USING SQLITE ,TKINTER AND PYTHON AND DEVELOPING LIBRARY MANAGEMENT SYSTEM

PROJECT REPORT

Submitted by

Aqib Baig 2022-310-037

in partial fulfilment for the award of the degree of

Bachelor of Technology Computer Science & Engineering

Under the supervision of

Dr. SURAIYA PARWEEN

Department of Computer Science & Engineering

School of Engineering Sciences & Technology

JAMIA HAMDARD
New Delhi-110062

3rd Year

DECLARATION

I, Mr. **Aqib Baig** a student of **Bachelor of Technology Computer Science & Engineering (B. TECH CSE), (Enrolment No: 2022-310-037)** hereby declare that the Project/Dissertation entitled“**USING SQLITE,TKINTER AND PYTHON AND DEVELOPING LIBRARY MANAGEMENT SYSTEM**” which is being submitted by me to the Department of Computer Science, Jamia Hamdard, New Delhi in partial fulfilment of the requirement for the award of the degree of **Bachelor of Technology Computer Science & Engineering** (B. TECH CSE), is my original work and has not been submitted anywhere else for the award of any Degree,
Diploma, Associateship, Fellowship or other similar title or recognition.

(Aqib Baig)

Date: : 06: 2025

Place: NEW DELHI

ACKNOWLEDGEMENT

I would like to express my gratitude to all of the individuals who contributed to the successful completion of Major Project from start to finish. My supervisor, DR. SURAIYA PARWEEN, and I are connected by a deep sense of affection and gratitude for their support and direction. I'm also grateful to my Parents, Sisters, & Brother for their spiritual support as I worked on this project.

Finally, I'd like to thank everyone for their kind words and encouragement in keeping this project on track and seeing it through to completion. I apologized to those who were unable to do so here.

"Thank you for driving me to do my best work and for always being there to provide me guidance when I needed it. I was able to see my inner power and was motivated to develop and take on more by being in your company. You have been an excellent instructor of classes both inside and outside of the classroom."

ABSTRACT

The **USING SQLITE, TKINTER AND PYTHON AND DEVELOPING LIBRARY MANAGEMENT SYSTEM** is a desktop-based application developed using **Python**, with **Tkinter** for the graphical user interface and **SQLite** for database management. This system is designed to simplify the process of managing book records in a library, allowing users to add, view, and delete books with ease.

The application provides an intuitive interface for librarians or users to input details such as book title, author, borrowing date, and due date. Each record is timestamped upon addition for tracking purposes. The stored data is maintained in a local SQLite database to ensure persistence and reliability.

Key features include:

- Adding new book records with time stamps.
- Viewing all stored books in a list format. •

  Deleting selected books from the database.

1.Title:

Library Management System Using Python and SQLite

2. Objective:

The primary objective of this project is to design and develop a **Library Management System** that simplifies the process of managing book borrowing activities in a library using Python with a graphical user interface (GUI) built on Tkinter and a backend database managed by SQLite. This system is intended to assist librarians and users by automating key tasks such as adding new books, viewing available books, tracking borrowed and due dates, and deleting book records.

Specifically, the system aims to:

- **Provide a user-friendly GUI** that allows easy navigation and interaction.
- **Facilitate book management**, including insertion, display, and deletion of records.
- **Maintain borrowing records** by storing the date borrowed and due date for each book.
- **Timestamp entries** with the actual date and time a book is added to the database.
- **Enable data persistence** using a lightweight and local SQLite database.
- **Ensure data integrity and prevent duplication** by assigning a unique ID to each book entry.

3. Introduction:

- In today's digital era, automation has become an essential aspect of every sector, including education and information management. Libraries play a crucial role in the academic and personal development of individuals, and managing them efficiently is a task of significant importance. Traditionally, library records have been maintained manually using registers, files, or spreadsheets, which often leads to errors, inefficiencies, and data redundancy. To overcome these limitations, a digital Library Management System is needed.

- This project aims to create a simple yet effective Library Management System using **Python**, **Tkinter**, and **SQLite**. Python is a powerful programming language that supports GUI development through Tkinter, and SQLite is a lightweight, serverless database that stores data in a structured manner. This combination provides a portable, low-maintenance solution ideal for small and medium-sized libraries.

- This system will assist librarians in managing the borrowing and return of books by offering functionalities to add, view, and delete book records. Each record includes vital information such as the book title, author name, date borrowed, and return due date. The records are stored persistently in the SQLite database, ensuring data consistency even when the application is closed or restarted.

- The graphical interface makes the system intuitive and easy to use, eliminating the need for technical expertise. It reduces paperwork, minimizes human error, and improves overall efficiency. Additionally, it introduces features like real-time timestamps for book entries and a visual listbox to display all records dynamically.

- Key features of the system include:

- **Add Book Functionality**: Users can input and save new book details to the database.

- **View Book Records**: All added books are viewable in a scrollable listbox.

- **Delete Records**: Users can delete selected book records directly from the GUI.

- **Persistent Storage**: All records are stored in a local SQLite database.

- This project not only demonstrates database interaction in Python but also showcases the power of combining frontend (Tkinter) and backend (SQLite) technologies in realworld applications. It is a practical example for students and developers interested in learning how to build desktop-based CRUD (Create, Read, Update, Delete) applications.

3. Problem Statement:

In most educational institutions, small community libraries, or personal book collections, the management of books and tracking of borrow/return dates are typically done manually using registers or spreadsheets. This traditional approach presents numerous challenges, including: **Human Error**: Manual entry of book details increases the chance of inaccuracies such as misspelled titles, incorrect dates, or duplicated records.

Time Consumption: Updating, searching, and maintaining paper-based records is a timeconsuming process, especially as the collection grows.

Data Loss: Physical logs are prone to damage or loss due to environmental factors, mishandling, or accidental deletion in digital files like Excel.

Lack of Real-Time Access: Manual systems do not provide real-time access or updates, causing delays in book lending or return processes.

Limited Scalability: As the number of users and books grows, the manual system becomes difficult to scale and manage efficiently.

No Search Functionality: Finding specific records or generating reports is tedious and often requires manually going through logs.

Libraries using manual systems face problems such as data loss, redundancy, and difficulty in tracking borrow/return dates.

There is no quick search or organized display of available books.

Borrower information and due dates often get misplaced.

The need for a robust, easy-to-use, and accessible digital system is evident.

5. Software Requirements Specification (SRS):

The graphical interface makes the system intuitive and easy to use, eliminating the need for technical expertise. It reduces paperwork, minimizes human error, and improves overall efficiency. Additionally, it introduces features like real-time timestamps for book entries and a visual listbox to display all records dynamically.

Key features of the system include:

Add Book Functionality: Users can input and save new book details to the database.
View Book Records: All added books are viewable in a scrollable listbox.
Delete Records: Users can delete selected book records directly from the GUI.
Persistent Storage: All records are stored in a local SQLite database.

This project not only demonstrates database interaction in Python but also showcases the power of combining frontend (Tkinter) and backend (SQLite) technologies in real-world applications. It is a practical example for students and developers interested in learning how to build desktop-based CRUD (Create, Read, Update, Delete) applications.

By implementing this system, libraries can move one step closer to digital transformation, reduce dependency on paper records, and ensure better management of their book collections.

Functional Requirements

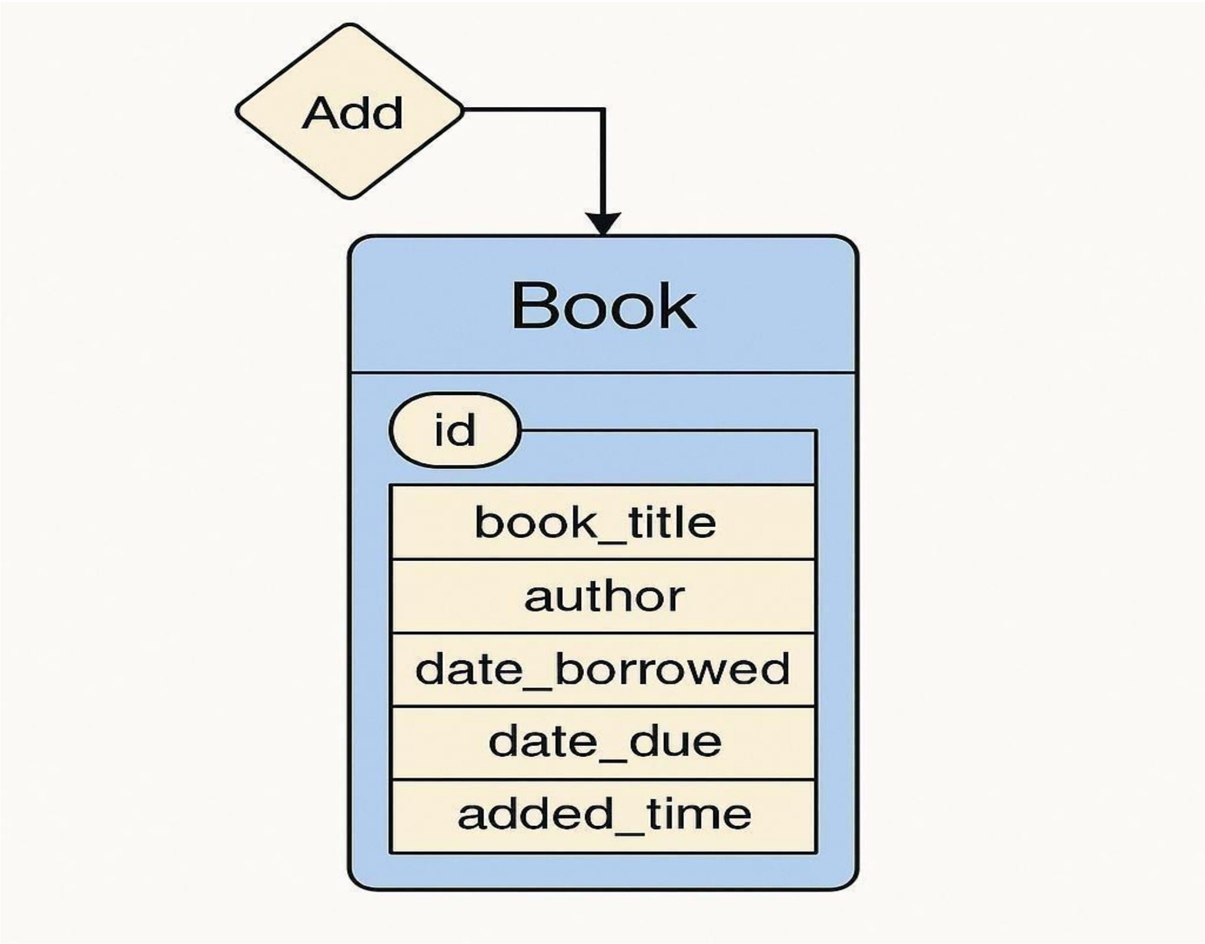
1. Add a new book
2. View list of books
3. Delete a selected book **Non-functional Requirements**

1. Simple GUI using Tkinter.
2. Local database management using SQLite.
3. Must respond within 2 seconds.
4. Error handling for user input.

6. Entity Relationship Diagram:

Entities: Book, User (optional)

Attributes of Book: ID, Title, Author, Date Borrowed, Date Due, Added Time **Relationships:** A Book is added, viewed, or deleted by the system admin



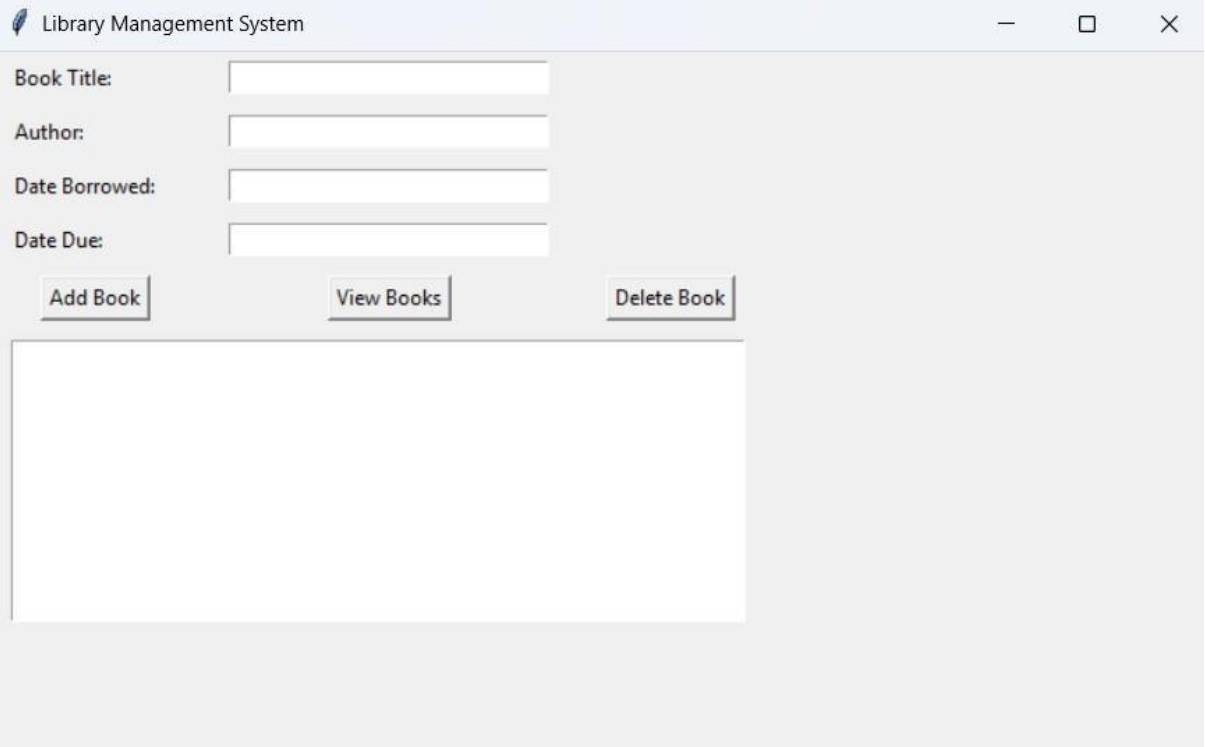
Test Case

6. Sample Test Cases:

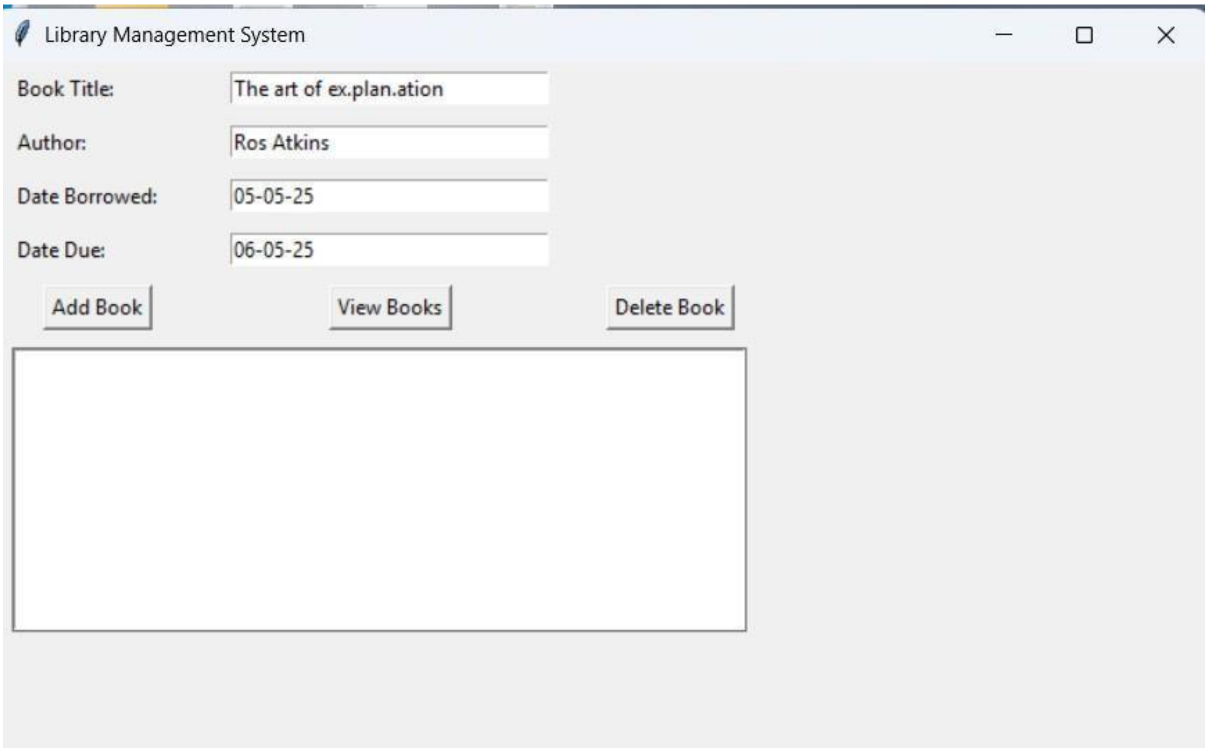
Test Case **ID**	**Description**	**Input Values** Title, Author, Dates	**Expected Result**
TC01	Add Book		Book added successfully
TC02	View Books	-	List of books displayed
TC03 listbox	Delete Book (valid ID)	Selected book from	Book deleted
TC04 fields	Add Book with empty	Title or Author missing	Warning message shown

8. Snapshots:

(Include GUI screenshots of Add Book form, View List, and Deletion confirmation dialogs)


Library Management System
Book Title:
Author:
Date Borrowed:
Date Due:
Add Book
View Books
Delete Book

Started

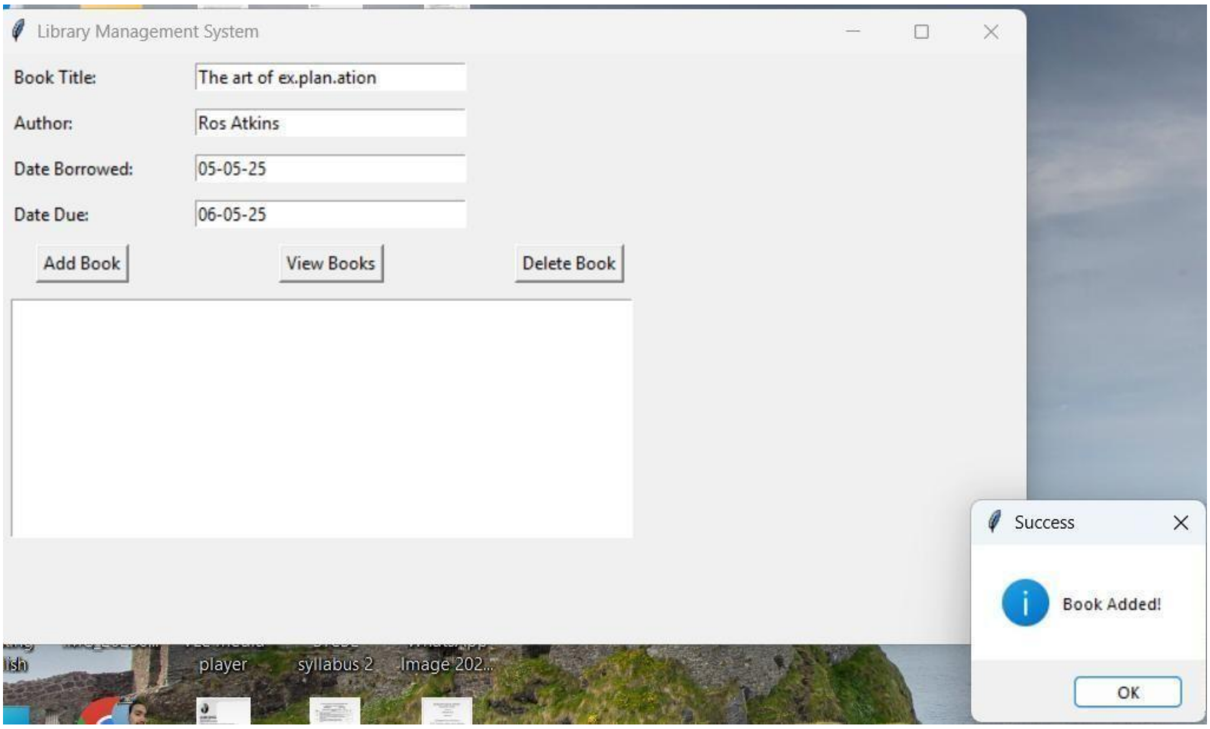


Book Title: The art of ex.plan.ation

Author: Ros Atkins

Date Borrowed: 05-05-25

Date Due: 06-05-25



Library Management System

Book Title: The art of ex.plan.ation

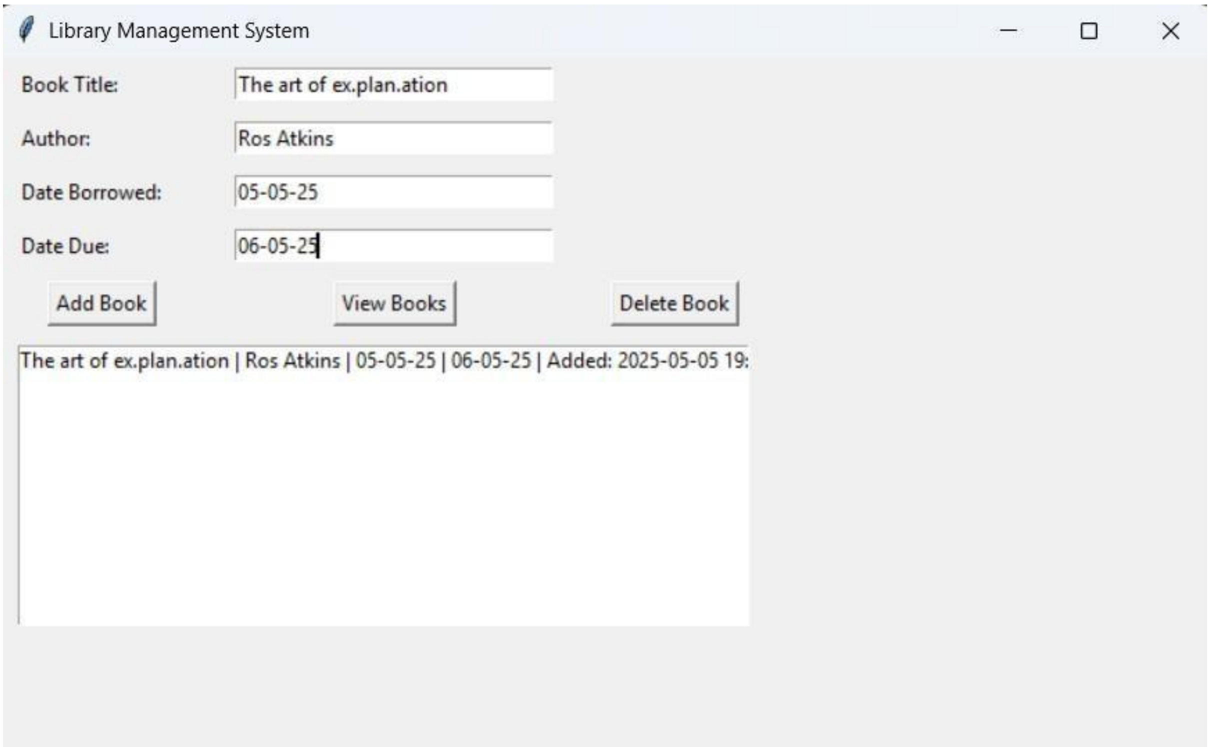
Author: Ros Atkins

Date Borrowed: 05-05-25

Date Due: 06-05-25

Success

Book Added!



Library Management System

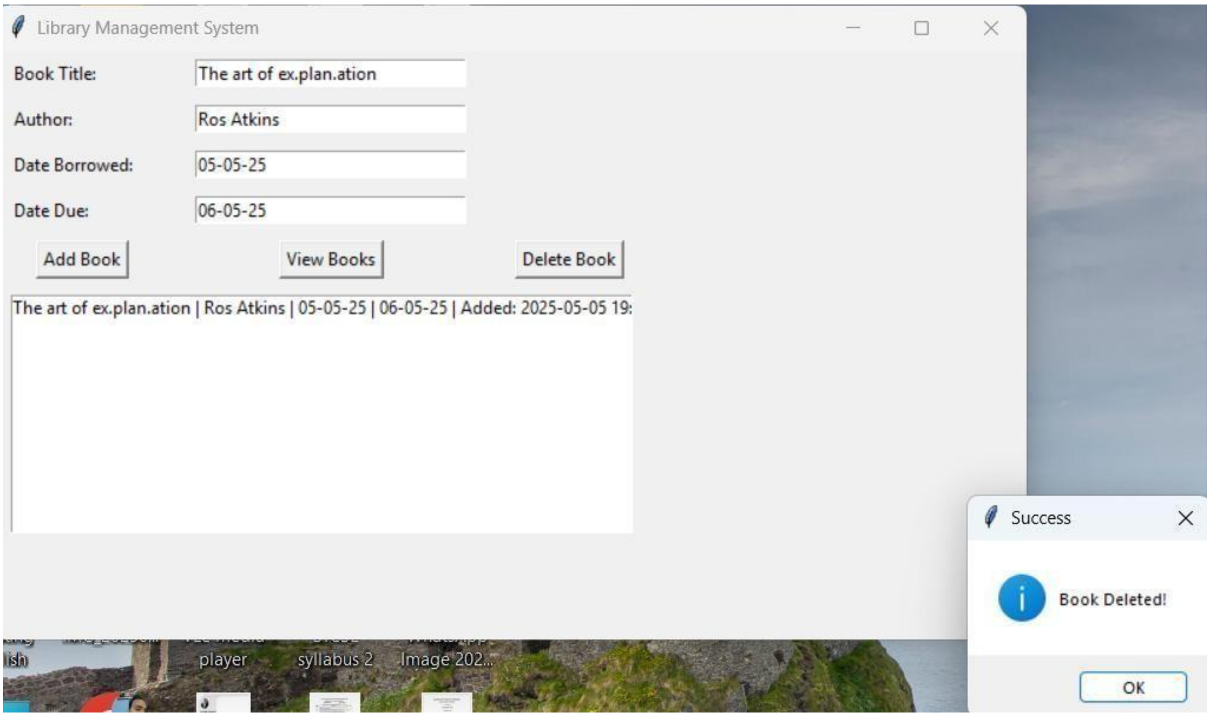
Book Title: The art of ex.plan.ation

Author: Ros Atkins

Date Borrowed: 05-05-25

Date Due: 06-05-25

```
The art of ex.plan.ation | Ros Atkins | 05-05-25 | 06-05-25 | Added: 2025-05-05 19:
```



Library Management System

Book Title: The art of ex.plan.ation

Author: Ros Atkins

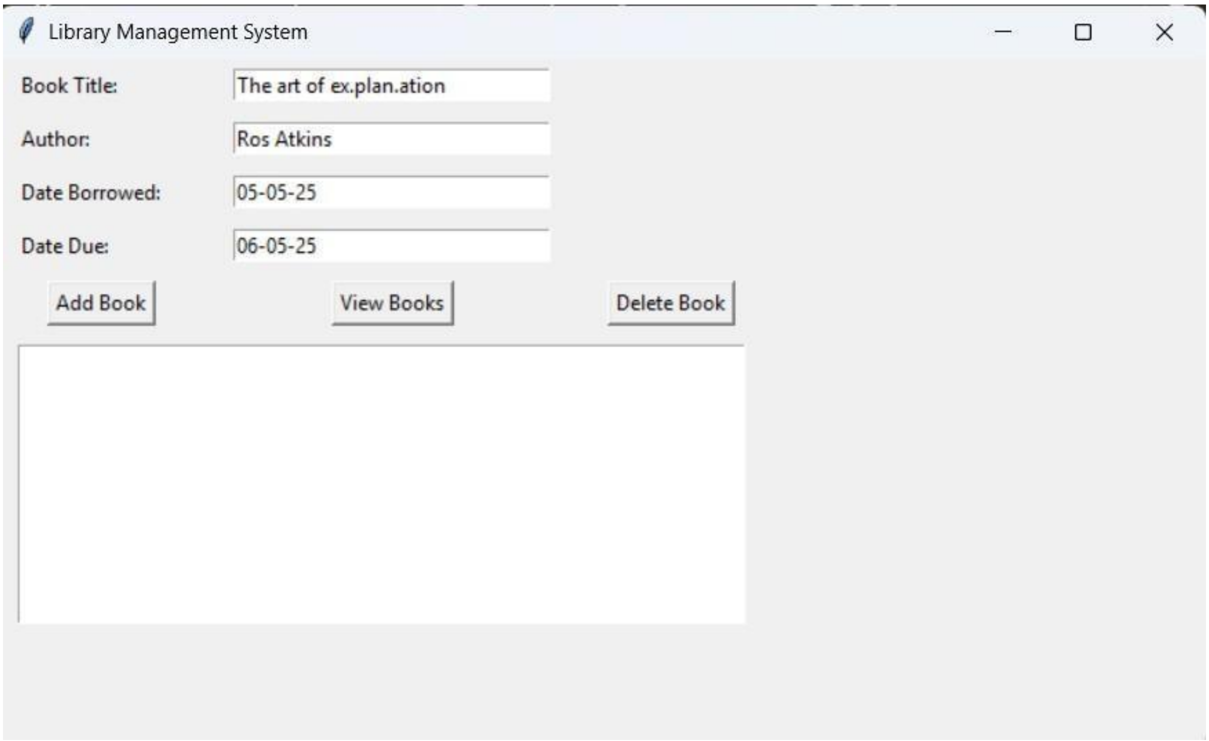
Date Borrowed: 05-05-25

Date Due: 06-05-25

```
The art of ex.plan.ation | Ros Atkins | 05-05-25 | 06-05-25 | Added: 2025-05-05 19:
```

Success

Book Deleted!



Book Title: The art of ex.plan.ation

Author: Ros Atkins

Date Borrowed: 05-05-25

Date Due: 06-05-25

9. Conclusion:

This project successfully demonstrates how Python and SQLite can be combined to develop a simple and efficient Library Management System. It enhances the book tracking process and minimizes the time and effort required to manage a library.

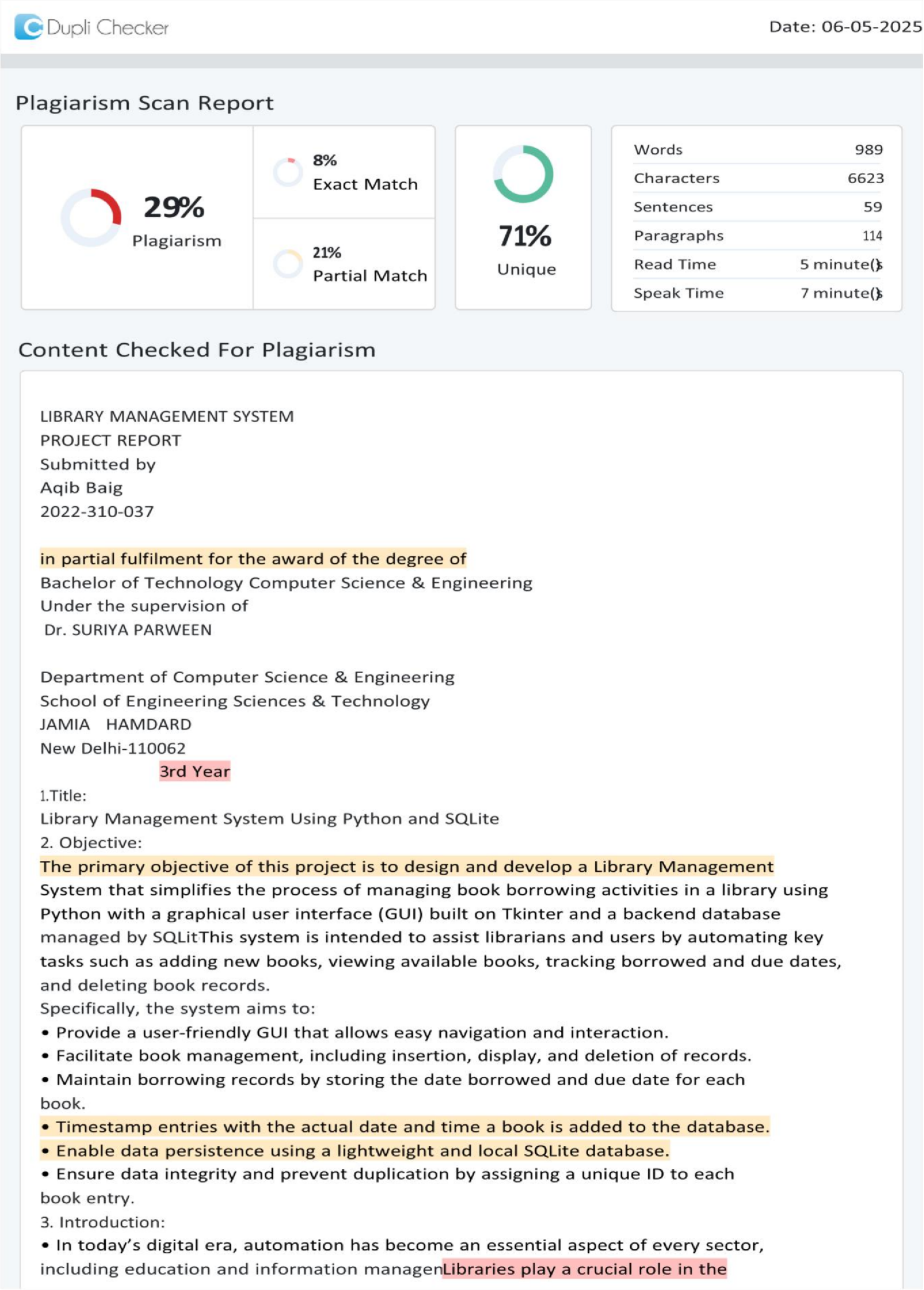
- The system simplifies and speeds up book management for small-scale libraries.
- Provides a user-friendly interface for non-technical users.
- Meets functional requirements with reliable data handling.
- A strong base for extending features like search, user login, or overdue alerts.

10. Limitation:

- Lacks advanced search or filtering options.
- Only desktop access is available (no web version)
- No user authentication or role-based access.
- Cannot handle multiple No user authentication/login system.
- users simultaneously.
- Only basic CRUD operations supported.

11. Bibliography

- Python Official Documentation: https://docs.python.org
- SQLite Documentation: https://sqlite.org/docs.html
- Tkinter Tutorials: https://tkdocs.com/



Dupli Checker

Date: 06-05-2025

Plagiarism Scan Report

29%
Plagiarism

8%
Exact Match

21%
Partial Match

71%
Unique

Words	989
Characters	6623
Sentences	59
Paragraphs	114
Read Time	5 minute(s)
Speak Time	7 minute(s)

Content Checked For Plagiarism

LIBRARY MANAGEMENT SYSTEM
PROJECT REPORT
Submitted by
Aqib Baig
2022-310-037

in partial fulfilment for the award of the degree of
Bachelor of Technology Computer Science & Engineering
Under the supervision of
Dr. SURIYA PARWEEN

Department of Computer Science & Engineering
School of Engineering Sciences & Technology
JAMIA HAMDARD
New Delhi-110062
3rd Year
1.Title:
Library Management System Using Python and SQLite
2. Objective:
The primary objective of this project is to design and develop a Library Management System that simplifies the process of managing book borrowing activities in a library using Python with a graphical user interface (GUI) built on Tkinter and a backend database managed by SQLitThis system is intended to assist librarians and users by automating key tasks such as adding new books, viewing available books, tracking borrowed and due dates, and deleting book records.
Specifically, the system aims to:
• Provide a user-friendly GUI that allows easy navigation and interaction.
• Facilitate book management, including insertion, display, and deletion of records.
• Maintain borrowing records by storing the date borrowed and due date for each book.
• Timestamp entries with the actual date and time a book is added to the database.
• Enable data persistence using a lightweight and local SQLite database.
• Ensure data integrity and prevent duplication by assigning a unique ID to each book entry.
3. Introduction:
• In today's digital era, automation has become an essential aspect of every sector, including education and information managenLibraries play a crucial role in the

academic and personal development of individuals, and managing them efficiently is a task of significant importance. Traditionally, library records have been maintained manually using registers, files, or spreadsheets, which often leads to errors, inefficiencies, and data redundancy. To overcome these limitations, a digital Library Management System is needed.

- This project aims to create a simple yet effective Library Management System using Python, Tkinter, and SQLite. Python is a powerful programming language that supports GUI development through Tkinter, and SQLite is a lightweight, serverless database that stores data in a structured manner. This combination provides a portable, low - maintenance solution ideal for small and medium - sized libraries.
- This system will assist librarians in managing the borrowing and return of books by offering functionalities to add, view, and delete book records. Each record includes vital information such as the book title, author name, date borrowed, and return due date. The records are stored persistently in the SQLite database, ensuring data consistency even when the application is closed or restarted.
- The graphical interface makes the system intuitive and easy to use, eliminating the need for technical expertise. It reduces paperwork, minimizes human error, and improves overall efficiency. Additionally, it introduces features like real - time timestamps for book entries and a visual listbox to display all records dynamically.
- Key features of the system include:
- Add Book Functionality: Users can input and save new book details to the database.
- View Book Records: All added books are viewable in a scrollable listbox.
- Delete Records: Users can delete selected book records directly from the GUI.
- Persistent Storage: All records are stored in a local SQLite database.
- This project not only demonstrates database interaction in Python but also showcases the power of combining frontend (Tkinter) and backend (SQLite) technologies in real world applications. It is a practical example for students and developers interested in learning how to build desktop -based CRUD (Create, Read, Update, Delete) applications. 4. Problem Statement:

In most educational institutions, small community libraries, or personal book collections, the management of books and tracking of borrow/return dates are typically done manually using registers or spreadsheets. This traditional approach presents numerous challenges, including: Human Error: Manual entry of book details increases the chance of inaccuracies such as misspelled titles, incorrect dates, or duplicated records.

Time Consumption: Updating, searching, and maintaining paper -based records is a time consuming process, especially as the collection grows.

Data Loss: Physical logs are prone to damage or loss due to environmental factors, mishandling, or accidental deletion in digital files like Excel.

Lack of Real - Time Access: Manual systems do not provide real - time access or updates, causing delays in book lending or return processes.

Limited Scalability: As the number of users and books grows, the manual system becomes difficult to scale and manage efficiently.

No Search Functionality: Finding specific records or generating reports is tedious and often requires manually going through logs.

Libraries using manual systems face problems such as data loss, redundancy, and difficulty in tracking borrow/return dates.

There is no quick search or organized display of available books.

Borrower information and due dates often get misplaced.

The need for a robust, easy - to - use, and accessible digital system is evident.

6 . Entity Relationship Diagram:

Entities: Book, User (optional)

Attributes of Book: ID, Title, Author, Date Borrowed, Date Due, Added Time Relationships: A Book is added, viewed, or deleted by the system admin

6. Test Case Sample Test Cases:

7. Description Input Values Expected Result

ID

Book added

TC01 Add Book Title, Author, Dates

successfully

TC02 View Books - List of books displayed

Selected book from listbox

TC03 Delete Book(valid ID) Book deleted

Add Book with empty Warning message

TC04 Title or Author missing

fields shown

9. Conclusion

- This project successfully demonstrates how Python and SQLite can be combined to develop a simple and efficient Library Management System. It enhances the book tracking process and minimizes the time and effort required to manage a library.
- The system simplifies and speeds up book management for small-scale libraries .
- • Provides a user-friendly interface for non-technical users.
- Meets functional requirements with reliable data handling. • A strong base for extending features like search, user login, or overdue alerts.
- **10.**

• Lacks advanced search or filtering options.

• Only desktop access is available (no web

- version) No user authentication or role-based access.
- Cannot handle multiple No user authentication/login system. users
- simultaneously.
- Only basic CRUD operations supported.

Matched Source Sep 13, 2024 The objective of the Library Management System (LMS) project is to design and implement an efficient and user-friendly system that automates the... https://www.geeksforgeeks.org/library-management-system Similarity

2%

Title:SQL Insert Timestamp: Quick & Easy Explanation - Five

Jun

Limitation

24, 2024 In SQL, a timestamp is a data type that stores both date and time information. Here are the key timestamp-related data types you'll encounter.Missing: actual | Show results with: https://five.co/blog/sql-insert-timestamp

Similarity 2%

Title:Use a SQLite database in a Windows app - Learn Microsoft

Aug 1, 2024 You can use SQLite to store and retrieve data in a lightweight database on the user's device. This guide shows y ou how to do it in your Windows App SDK apps.Missing: • Enable | Show results with:

https://learn.microsoft.com/e n-us/windows/apps/develop/data-access/sqlite-data-access

Similarity 2%

Title:The role of libraries in promoting published research | For Librarians

Libraries play a crucial role in the creation and dissemination of knowledge and can equip researchers with the necessary ski lls to promote their work effectively. This support is particularly vital for early -career researchers who may not realize they need help.Feb 24, 2025

https://www.springernature.com/gp/librarians/th e-link/researcher-support-blogpost/empoweringresearcher s-libraries-promoting-research/27739900

Similarity 2%

Title:How to Build Library Management System Using NodeJS?

Mar 21, 2025 In this article, we will walk through how to build a simple Library Management System using NodeJS. What We Are Going to Create? We will create...

https://www.geeksforgeeks.org/ho w-to-build-library-management-system-using-node-js

Similarity 2%

Title:Why is Python Considered a High-Level Programming Language?

Jun 13, 2024 · Python is a powerful programming language that can be used for a variety of purposes. With its easy -to-read code lines
and extensive support libraries, Python is an ideal choice for those looking to improve their programming skills.

https://www.xccelerate.co/blog/why-python-a-high-level-programming-language

Similarity 2%

Title:Is it okay to use Sqlite in production? : r/django - Reddit

Nov 15, 2023 SQLite is a lightweight, embedded database that comes bundled with Python and requires minimal setup. It's wellsuited for development and...Missing: GUI Tkinter,

https://www.reddit.com/r/django/comments/17vn181/is_it_okay_to_use_sqlite_in_production

Similarity 2%

Title:3 Best Library Management Systems 2023 | by Lijon - Medium

Oct 7, 2023 Librarika's simplicity and cost-effectiveness make it an attractive option for small to mediumsized libraries, schools, and community...Missing: low- | Show results with: https://medium.com/%40lijonseo/3-best-library-management-systems20239f51ec2b28a3

Similarity 2%

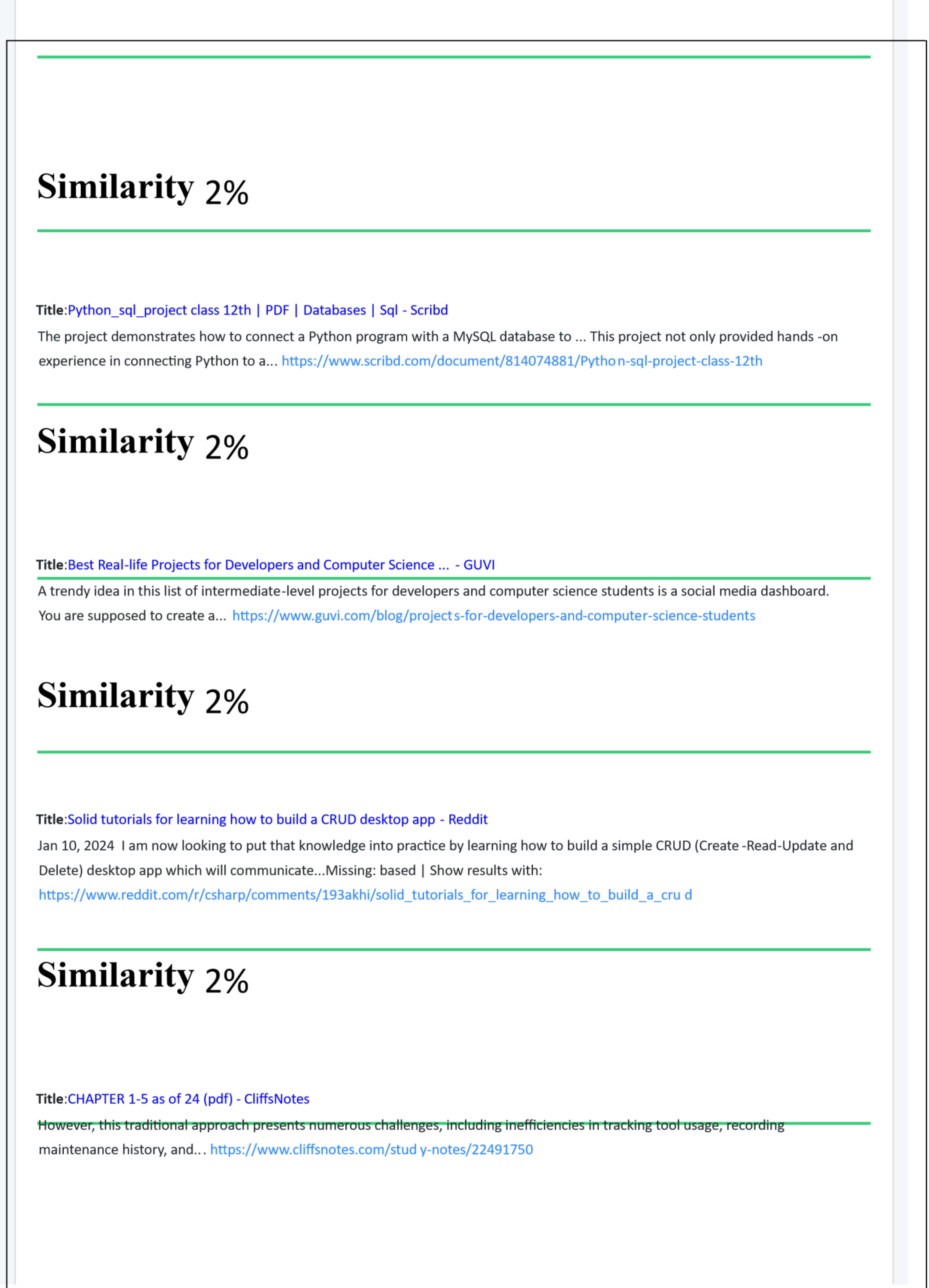
Title:Textbook Loan Program - NYSED / State Aid

Such records should include: information necessary to identify the books such as title, author, publisher and copyright date; and; information to identify the...Missing: vital | Show results with: https://stateaid.nysed.gov/tsl/html_docs/txtbk03.htm

Similarity 2%

Title:Exploring Phreesia Patient Intake System: Benefits & Challenges

Phreesia acts as both a time-saving and cost-effective solution. For healthcare providers, it reduces paperwork, minimizes human error, and provides patients... https://selectraze.com/articles/phreesia-patient-intake-system-benefits-challenges



Similarity 2%

Title:Python_sql_project class 12th | PDF | Databases | Sql - Scribd

The project demonstrates how to connect a Python program with a MySQL database to ... This project not only provided hands -on experience in connecting Python to a... https://www.scribd.com/document/814074881/Pytho n-sql-project-class-12th

Similarity 2%

Title:Best Real-life Projects for Developers and Computer Science ... - GUVI

A trendy idea in this list of intermediate-level projects for developers and computer science students is a social media dashboard. You are supposed to create a... https://www.guvi.com/blog/project s-for-developers-and-computer-science-students

Similarity 2%

Title:Solid tutorials for learning how to build a CRUD desktop app - Reddit

Jan 10, 2024 I am now looking to put that knowledge into practice by learning how to build a simple CRUD (Create -Read-Update and Delete) desktop app which will communicate...Missing: based | Show results with: https://www.reddit.com/r/csharp/comments/193akhi/solid_tutorials_for_learning_how_to_build_a_cru d

Similarity 2%

Title:CHAPTER 1-5 as of 24 (pdf) - CliffsNotes

However, this traditional approach presents numerous challenges, including inefficiencies in tracking tool usage, recording maintenance history, and... https://www.cliffsnotes.com/stud y-notes/22491750