books" and noting the number next to the book name
  - o **Borrower ID:** Find this by going to "Readers" and noting the student's ID
3. Enter both IDs
4. Click "**Submit**"
5. Book status changes to "**Not Available**" (Red color)
6. System records the borrow date automatically

**Tip:** Write down the Book ID and Reader ID to make this faster.

---

## Task 4: Recording a Book Return

### When a student returns a book:

1. Click "**Active Borrows**" in navigation menu
2. See list of all currently borrowed books
3. Find the book being returned
4. Click "**Confirm Return**" or "**Mark Returned**" button
5. System updates the book status to "**Available**" (Green color)
6. Return date is recorded automatically

**Result:** Student's record is updated, book can be borrowed by someone else.

---

## Task 5: Searching for Books and Readers

## **Search for a Book**

1. Use the **search bar** at the top of the "Books" page
2. Type the book title or author name
3. Results appear instantly
4. Click on a book to see full details

## **Search for a Reader**

1. Go to "**Readers**" section
2. Use the search bar
3. Type student name or ID
4. Results appear instantly

## **View Borrowing History**

1. Click "**History**" in navigation menu
  2. See all past borrow and return records
  3. Use search/filter options to find specific records
  4. Dates and details are shown for each transaction
- 

## **Understanding Status Colors**

### **Book Status Indicators:**

Color	Status	Meaning
<span style="color: green;">●</span> <b>Green</b>	Available	Book can be borrowed
<span style="color: red;">●</span> <b>Red</b>	Not Available	Book is currently borrowed
<span style="color: yellow;">●</span> <b>Yellow</b>	Overdue	Book should have been returned

### **What to do with Yellow (Overdue):**

- Follow up with the student who borrowed the book
  - Arrange for return
  - Update return date if extended
- 

## **Common Problems & Solutions {#troubleshooting}**

## **Problem 1: "Access Denied" when starting the application**

**Cause:** MySQL password in `librarysys.py` is wrong

**Solution:**

1. Open `librarysys.py` in a text editor
  2. Go to line 18
  3. Check the password is correct:  
`mysql://root:YOUR_PASSWORD@localhost/library`
  4. Make sure it matches your MySQL root password exactly
  5. Save the file
  6. Restart Flask (press Ctrl+C and type `flask run` again)
- 

## **Problem 2: "ModuleNotFoundError" or "No module named..."**

**Cause:** Python packages not installed

**Solution:**

1. Ensure virtual environment is activated (you see `(.venv)` in terminal)
  2. Run:  
`pip install -r requirements.txt`
  3. Wait for installation to complete
  4. Try running Flask again: `flask run`
- 

## **Problem 3: Cannot access website at <http://127.0.0.1:5000>**

**Cause:** Flask application is not running

**Solution:**

1. Check your terminal - does it show "Running on <http://127.0.0.1:5000>"?
  2. If not, run: `flask run`
  3. Open a new browser tab and try again
  4. **Important:** Use `http://` not `https://`
  5. Check firewall settings (Windows might block it)
-

## **Problem 4: "Tables don't exist" error**

**Cause:** Database wasn't set up properly

**Solution:**

1. Open MySQL Workbench
  2. Go to "File" → "Open SQL Script"
  3. Select `data.sql` file
  4. Click the  (lightning) icon
  5. Verify tables appear in left panel under SCHEMAS
  6. Restart Flask
- 

## **Problem 5: Cannot login with admin account**

**Cause:** Admin account not created properly

**Solution:**

1. Ensure you followed Step 2 (database setup) correctly
  2. Default credentials should work:
    - o Email: `admin@library.com`
    - o Password: `admin123`
  3. If still not working, check MySQL to verify the users table has the admin entry
- 

## **Problem 6: Website looks broken (missing colors/formatting)**

**Cause:** CSS file not loading

**Solution:**

1. Check that folder `static/css/library.css` exists in your project
  2. Press `Ctrl+F5` in browser (hard refresh) to clear cache
  3. Try a different browser
  4. Restart Flask application
- 

## **Problem 7: Wrong FLASK\_APP setting error**

**Cause:** Incorrect Flask app file name

**Solution:**

1. Set it correctly:  
\$env:FLASK\_APP = `librarysys.py`
  2. Make sure you're in the project folder
  3. The filename must be exactly `librarysys.py`
  4. Restart Flask
- 

## Quick Reference {#reference}

### Key Files and Folders

Project Folder/

```
|   └── librarysys.py ← Main application file
|   └── webforms.py ← Login/registration forms
|   └── data.sql ← Database structure
|   └── requirements.txt ← Python packages needed
|
|   └── static/
|       └── css/
|           └── library.css ← Styling
|
|   └── templates/
|       └── base.html ← Main template
|       └── librarian_login.html
|       └── reader_login.html
|       └── registration.html
|       └── bookview.html
|           └── ... (other pages)
|
└── .venv/ ← Virtual environment (created by you)
```

### Quick Command Reference

Task	Windows PowerShell	Mac/Linux Terminal
Create venv	<code>python -m venv .venv</code>	<code>python3 -m venv .venv</code>
Activate venv	<code>.\venv\Scripts\Activate.ps1</code>	<code>source .venv/bin/activate</code>
Install packages	<code>pip install -r requirements.txt</code>	<code>pip install -r requirements.txt</code>
Set Flask app	<code>\$env:FLASK_APP = librarysys.py</code>	<code>export FLASK_APP=librarysys.py</code>
Run application	<code>flask run</code>	<code>flask run</code>

Stop application	<code>Ctrl+C</code>	<code>Ctrl+C</code>
------------------	---------------------	---------------------

## Useful URLs

Location	URL
Application Home	<code>http://127.0.0.1:5000</code>
Librarian Login	<code>http://127.0.0.1:5000/librarianlogin</code>
Reader Login	<code>http://127.0.0.1:5000/readerlogin</code>
Registration	<code>http://127.0.0.1:5000/register</code>

## Default Credentials

User Type	Email	Password
Librarian	<code>admin@library.com</code>	<code>admin123</code>
Reader	Self-created	Self-created

## Important Passwords to Remember

- **MySQL Root Password:** (The one YOU set during MySQL installation)
  - **Admin Email:** `admin@library.com`
  - **Admin Password:** `admin123`
- 

## Tips to Avoid Lag and Problems

1. **Keep virtual environment activated** - Always see (`.venv`) in your terminal
  2. **Use correct password in `librarysys.py`** - Exactly as set during MySQL installation
  3. **Don't close the Flask terminal** - Application needs it to keep running
  4. **Use modern browser** - Chrome, Firefox, or Edge work best
  5. **Clear browser cache** - Press `Ctrl+F5` if page looks wrong
  6. **Check database connection** - Ensure MySQL is running before starting Flask
  7. **Install all packages** - Don't skip `pip install -r requirements.txt`
  8. **Use http not https** - Always `http://127.0.0.1:5000`
-

# Need More Help?

If you encounter issues:

1. **Check this manual** - Most problems are covered in the Troubleshooting section
2. **Look at terminal errors** - Error messages tell you exactly what's wrong
3. **Verify each installation step** - Go back and check you didn't miss anything
4. **Restart everything** - Close Flask, restart the terminal, try again
5. **Check file paths** - Make sure all files are in the correct locations

---

**Congratulations!** You now have the complete Library Management System ready to use.  
Happy managing! 

*Last Updated: January 2026*