

“There is no programming language, no matter how structured, that will prevent programmers from making bad programs.” Larry Flon

Project Library Management System

by Kovid Joshi
Project Manager

&

Shikar Joshi
QC/QA and Publisher

**DON BOSCO SCHOOL
PITHORAGARH**



Introduction

The LMS Project short for the Library management project is a Program written in `python 3.10.7` language that has a extensive catalog of books that is further extendable to a larger library of books. The Program allows the user to browse the books, by Author, Title, year and ISBN number.

This program is written on `python 3.10.7` version of python and uses custom and built-in libraries from the python. The connection through the MySQL database is possible through the `MySQL-connector-python` or `MySQL-connector` modules downloaded through command line.

1

¹The Python logo belongs to the Python Foundation and is not used for any illicit purpose. Please visit <https://www.python.org/community/logos/>

Acknowledgment

This Project is a combined effect of me and my project partner. We collectively worked and tested the program for bugs and problems, to fix them for the end user. Our collective efforts have made a program that is able work as it claims to be. Further we thank our Teacher for guiding and correcting us. The software we used to make multiple this Project possible have a great contribution. Python development took place over the Jet Brains PyCharm 2022.2.3 that itself is a sufficient IDLE for its task. Further I want to thank my colleague to review this code for any bugs and errors. And Further using the powerful MySQL database system and its software MySQL workbench to complete the query task

Overall this was a interesting and comprehensive task to make such a project and we are grateful to get such a opportunity.

Kovid Joshi (Project Manager)

Contents

Introduction	1
Acknowledgment	2
I System and Feasibility	6
System and Factors of Feasibility	7
System Analysis	7
Terminology	7
ISBN	7
Cataloging	8
Configuration file	8
Feasibility Study	8
Usability Analysis and features	9
II Source and Program Structure	10
Source Code	11
main.py	11
SQL utility	14
Menu and Help	21
About Package lms	23
SQL Commands Imported	23
Users database	23
Books database	26
Data Flow of the Program	28
SQL Database Structure	29
Program Dependency tree	32
III Post Updates	33
Future Updates	34

IV Case Study	35
Login	36
Running	38
Working	42
Case 1: Adding the books to the books database	42
Case 2: Browsing and Exploring the library database	43
Case 3: Using the Different Searches	44
Case 4: using the help	46
Exception Handling	48
Overall Execution	50
Real Output Screen	51
Program Module information	60
Usage of Modules	61
Logging the Actions	63
Software CD	64

List of Figures

1	ISBN number in a barcode	8
2	Dependency of main.py to the lms package	23
3	Data flow diagarm 0 Level	30
4	Directory Structure	32
5	Login in to the Program	52
6	Browsing the books in the catalog	53
7	Exploring the database of Books and Authors	54
8	Searching using a partial name of the book	55
9	Searching using the ISBN of a Book	56
10	Searching the Author's Books in library	57
11	Adding the books to the database	58
12	Unknown value handling by the program	59

Part I

System and Feasibility

System and Factors of Feasibility

System Analysis

Library Management Software is a software that helps a library to catalog and list the books in their library. This Project is based on such a problem to solve some problems regarding –

- Listing the books
- Adding the books
- Searching the books

The extensive catalog of books around the world requires a powerful and efficient database system that is maintained and updated regularly by the developers, one such system is MySQL that is fast and powerful. With Integration with python to make the most out of it, this Project is focused in the connectivity of the two, both python and SQL becomes ideal for a large database like from a library to work fast and responsive. The SQL can handle both very large to extremely large data in a relational form(Tabular form).

A typical Library Management System comprises of

- Cataloging
- Retrieval
- Adding

as one of the core functionality. This program is based on cataloging the data of a library, making the librarian and the users of the library aware of the books inside of the library such that it can be further used for different purposes.

Terminology

ISBN

ISBN stands for the International Standard Book Number is a commercial book identifier intended to be unique². It can be seen in a 10 digit version in old publication and 13 digit in new publication. The ISBN is issued to all the major commercial publication almost all the books have a ISBN. In a book the ISBN looks like a bar-code.

²<https://en.wikipedia.org/wiki/ISBN>



Figure 1: ISBN number in a barcode

Cataloging

library catalog is a register of all bibliographic items found in a library or group of libraries, such as a network of libraries at several locations³. The conventional way of cataloging the books are using the card system. Card Catalog is the a method used for generations, most of the library maintains the card catalog that contains the bibliography of the books. *This program is also made to used as a cataloging program.*

Configuration file

This is a special file that stores the configuration information of the user, like storing details like the user's password, database, user name etc. This file is the file that is used to make the connection to the program that can be edited safely but only changing the field values not the field itself. Here in this program the configuration file which stores the data for the users connection to the MySQL is stored in a special format of file famous for configuration.

Feasibility Study

Feasibility of the program can be divided into

Social The Library Management project is developed taking care of the usability.

Its main objective of this Program is being a usable utility to the Librarians in the world. This program allows the user around the world to maintain their own catalog of book in this system for free. The end user can upgrade and modify the source code according to their requirements and can use it and redistribute it to more audience.

Technical Technically the Program is based on a command line interface and is lightweight. Thanks to python this program is OS independent, that makes the this program to work on any operating system. With some packages and MySQL installed this program must not cause problems while execution.

³https://en.wikipedia.org/wiki/Library_catalog

Technically this program can run on a very low spec machine and can be used only when all the dependencies are installed. The minimum requirements of this program is 2 GB memory, 10 GB storage(persistent) and a operating system of choice with Python and MySQL installed init.

Technically this program is based on cataloging software so display resolution is a matter, a high definition monitor is required for this program to appear intact, because some of the text can be wrapped to another line that can cause the program not so appealing.

Financial This program is based on Open Source Code and is free to use.

Usability Analysis and features

The usability of the program can be described in the following points

- The use of library management system is crucial as it allows the librarian to display and manage the contents of library. Other person who want access to this system can access it by registering it from the website.
- The program is made solid out of python and MySQL. Both of the programs are powerful and secure. MySQL being a very popular database management application is used with the very readable python language.
- Using the Pypi aka pip Library for modules like PyYAML and other bulitins. Python allows the user to configure the MySQL using the configuration file that is a .yaml file which is a popular configuration file. Further modules like os, json, secrets, string, time, random and yaml from the PyYAML package from the Pypi
- In a Library, management plays a crucial role because of the simplicity of the LMS program it allows the user to maintain a clean record that is easily maintained.

Part II

Source and Program Structure

Source Code

main.py

The main file is the integration of all the libraries and is the file that will be executed when running the program

```
1  """
2  main program file
3  """
4
5  import lms.sql_util
6  import lms.menu
7  # import getpass
8
9  i = 0
10
11 # -----Login-----
12 ask_name = None
13
14 while i < 3:
15     ask_name = input("Enter your name ").title().strip()
16     ask_pass = input("Enter your password ")
17     check_data = (ask_name, ask_pass)
18     # -----passwords retrieval
19     if lms.sql_util.pass_checker(check_data) is False:
20         print(" Invalid user, wrong password or name\nplease try
again or register as a new user")
21         i += 1
22         print(f"you have {3 - i} if 3 - i != 0 else exit() } tries")
23         lms.sql_util.logit(message='Login Failed')
24     else:
25         # if the user is found in the database of the users
26         break
27
28
29 lms.sql_util.logit(message='Logged in!')
30
31 # Body of the program
32 lms.menu.menu(user=ask_name)
33 while True:
34     ask_option = input(" ==> ").strip().casefold()
35
36     if ask_option in ['browse', '1']:
37         # display all the isbn details and the books by them
38         lms.sql_util.display(table_name='books')
39         lms.sql_util.logit('displaying the books')
40
41     elif ask_option in ['search', 'find', '2']:
```

```

42 # search options for more exact searching of the books in
43 # the books cataloging
44 search_options = input(""""
45 SEARCH mode
46 search by — ISBN(isbn), author(author) or name(name)
47 -> """).strip().casefold()
48
49 if search_options in ['isbn', '1']:
50     # searching the book using the books ISBN
51     ask_isbn = input("Enter the ISBN number of the book ")
52     # filtering the input so that only numbers get into the
53     # sql input query
54     if ask_isbn.isnumeric():
55         lms.sql_util.search_on_isbn(ask_isbn)
56     else:
57         print("please enter a valid ISBN number")
58         lms.sql_util.logit('searching for a book by its ISBN')
59
60 elif search_options in ['author', '2']:
61     # searching using the author name
62     ask_author = input("Enter the author to search ").title()
63     .strip()
64
65     lms.sql_util.search_on_author(ask_author)
66     lms.sql_util.logit("Searching on the basis of author ")
67
68 elif search_options in ['name', 'book name', 'title', '3']:
69     # searching using the books name
70     ask_title = input("Enter the Title of the book ").strip()
71
72     lms.sql_util.search_on_title(ask_title)
73     lms.sql_util.logit("searching for a book by title")
74
75 elif ask_option in ['add', 'contribute', 'add books']:
76     # adding the books by the user as a contribution
77     print("To Add books you have to verify that it's you!")
78     verify_user = input("Please enter your name ").strip().title
79     ()
80     verify_pass = input("verify your password ")
81     # using the add_books function of the sql_util package to
82     lms.sql_util.add_books((verify_user, verify_pass))
83     lms.sql_util.logit('Adding to the database')
84
85 elif ask_option in ['menu', 'options']:
86     # main menu
87     lms.menu.menu()
88
89 elif ask_option in ['help', 'save me']:
90     # help regarding options

```

```

89         lms.menu.helpme()
90
91     elif ask_option in ['explore', '4']:
92         # explore for the library books
93
94         lms.sql_util.explore()
95
96     elif ask_option in ['exit', 'quit', '5', 'close']:
97         # exiting the program
98         print("Exiting the program")
99         lms.sql_util.logit("Exiting the program ")
100        exit()
101
102    elif ask_option in ['version']:
103        # program version information
104        lms.menu.version()
105
106    else:
107        # for unknown commands
108        print("I don't recognize that need help type help or menu")
109
110 # using the logit function from lms.log
111 # for logging the functions happened in the program

```

main.py

SQL utility

This file is used for the utilities in the SQL database and stores a majority of functions

```
1  """
2  mysql interaction and other main functions for the main.py file.
3  """
4  import os
5  import yaml
6  import mysql.connector
7  import random
8  import secrets
9  import json
10 import time
11 import string
12
13 # -----TABLE NAME-----
14 USER_TABLE = 'lms_users'      # allowed users table
15 BOOKS_TABLE = 'books'        # main library books catalog table
16 DEBUG_TABLE = 'test_books'    # table used temporarily for testing
17 # -----
18
19 def main_cnx(user_id='user'):
20     """
21     function that returns the login connection using the
22     cnx_data.yml file
23     """
24     # changing to the data directory
25     try:
26         if os.path.exists('cnx_data.yml') is False:
27             # os.chdir('.')
28             os.chdir('data')
29         with open('cnx_data.yml') as data_file:
30             data = yaml.load(data_file, yaml.SafeLoader)
31
32         cnx = mysql.connector.connect(**data[user_id])
33         return cnx
34     except FileNotFoundError:
35         # if the data directory is not found in the current directory
36         # print that the
37         print("FATAL ERROR :The directory 'data' does not exists
38         please recover the data directory")
39         exit()
40
41 def pass_checker(user_data):
42     """
43     checking the user input to the registered users
```

```

44     in the database
45     :return: boolean value
46     """
47     # starting the defined connection using the main_cnx() function
48     cnx = main_cnx()
49
50     cursor = cnx.cursor()
51     # executing the command using execute statement
52
53     cursor.execute(f'select * from {USER_TABLE}')
54     # getting the data in the desired form
55     database_data = cursor.fetchall()
56
57     # checking the database from the file data
58     if user_data in database_data:
59         return True
60     else:
61         # return false as the value if the password is wrong
62         return False
63
64
65 def display(table_name='books'):
66     """
67     show the books, isbn author from the database
68     :param table_name:
69     :return:
70     """
71     # initiating the connection
72     cnx = main_cnx()
73     cursor = cnx.cursor()
74
75     # executing the sql statement for the data
76     cursor.execute(f"select * from {table_name}")
77
78     # printing the data form stored in the cursor
79     for lines in cursor:
80         print(f'{lines[0]:14} {lines[1]:45} by {lines[2]}')
81
82
83 def search_on_isbn(isbn_number: str):
84     """
85     searching using the isbn of the book
86     :return:
87     """
88     cnx = main_cnx()
89     cursor = cnx.cursor()
90     if isbn_number.isnumeric():
91         cursor.execute(f"select * from {BOOKS_TABLE} where isbn = {
            isbn_number!r}")

```



```

92         # fetching the data from the database
93         data = cursor.fetchall()
94         # checking for empty data
95         if not data:
96             print(f"Sorry no book is found having ISBN {isbn_number}")
97     )
98     else:
99         # if the book is found print found
100         print('Found')
101         print(f"""
102         ISBN: {data[0][0]}
103         Title: {data[0][1]}
104         Author: {data[0][2]}
105         Published: {data[0][3]} """)
106     else:
107         print("Please enter a number to search")
108
109 def search_on_author(author_name: str):
110     """
111     searching function using the author name
112     :return:
113     """
114
115     cnx = main_cnx()
116     cursor = cnx.cursor()
117     cursor.execute(f"SELECT book_name, published from {BOOKS_TABLE}
118     where author = {author_name!r}")
119     data = cursor.fetchall()
120     # printing the data retrieved from database
121     # listing of the all the books from the author
122     if data:
123         print(f"Books by {author_name}")
124         print(f"Title {'-'*35} Publishing date")
125         for books in data:
126             print(f"{books[0]:40} {books[1]:5}")
127     else:
128         print(f"Author {author_name!r} not found\nPlease check for
129         any typos in the author name and try again")
130
131 def search_on_title(book_name: str):
132     """
133     searching the books in the database using the sql query like
134     functionality
135     :param book_name:
136     :return:
137     """

```

```

137     cnx = main_cnx()
138     cursor = cnx.cursor()
139
140     # executing the query for searching the books database using the
141     # title of the book
142     cursor.execute(f"SELECT book_name, published, author from {
143     BOOKS_TABLE} where book_name like {book_name+'%!r}")
144
145     # get the returned data and store it in the data variable
146     data = cursor.fetchall()
147
148     # if there is data in the variable data
149     if data:
150         print("Found")
151         for books in data:
152             print(f"{books[0]:40} {books[1]}, by {books[2]}")
153
154         return True
155
156     # else if the value is not found give this message
157     else:
158         print(f"Not Found with title {book_name!r}")
159         return False
160
161 def add_books(verify_user):
162     """
163     Adding the books by the user as a contribution to the project
164     database
165     helping it to grow to a more vast book library
166     :param verify_user:
167     :return:
168     """
169     if pass_checker(verify_user) is False:
170         print("Sorry the credentials are wrong")
171     else:
172         cnx = main_cnx()
173         # making the cursor
174         cursor = cnx.cursor()
175         # asking the details of the books by the valid user
176         while True:
177             try:
178                 print("Enter the following details of the book exit
179                 to leave \n")
180                 ask_isbn = input("Enter the isbn number ").strip().
181                 casefold()
182                 if ask_isbn in ['exit', 'quit']:
183                     break
184                 ask_book_name = input("Enter the book name ").strip()

```

```

181         ask_author = input(f"Enter the Author of the book {
ask_book_name!r} ").title().strip()
182         ask_year = input("Enter the year of publishing ")
183         # if no exception occurs break the loop
184         # -----tmp-----##
185         cursor.execute(f"insert into {DEBUG_TABLE} values ({
ask_isbn!r}, {ask_book_name!r}, {ask_author!r}, "
186                             f" {ask_year})")
187         # executing the changes to the table
188         cnx.commit()
189         print("*Successfully* added the book to the library
thanks for the contribution \n"
190                 "help this project to grow.\n")
191
192     except (mysql.connector.errors.DatabaseError, mysql.
connector.errors.InterfaceError):
193         print(f" {'*'*9}SORRY! there was an error, sorry for
the inconvenience {'*'*9}")
194         print(f"{'*'*9}Please enter a number value for the
publishing year{'*'*9}")
195
196
197 def explore():
198     """
199     exploring the data of the LMS database
200     :return:
201     """
202
203     # initiate the connection
204     cnx = main_cnx()
205
206     cursor = cnx.cursor()
207
208     # getting data for the author
209     cursor.execute(f"select author from {BOOKS_TABLE}")
210     author = cursor.fetchall()
211
212     # getting the number of books in the database
213     cursor.execute(f'select count(*) from {BOOKS_TABLE}')
214     times = cursor.fetchall()
215
216     # getting the old books in database
217     cursor.execute(f'select book_name, author from {BOOKS_TABLE}
where published < 2000 ')
218     old = cursor.fetchall()
219
220     # processing the retried values
221     classic_time = random.randint(0, len(old) - 1)
222     random_author = author[random.randint(0, len(author) - 1)][0]

```

```

223 classic_book = old[classic_time][0]
224 classic_author = old[classic_time][1]
225 total_books = times[0][0]
226
227 # printing the result in Command line using the formatted string
228 print(f"""
229 +{'-' * 30}LIBRARY MANAGEMENT SYSTEM{'-' * 30}+
230 |{" " * 85}|
231 |   Read 'By Authors like{" " * 61}|
232 |   {random_author}{" " * (91 - (8 + 1 + len(random_author)))}|
233 |   '~~~~~ Total books in library {total_books} ~~~~~{' "
234 |   ~Time less classics{" " * 63}|
235 |   {classic_book}      by' {classic_author}{" " * (91 - (17+1+len(
236 |   {" " * 85}|
237 +{'-' * 30}{'*' * 25}{'-' * 30}+
238     """)
239
240
241 def logit(message=''):
242     """
243     logging the events happened in the LMS in the separate file
244     called logfile
245     :param message: str
246     :return: number_id -> str
247     """
248
249     # if the file logfile.log does not exist create the new file
250     # named logfile.log
251     if os.path.exists('logfile.log') is False:
252         with open('logfile.log', 'x') as _:
253             pass
254
255     # generating the random number
256     number_id = ''.join(secrets.choice(string.digits) for _ in range
257 (5))
258
259     # making the log data
260     log_data = [time.asctime(time.localtime()), number_id, message]
261
262     # using the json to dump the list into a file and adding the new
263     # line after each dump
264     with open('logfile.log', 'a') as log_file:
265         # dumping the list of the log data to the log file
266         json.dump(log_data, log_file)
267         # adding the new line at the end of the file
268         log_file.write('\n')
269
270     return number_id

```

sql_util.py

Menu and Help

Menu file stores the menus and helps

```
1  """
2  menu, options and help for the file:main.py
3  """
4
5
6  def menu(user=' '):
7      print(f"""
8      +{'-'*60}+
9      |               Library Management System               |
10     | Hi {user}{" "*(65-(1+8+len(user)))}|
11     |   1.Browse books (browse)
12     |   2.Search for the book (find)
13     |   3.Add Books (add)
14     |   4.Explore (explore)
15     |   5.exit (exit)
16     +{'-'*60}+
17     | For help enter help, for version information enter version |
18     +{'-'*60}+
19     """)
20
21
22  def helpme():
23      print("""
24      USER HELP
25
26      *browse*
27      Browse helps the user to browse the extensive catalog of books
28      from
29      the LMS database.
30
31      Search
32      search comprises of the multiple type of search in the books
33      database
34      this options has 3 sub options inside it
35      1.ISBN search
36      2.Author search
37      3.Search by Title of the Book
38
39      *add*
40      Add is a option for people who want to add data to the database
41      for
42      making new books in the library catalog
43
44      *help*
45      gets you here
46      """)
```

```
44     *explore*
45     get the some great recommendations from the some of the best
46     authors
47     and books in the library
48
49     for version type version
50         """
51
52 def version():
53     print("""
54     version information '0.5' 'Bloodymary'
55     """)
```

menu.py

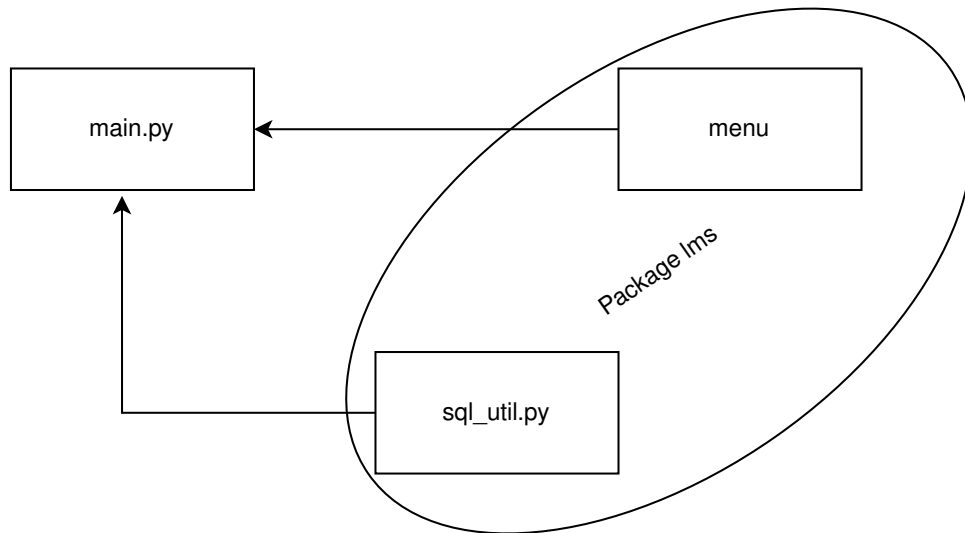


Figure 2: Dependency of main.py to the lms package

About Package lms

The Package `lms` is a custom made package that contains the files for the main execution of the program.

The `lms` package contains files –

1. `sql_util.py`
2. `menu.py`

Both these file contribute to the main file to the core of it. The `menu.py` is the file that contains all the menu, options and help in it. These are crucial for the working of the program. The user can access using the suitable commands supplied by this file.

The `sql_util.py` is a very important file because it contains most of the functions that are required by the `main.py` to work. Further the user's most of the functionality are done by the functions of this file, combining the power of other modules it does the suitable operation for the user such that the user get the desired feedback.

SQL Commands Imported

Users database


```

1  — MySQL dump 10.13  Distrib 8.0.31, for Linux (x86_64)
2  —
3  — Host: localhost    Database: people
4  —
5  — Server version 8.0.31-0ubuntu2
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 /*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;
12 /*!40103 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 'lms_users'
20 —
21
22 DROP TABLE IF EXISTS 'lms_users' ;
23 /*!40101 SET @saved_cs_client      = @@character_set_client */;
24 /*!50503 SET character_set_client = utf8mb4 */;
25 CREATE TABLE 'lms_users' (
26   'name' varchar(50) DEFAULT NULL,
27   'password' varchar(23) DEFAULT NULL
28 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
29 /*!40101 SET character_set_client = @saved_cs_client */;
30
31 —
32 — Dumping data for table 'lms_users'
33 —
34
35 LOCK TABLES 'lms_users' WRITE;
36 /*!40000 ALTER TABLE 'lms_users' DISABLE KEYS */;
37 INSERT INTO 'lms_users' VALUES ('Kate Stewart','kate123'),('Brian
    Smith','123'),('Sam Raimi','supersam'),('Monte Cue','python.org');
38 /*!40000 ALTER TABLE 'lms_users' ENABLE KEYS */;
39 UNLOCK TABLES;
40 /*!40103 SET TIME_ZONE=@OLD_TIME_ZONE */;
41
42 /*!40101 SET SQL_MODE=@OLD_SQL_MODE */;
43 /*!40014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS */;
44 /*!40014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS */;
45 /*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;

```

```
46 /*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;  
47 /*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;  
48 /*!40111 SET SQL_NOTES=@OLD_SQL_NOTES */;  
49  
50 — Dump completed on 2022-11-12 0:07:54
```

users_database.sql

Books database

```
1  — MySQL dump 10.13  Distrib 8.0.31, for Linux (x86_64)
2  —
3  — Host: localhost    Database: people
4  —
5  — Server version 8.0.31-0ubuntu2
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 /*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;
12 /*!40103 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 'books'
20 —
21
22 DROP TABLE IF EXISTS 'books';
23 /*!40101 SET @saved_cs_client      = @@character_set_client */;
24 /*!50503 SET character_set_client = utf8mb4 */;
25 CREATE TABLE 'books' (
26   'isbn' varchar(20) NOT NULL,
27   'book_name' varchar(200) DEFAULT NULL,
28   'author' varchar(40) DEFAULT NULL,
29   'published' int DEFAULT NULL,
30   PRIMARY KEY ('isbn')
31 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
32 /*!40101 SET character_set_client = @saved_cs_client */;
33
34 —
35 — Dumping data for table 'books'
36 —
37
38 LOCK TABLES 'books' WRITE;
39 /*!40000 ALTER TABLE 'books' DISABLE KEYS */;
40 INSERT INTO 'books' VALUES ('0073406732','The Art of Public Speaking,
    11th Edition','Stephen Lucas',2011),('0340951451','It','Stephen
    King',2007),('0393919390','Essentials of Geology (Fourth Edition)',
    'Stephen Marshak',2012),('0451526937','King Lear(Signet Classics)',
    'William Shakespeare',1998),('0553380168','A Brief History of
    Time','Stephen Hawking',1998),('0809063492','KING','Harvard
```

```

    Sitkoff',2009),('1555838537','Stone Butch Blues: A Novel','Leslie
    Feinberg',2004),('1580054838','Fast Times in Palestine','Pamela J.
    Olson',2013),('9780143333623','Grandma\'s Bag of Stories','Sudha
    Murty',2015),('9780385086950','Carrie','Stephen King',1974),('
    9780717260591','The Cat in the Hat','Dr Seuss',1957),('
    9781847490599','Anna Karenina','Leo Tolstoy',1878);
41 /*!40000 ALTER TABLE 'books' ENABLE KEYS */;
42 UNLOCK TABLES;
43 /*!40103 SET TIME_ZONE=@OLD_TIME_ZONE */;
44
45 /*!40101 SET SQL_MODE=@OLD_SQL_MODE */;
46 /*!40014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS */;
47 /*!40014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS */;
48 /*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
49 /*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
50 /*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
51 /*!40111 SET SQL_NOTES=@OLD_SQL_NOTES */;
52
53 — Dump completed on 2022-11-12 0:08:09

```

books_database.sql

Data Flow of the Program

Data flow diagram for the program is –

- Program execution takes multiple steps to reach the final of the program and data is traveled from the python to MySQL as a query or MySQL to python as a result. Further the data execution takes place from the main menu where the user types a certain output to execute a particular function or a query in MySQL to fetch data.
- Program first asks the data for the credentials to put forward the main menu i.e. to login to the program. The user has to enter his credentials to get access to the MySQL database and use the commands in the main menu. Other wise if the credentials are wrong the user has is thrown out of the program after 3 wrong attempt to verify his credentials. Further more the person has to contact the administrator of the MySQL database who manages the users data is to contacted to register to the program user base, this is done to ensure to keep out any unwanted users from using the database.
- After a successful login attempt the user is prompted with the main menu of the program. Further he or she can access the data or add data to the tables of MySQL, and can display using the **explore** command in the program.
- The user has multiple options to choose like search will retrieve the data from the MySQL and display it to the user, add will add the data to the books database here.

The details regarding the database and its structure is given as follows

- This project comprises of multiple data flow model used in System Development Life Cycle. The Project uses a hybrid data flow model that comprises of the
 - Waterfall
 - Circular
- The Circular model is used in the beginning of the program and is generally used for a login screen where a user is looped through a cycle of operations when satisfied in this program enters to the waterfall model where the user is popped with the menu and options to choose from. The options and menu that can be selected by typing it into the command line.

SQL Database Structure

The SQL tables are arranged in the following way such that the tables are accessed using the same database.

There are 2 tables in work with the program.

- lms_users
- books

Both these tables have their own requirements in the program and further in future many others might be added to the database for multiple functionality.

The `lms_user` table is the table that stores the user's database information the basic description of the table is as follows

```
mysql> desc lms_users;
```

Field	Type	Null	Key	Default	Extra
name	varchar(50)	YES		NULL	
password	varchar(23)	YES		NULL	

2 rows in set (0.00 sec)

The `books` database is the table that is used to store all the books and authors in the table. The books and the author information in this table is concise and allows the user to view or add to this table. The simple description from MySQL is as follows

```
mysql> desc books;
```

Field	Type	Null	Key	Default	Extra
isbn	varchar(20)	NO	PRI	NULL	
book_name	varchar(200)	YES		NULL	
author	varchar(40)	YES		NULL	
published	int	YES		NULL	

4 rows in set (0.01 sec)

Further in the testing phase of the program the data is added to the `test_books` table for convenience in collective data integrity of the main table which is `books`.

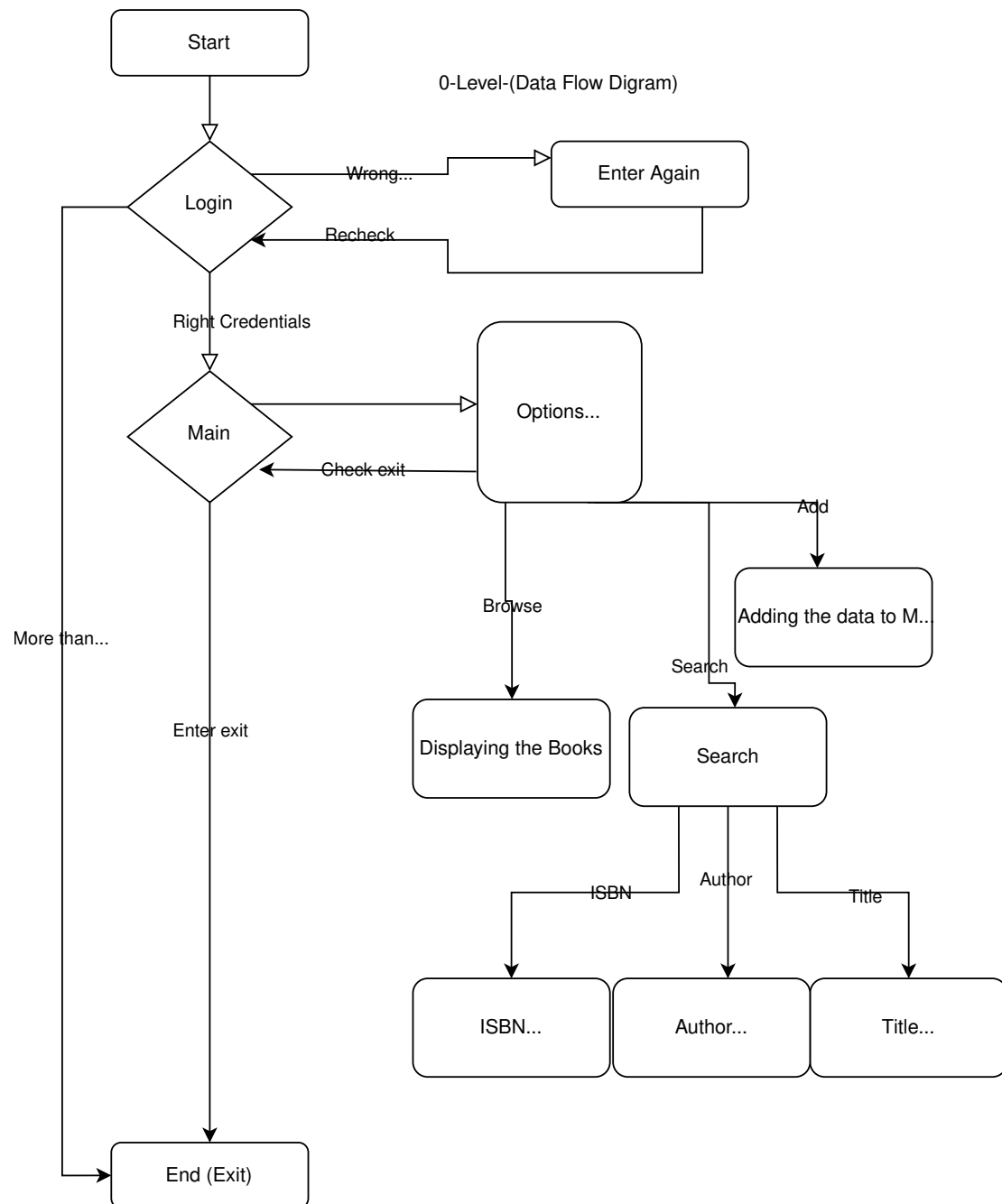


Figure 3: Data flow diagram 0 Level

```
mysql> desc test_books;
```

Field	Type	Null	Key	Default	Extra
isbn	varchar(50)	YES		NULL	
book_name	varchar(100)	YES		NULL	
author	varchar(20)	YES		NULL	
published	int	YES		NULL	

```
4 rows in set (0.00 sec)
```

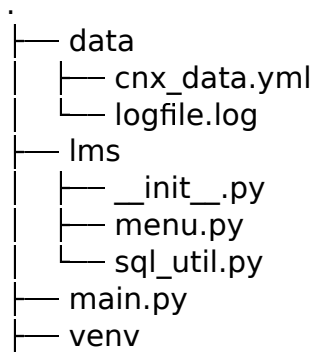



Figure 4: Directory Structure

Program Dependency tree

the tree of the directory structure is given above. The **main.py** file is the main file that executes the program the user has to run this file in order to get the program running. Other than the main file other custom made library directory called as **lms** short hand for library management system is used for further working of the program. another file called as **sql_util.py** and **menu.py** are the files that provide many functionality to the program to work. The **menu.py** file is based on the menu and help, the **sql_util.py** file one of the very important file in the program that allows most of the functions used in the **main.py** file and for further logging function is also provided.

Getting out of the **lms** directory we can see the **data** directory provides and stores the information regarding the configuration and log data. The **cnx_data.yml** stores the configuration data for the MySQL user in MySQL like password, user, database etc can be stored in this file, other file called as **logfile.log** is used to store the log data of the program. The **venv** is a file for the virtual environment that allows us to include packages in a septate environment away from the base interpreters installed packages that might conflict with the other packages or setting up a different environment for the program.

Part III

Post Updates

Future Updates

The following program like the rest of the programs are not prefect. The following program can be improved in feature and security.

- This program is vulnerable to a SQL injection where a hacker can inject a SQL to alter, delete, view and do all sorts of things with the SQL database. The solution of this problem is that the given program takes a filtered input of the things from the users side.
- The program can be made online in cloud, rather than running the SQL locally by setting up a server that can act as a universal server where database can be accessed and data can be retrieved
- Searching using the regular expression can improve the query result and can make it more useful in searching over the words from the database or files.
- further advanced commands to link tables in MySQL can improve the overall functionality of the program and can truly bring the concept of foreign key to work.
- For more security while entering the password the use of `getpass` module that can remove the echo of the password when typed.
- For any further development of this program it can be made out of the command line interface to the graphical user interface. The graphical user interface will reach much more audience and will provide a visual appeal to the program including the benefits of the easier user interface and handling

Part IV

Case Study

Login

The program is allows only specific people to login or use the LMS. This is done to prevent unauthorized access to the database and prevent any unwanted changes in the database. Further the data can be stored (added) by authorized people only. It asks the user password and name. The user is given 3 chances to present the correct user name and password that is stored inside the database itself. The figure below we can see that the name is a case insensitive but the password is sensitive. Further we are greeted by 'Hi' and name in the main screen of the program.

```
Enter your name kate stewart
Enter your password kate123
```

```
+-----+
|           Library Management System           |
| Hi Kate Stewart                               |
|   1.Browse books (browse)                     |
|   2.Search for the book (find)                 |
|   3.Add Books (add)                           |
|   4.Explore (explore)                         |
|   5.exit (exit)                               |
+-----+
| For help enter help, for version information enter version |
+-----+

==>
```

For asking for credentials in the following page user has to provide credentials for further changes in the addition of the books in the books database that require further verification of the user, as being a very sensitive work that can only be modified by the MySQL side. further for wrong credentials the program will not allow the user to add data to the database. □

```
==> add
To Add books you have to verify that it's you!
Please enter your name kate stewart
verify your password kate123
Enter the following details of the book exit to leave
```

Enter the isbn number 90990323134
Enter the book name The Robin Hood
Enter the Author of the book 'The Robin Hood' Helber Osbone
Enter the year of publishing 2001
Successfully added the book to the library thanks for the contribution
help this project to grow.

Enter the following details of the book exit to leave

Enter the isbn number exit
==> add
To Add books you have to verify that it's you!
Please enter your name kate stewart
verify your password alkdjsf
Sorry the credentials are wrong
==>

Running

The program is based on waterfall model there after the Circular model encountered in the login screen where the program asks for the verification of the user.

After the program is started and the user is logged in to the program the user is prompted with a welcome screen or a home page. The user can then select the usable options in the menu and accordingly do the work.

The program menu offers following features in the listing option

1. Browse
2. Search
3. Add
4. Explore
5. Exit

other than these options the user can also access the *help* and *version* options respectively. Also a menu option is available that prints the value of the text listed below.

□

==> menu

```
+-----+
|           Library Management System           |
| Hi                                           |
|   1.Browse books (browse)                   |
|   2.Search for the book (find)               |
|   3.Add Books (add)                         |
|   4.Explore (explore)                       |
|   5.exit (exit)                             |
+-----+
| For help enter help, for version information enter version |
+-----+
==>
```

The menu of the program can be explicitly called, but this time the user name after the word hi is not displayed. Further a person can access the items thereafter by typing it in the prompt below.

The browse – This option provides the user to browse the extensive library of the LMS. It shows the user that ISBN of the book, its title, author name and published date. further this can be used to view or look at the books in the library.

□

==> browse

0073406732	The Art of Public Speaking, 11th Edition	by Stephen Lucas
0340951451	It	by Stephen King
0393919390	Essentials of Geology (Fourth Edition)	by Stephen Marshak
0451526937	King Lear(Signet Classics)	by William Shakespeare
0553380168	A Brief History of Time	by Stephen Hawking
0809063492	KING	by Harvard Sitkoff
1555838537	Stone Butch Blues: A Novel	by Leslie Feinberg
1580054838	Fast Times in Palestine	by Pamela J. Olson
9780143333623	Grandma's Bag of Stories	by Sudha Murty
9780385086950	Carrie	by Stephen King
9780717260591	The Cat in the Hat	by Dr Seuss
9781847490599	Anna Karenina	by Leo Tolstoy

==>

The find – This option is most extensive of all the options and take a good use of the powerful MySQL system using the connector to connect to the MySQL database. Further it allows the user to find the book in the database using either its title, author or ISBN.

□

==> find

SEARCH mode

search by -- ISBN(isbn), author(author) or name(name)

->

The user can then search on the basis of the ISBN, author or name of the book that he or she wants to find. Then he can type the rest and make the program to find the book in the database.

ISBN SEARCHING

ISBN stands for International Standard Book number. This number is issued to books, journals, articles and magazines. unique number is used to identify a book either in a book store or in a library. The ISBN number can be categorized in 10 digits or 13 digits, The program is made in such a way that it allows both the formats to work in its environment. □


```
==> find
```

```
SEARCH mode
```

```
search by -- ISBN(isbn), author(author) or name(name)
```

```
-> isbn
```

```
Enter the ISBN number of the book 0809063492
```

```
Found
```

```
ISBN: 0809063492
```

```
Title: KING
```

```
Author: Harvard Sitkoff
```

```
Published: 2009
```

```
==>
```

AUTHOR SEARCH

This option is used to find the books by a particular author in the library database. The user has to provide the author's name and the database fetches the result of all the books that belongs to the author. Here also the program uses the powerful techniques to integrate the MySQL to get the result of the desired query. Asking user for the author name either in capital or small as the program turns the string into a title case and then asks for the query, the user is then given a response of the title and publication date of the book by the author. □

```
==> find
```

```
SEARCH mode
```

```
search by -- ISBN(isbn), author(author) or name(name)
```

```
-> author
```

```
Enter the author to search stephen king
```

```
Books by Stephen King
```

```
Title -----Publishing date
```

```
It 2007
```

```
Carrie 1974
```

```
==>
```

TITLE SEARCH

This search is used to search details of a particular books title it is used for further finding the books whose only partial titles were like only a part of title is known. Here the source code utilizes the potential of MySQL where "LIKE" keyword is used. This allows the user to enter the first matching characters and then the return result is based on the result found by the program. □

==> find

SEARCH mode

search by -- ISBN(isbn), author(author) or name(name)

-> name

Enter the Title of the book the

Found

The Art of Public Speaking, 11th Edition 2011, by Stephen Lucas

The Cat in the Hat 1957, by Dr Seuss

==>

The query is case insensitive i.e. the result is based on characters not on the case of the letters. This type of search is very useful and allows the user to search the database much usefully.

□

==> find

SEARCH mode

search by -- ISBN(isbn), author(author) or name(name)

-> name

Enter the Title of the book fast tim

Found

Fast Times in Palestine 2013, by Pamela J. Olson

==>

Working

Case 1: Adding the books to the books database

The data can be added to the books database using the LMS program. By selecting the add option in the main menu a person will get into the add menu of the program. Then the user has to enter his or her credentials to verify that it is him who is adding to the program. After conforming the credentials the person can add the data to the MySQL by answering the questions regarding the new book. The addition command is made on loop so *a person can add multiple books without getting out of the program* □

Enter your name kate stewart

Enter your password kate123

```
+-----+
|               Library Management System               |
| Hi Kate Stewart                                       |
|   1.Browse books (browse)                             |
|   2.Search for the book (find)                         |
|   3.Add Books (add)                                    |
|   4.Explore (explore)                                 |
|   5.exit (exit)                                       |
+-----+
| For help enter help, for version information enter version |
+-----+
```

==> add

To Add books you have to verify that it's you!

Please enter your name kate stewart

verify your password kate123

Enter the following details of the book exit to leave

Enter the isbn number 90990323134

Enter the book name The Robin Hood

Enter the Author of the book 'The Robin Hood' Helber Osbone

Enter the year of publishing 2001

Successfully added the book to the library thanks for the contribution
help this project to grow.

Enter the following details of the book exit to leave

```
Enter the isbn number exit
==>
```

Case 2: Browsing and Exploring the library database

The LMS program is made for better provide a friendly user experience for showing the books from the library by picking the author and displaying it to the user. A user can type explore or browse to see books from the library

□

```
Enter your name kate stewart
Enter your password kate123
```

```
+-----+
|           Library Management System           |
| Hi Kate Stewart                               |
|   1.Browse books (browse)                     |
|   2.Search for the book (find)                 |
|   3.Add Books (add)                           |
|   4.Explore (explore)                         |
|   5.exit (exit)                               |
+-----+
| For help enter help, for version information enter version |
+-----+
```

```
==> browse
0073406732    The Art of Public Speaking, 11th Edition    by Stephen Lucas
0340951451    It                                           by Stephen King
0393919390    Essentials of Geology (Fourth Edition)      by Stephen Marshak
0451526937    King Lear(Signet Classics)                 by William Shakespeare
0553380168    A Brief History of Time                    by Stephen Hawking
0809063492    KING                                         by Harvard Sitkoff
1555838537    Stone Butch Blues: A Novel                 by Leslie Feinberg
1580054838    Fast Times in Palestine                    by Pamela J. Olson
9780143333623 Grandma's Bag of Stories                    by Sudha Murty
9780385086950 Carrie                                       by Stephen King
9780717260591 The Cat in the Hat                         by Dr Seuss
9781847490599 Anna Karenina                              by Leo Tolstoy
==> explore
```

```

+-----LIBRARY MANAGEMENT SYSTEM-----+
|
|   Read `By Authors like
|   Dr Seuss
|   `~~~~~ Total books in library 12 ~~~~~
|   ~Time less classics
|   A Brief History of Time      by' Stephen Hawking
|
+-----*****-----+

```

==> find

```

SEARCH mode
search by -- ISBN(isbn), author(author) or name(name)
-> author
Enter the author to search Stephen king
Books by Stephen King
Title -----Publishing date
It                      2007
Carrie                   1974
==>

```

The exploring and search options are extensive and allows the person to search and explore⁴ the database of the library.

Case 3: Using the Different Searches

The program allows the user to search in multiple ways in the SQL database. The available options are

1. Search using the ISBN
2. Search using the Author name
3. Search using the Title of the Book

Using the find command in the program a person can access the database for search using any of the following above options. The different search options come in handy in case the person is partially aware of the Book.

⁴The explore option is altered here for the sake of output and differs from the real output in the main file

```

+-----+
|           Library Management System           |
| Hi Kate Stewart                             |
|   1.Browse books (browse)                   |
|   2.Search for the book (find)              |
|   3.Add Books (add)                         |
|   4.Explore (explore)                       |
|   5.exit (exit)                             |
+-----+
| For help enter help, for version information enter version |
+-----+

```

==> find

```

SEARCH mode
search by -- ISBN(isbn), author(author) or name(name)
-> isbn

```

Enter the ISBN number of the book 9781847490599
Found

```

ISBN: 9781847490599
Title: Anna Karenina
Author: Leo Tolstoy
Published: 1878

```

==> find

```

SEARCH mode
search by -- ISBN(isbn), author(author) or name(name)
-> author

```

Enter the author to search Dr Seuss
Books by Dr Seuss

Title -----	Publishing date
The Cat in the Hat	1957

==> find

```

SEARCH mode
search by -- ISBN(isbn), author(author) or name(name)
-> name

```

Enter the Title of the book the
Found

The Art of Public Speaking, 11th Edition 2011, by Stephen Lucas
The Cat in the Hat 1957, by Dr Seuss
==>

Case 4: using the help

Help is a very important command used by anyone using the program. The help command is what is used to display the help in the program.

Enter your name kate stewart
Enter your password kate123

```
+-----+
|           Library Management System           |
| Hi Kate Stewart                               |
|   1.Browse books (browse)                     |
|   2.Search for the book (find)                 |
|   3.Add Books (add)                           |
|   4.Explore (explore)                         |
|   5.exit (exit)                               |
+-----+
| For help enter help, for version information enter version |
+-----+
```

==> help

USER HELP

browse

Browse helps the user to browse the extensive catalog of books from the LMS database.

Search

search comprises of the multiple type of search in the books database this options has 3 sub options inside it

- 1.ISBN search
- 2.Author search
- 3.Search by Title of the Book

`*add*`

Add is a option for people who want to add data to the database for making new books in the library catalog

`*help*`

gets you here

`*explore*`

get the some great recommendations from the some of the best authors and books in the library

for version type version

`==>`

Exception Handling

For every wrong command the program tells the user to write a better command rather than it already is

□

```
+-----+
|           Library Management System           |
| Hi                                           |
|   1.Browse books (browse)                   |
|   2.Search for the book (find)              |
|   3.Add Books (add)                         |
|   4.Explore (explore)                      |
|   5.exit (exit)                            |
+-----+
| For help enter help, for version information enter version |
+-----+
```

```
==> exi
```

```
I don't recognize that need help type help or menu
```

```
==> exit
```

```
Exiting the program
```

for any option in the user side any exceptions are either handled using the `try` and `except` block and to prevent any input error the use of numeric datatype is limited. The inputs are taken in the string form to minimize the data handling error and is type caste to `int` or other format using the suitable function.

Further the program is giving messages for exception that occurs while the program is running to prevent crash and to ensure the smooth functioning of the program.

□

```
Enter your name david bechem
```

```
Enter your password beck123
```

```
Invalid user, wrong password or name
```

```
please try again or register as a new user
```

```
you have 2 tries
```

```
Enter your name
```

above shows the message that is shown in the file when the user is entering a wrong password or name, the program gives him or her three chances to correctly write the input.

□

==> find

SEARCH mode

search by -- ISBN(isbn), author(author) or name(name)

-> author

Enter the author to search jack

Author 'Jack' not found

Please check for any typos in the author name and try again

==>

The basic search also gives an error if not found, when the user enters a author which does not exists then the program tells the user to check for any spelling mistakes and try again. The add command is a sensitive option that requires a lot of control over the program input from the user in case where the user has entered the wrong thing in the date option the user is prompted with the error message. Also if that field is left blank then also, the user is prompted with the message to prevent the further execution of the program to cause error on the MySQL server side.

□

==> add

To Add books you have to verify that it's you!

Please enter your name kate stewart

verify your password kate123

Enter the following details of the book exit to leave

Enter the isbn number 99021314

Enter the book name

Enter the Author of the book ''

Enter the year of publishing

*****SORRY! there was an error, sorry for the inconvenience *****

*****Please enter a number value for the publishing year*****

Enter the following details of the book exit to leave

Overall Execution

A look at the general execution of the program⁵ □

Enter your name kate stewart

Enter your password kate123

```
+-----+
|               Library Management System               |
| Hi Kate Stewart                                       |
|   1.Browse books (browse)                             |
|   2.Search for the book (find)                         |
|   3.Add Books (add)                                    |
|   4.Explore (explore)                                  |
|   5.exit (exit)                                        |
+-----+
| For help enter help, for version information enter version |
+-----+
```

==> browse

0073406732	The Art of Public Speaking, 11th Edition	by Stephen Lucas
0340951451	It	by Stephen King
0393919390	Essentials of Geology (Fourth Edition)	by Stephen Marshak
0451526937	King Lear(Signet Classics)	by William Shakespeare
0553380168	A Brief History of Time	by Stephen Hawking
0809063492	KING	by Harvard Sitkoff
1555838537	Stone Butch Blues: A Novel	by Leslie Feinberg
1580054838	Fast Times in Palestine	by Pamela J. Olson
9780143333623	Grandma's Bag of Stories	by Sudha Murty
9780385086950	Carrie	by Stephen King
9780717260591	The Cat in the Hat	by Dr Seuss
9781847490599	Anna Karenina	by Leo Tolstoy

==> find

SEARCH mode

search by -- ISBN(isbn), author(author) or name(name)

-> author

Enter the author to search stephen king

Books by Stephen King

⁵The output of the program here is changed to fit the typesetting the document. The out put is modified

Title -----	Publishing date
It	2007
Carrie	1974

==> add

To Add books you have to verify that it's you!

Please enter your name kate stewar

verify your password kate

Sorry the credentials are wrong

==> explore

```

+-----LIBRARY MANAGEMENT SYSTEM-----+
|
|   Read `By Authors like
|   Sudha Murty
|   ~~~~~ Total books in library 12 ~~~~~
|   ~Time less classics
|   A Brief History of Time      by' Stephen Hawking
|
+-----*****-----+

```

==> find

SEARCH mode

search by -- ISBN(isbn), author(author) or name(name)

-> name

Enter the Title of the book grandma

Found

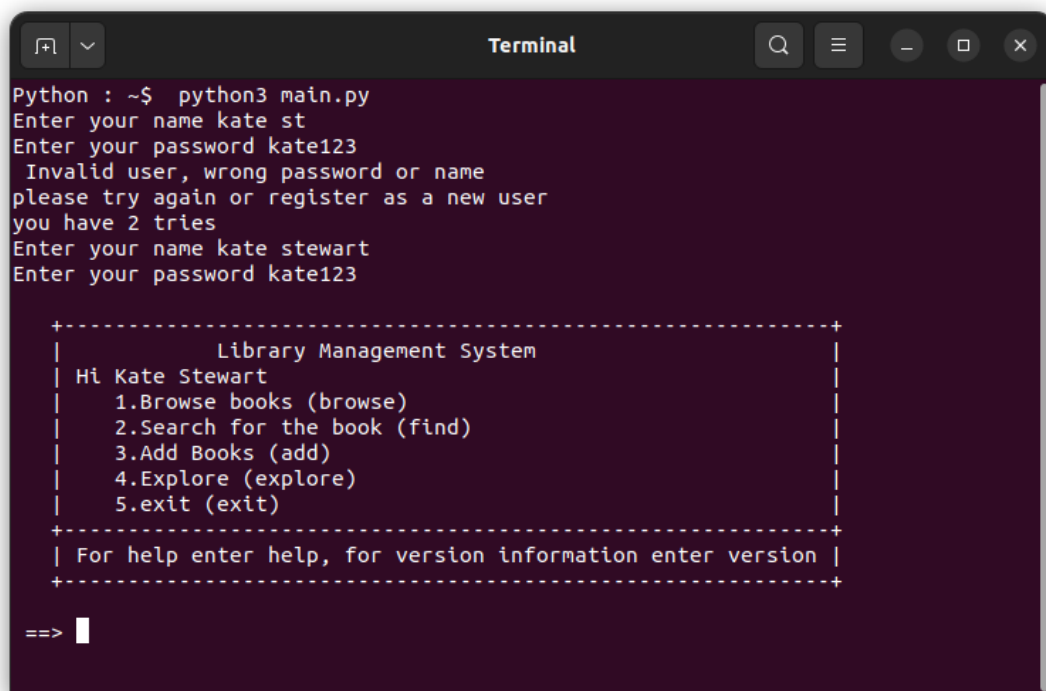
Grandma's Bag of Stories

2015, by Sudha Murty

==> exit

Exiting the program

Real Output Screen



```
Python : ~$ python3 main.py
Enter your name kate st
Enter your password kate123
Invalid user, wrong password or name
please try again or register as a new user
you have 2 tries
Enter your name kate stewart
Enter your password kate123

+-----+
|           Library Management System           |
| Hi Kate Stewart                               |
| 1.Browse books (browse)                       |
| 2.Search for the book (find)                  |
| 3.Add Books (add)                             |
| 4.Explore (explore)                           |
| 5.exit (exit)                                 |
+-----+
| For help enter help, for version information enter version |
+-----+

==> 
```

Figure 5: Login in to the Program

```
Terminal
Invalid user, wrong password or name
please try again or register as a new user
you have 2 tries
Enter your name kate stewart
Enter your password kate123

+-----+
|               Library Management System               |
| Hi Kate Stewart                                       |
| 1.Browse books (browse)                               |
| 2.Search for the book (find)                         |
| 3.Add Books (add)                                    |
| 4.Explore (explore)                                 |
| 5.exit (exit)                                        |
+-----+
| For help enter help, for version information enter version |
+-----+

==> browse
0073406732 The Art of Public Speaking, 11th Edition by Stephen Lucas
0340951451 It by Stephen King
0393919390 Essentials of Geology (Fourth Edition) by Stephen Marshak
0451526937 King Lear(Signet Classics) by William Shakespea
re
0553380168 A Brief History of Time by Stephen Hawking
0809063492 KING by Harvard Sitkoff
1555838537 Stone Butch Blues: A Novel by Leslie Feinberg
1580054838 Fast Times in Palestine by Pamela J. Olson
9780143333623 Grandma's Bag of Stories by Sudha Murty
9780385086950 Carrie by Stephen King
9780717260591 The Cat in the Hat by Dr Seuss
9781847490599 Anna Karenina by Leo Tolstoy
==> 
```

Figure 6: Browsing the books in the catalog

```
Python : ~$ python3 main.py
Enter your name kate stewart
Enter your password kate123

+-----+
|               Library Management System               |
| Hi Kate Stewart                                       |
| 1.Browse books (browse)                               |
| 2.Search for the book (find)                         |
| 3.Add Books (add)                                    |
| 4.Explore (explore)                                 |
| 5.exit (exit)                                        |
+-----+
| For help enter help, for version information enter version |
+-----+

==> explore

+-----LIBRARY MANAGEMENT SYSTEM-----+
| Read `By Authors like                               |
| Stephen Hawking                                     |
| ~~~~~~ Total books in library 12 ~~~~~~             |
| ~Time less classics                                |
| King Lear(Signet Classics)      by' William Shakespeare |
+-----+*****+

==> 
```

Figure 7: Exploring the database of Books and Authors

```
Python : ~$ python3 main.py
Enter your name kate stewart
Enter your password kate123

+-----+
|               Library Management System               |
| Hi Kate Stewart                                       |
| 1.Browse books (browse)                               |
| 2.Search for the book (find)                         |
| 3.Add Books (add)                                    |
| 4.Explore (explore)                                 |
| 5.exit (exit)                                        |
+-----+
| For help enter help, for version information enter version |
+-----+

==> find

      SEARCH mode
      search by -- ISBN(isbn), author(author) or name(name)
      -> name
Enter the Title of the book anna
Found
Anna Karenina                                1878, by Leo Tolstoy
==> 
```

Figure 8: Searching using a partial name of the book


```
Python : ~$ python3 main.py
Enter your name kate stewart
Enter your password kate123

+-----+
|               Library Management System               |
| Hi Kate Stewart                                       |
| 1.Browse books (browse)                               |
| 2.Search for the book (find)                         |
| 3.Add Books (add)                                    |
| 4.Explore (explore)                                 |
| 5.exit (exit)                                       |
+-----+
| For help enter help, for version information enter version |
+-----+

==> find

SEARCH mode
search by -- ISBN(isbn), author(author) or name(name)
-> isbn
Enter the ISBN number of the book 9780385086950
Found

ISBN: 9780385086950
Title: Carrie
Author: Stephen King
Published: 1974

==>
```

Figure 9: Searching using the ISBN of a Book

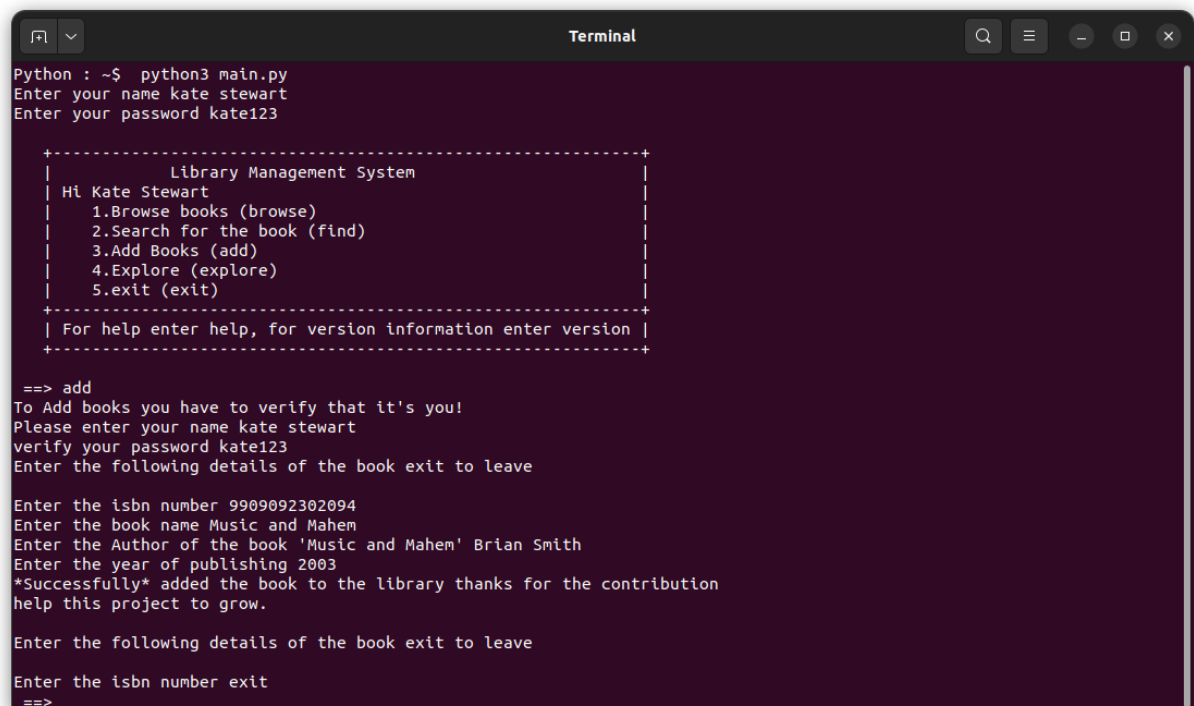
```
Python : ~$ python3 main.py
Enter your name kate stewart
Enter your password kate123

+-----+
|               Library Management System               |
| Hi Kate Stewart                                       |
| 1.Browse books (browse)                               |
| 2.Search for the book (find)                         |
| 3.Add Books (add)                                    |
| 4.Explore (explore)                                 |
| 5.exit (exit)                                        |
+-----+
| For help enter help, for version information enter version |
+-----+

==> find

SEARCH mode
search by -- ISBN(isbn), author(author) or name(name)
-> author
Enter the author to search stephen king
Books by Stephen King
Title -----Publishing date
It 2007
Carrie 1974
==> 
```

Figure 10: Searching the Author's Books in library



```
Python : ~$ python3 main.py
Enter your name kate stewart
Enter your password kate123

+-----+
|               Library Management System               |
| Hi Kate Stewart                                       |
| 1.Browse books (browse)                               |
| 2.Search for the book (find)                         |
| 3.Add Books (add)                                    |
| 4.Explore (explore)                                 |
| 5.exit (exit)                                        |
+-----+
| For help enter help, for version information enter version |
+-----+

==> add
To Add books you have to verify that it's you!
Please enter your name kate stewart
verify your password kate123
Enter the following details of the book exit to leave

Enter the isbn number 9909092302094
Enter the book name Music and Mahem
Enter the Author of the book 'Music and Mahem' Brian Smith
Enter the year of publishing 2003
*Successfully* added the book to the library thanks for the contribution
help this project to grow.

Enter the following details of the book exit to leave

Enter the isbn number exit
==>
```

Figure 11: Adding the books to the database

```
Python : ~$ python3 main.py
Enter your name kate stewart
Enter your password kate123

+-----+
|               Library Management System               |
| Hi Kate Stewart                                       |
| 1.Browse books (browse)                               |
| 2.Search for the book (find)                         |
| 3.Add Books (add)                                    |
| 4.Explore (explore)                                 |
| 5.exit (exit)                                        |
+-----+
| For help enter help, for version information enter version |
+-----+

==> exit
I don't recognize that need help type help or menu
==> show
I don't recognize that need help type help or menu
==> find

SEARCH mode
search by -- ISBN(isbn), author(author) or name(name)
-> author
Enter the author to search 241
Author '241' not found
Please check for any typos in the author name and try again
==> add
To Add books you have to verify that it's you!
Please enter your name kate stewart
verify your password kate
Sorry the credentials are wrong
==> 
```

Figure 12: Unknown value handling by the program

Program Module information

The LMS program consists of many imported modules, most of them found in the local builtin library of python by some can be installed using the Python package index website or pip a common package manager in the command line⁶. There are following modules used in the program.

1. `os`
2. `json`
3. `secrets`
4. `string`
5. `random`
6. `time`
7. `mysql-connector-python` or `mysql-connector`
8. `yaml`

The modules `os`, `json`, `secrets`, `random` & `time` are builtins and come preinstalled in the python, rest the other 2 modules `yaml`, MySQL connector `mysql-connector-python`⁷ and `mysql-connector` are installed using pip. For pip to work following conditions should be met –

- *pip must be up-to-date.* The pip does not work when it is lower version and does not install packages in such a condition.
- *Path configured python installation.* While installing the python programming language using the installer in windows the user must select the add to path checkbox in the installer, if not selected the pip will not be installed to the path and will not work.

The following modules `yaml`, `mysql-connector` can be installed using the following commands.

⁶Python Software Foundation recommends using pip for installing the packages

⁷for updated 8+ version of Mysql

```
pip install PyYAML
```

```
pip install mysql-connector-python or pip install mysql-connector
```

Usage of Modules

Modules are used to increase the functionality of the program. They as a helping hand to the programmer by proving him more functions to work with and to make the program feature rich. The modules imported in this Project are of critical importance and they provide much more functionality to the programmer and the end user. The usage of the modules in the Project is mentioned below –

os This module is used checking the file's existence and other directory related operations. The functions like `os.getcwd()` for getting the current working directory of the program when executed, to change the directory the `os.chdir()` is used to change to the desired directory. Further for checking the existence of the files in the current working directory.

json JSON⁸ is a file format for interchanging data, this module helps us the serialize the list generated by the log function. It is used to dump the list in the file such that it is directly put to the file. In this program functions like `json.dump(list)` for any list is used for dumping the data in the serialized format. Also it was preferred than another builtin module `pickle` because the pickle module handles the data in the binary but the log file is meant to be human readable.

secrets This module is made for securely doing the random bytes or making random choices, this module is just like the `random` module but more secure. By secure it means that the randomness in this module is preferred over the `random` module in cases like, random choice maker, random byte generator, etc. In this project this module provides functionality to the log generator, to make a random number for uniquely identifying a particular log in the log file.

random Random module is used for making random number, not as secure as module `secrets` but helps in generating random numbers, integers, making choices etc. Here it is used in function for exploring the Library catalog by randomly selecting the author and book from the Library database accordingly using the list indexing and suitable SQL query.

time Time is used to get the current time in the program. The function `time.localtime()` is used to get the current time and is placed inside of the `time.asctime()`

⁸Java Script Object Notation

to get the formatted current time used by the log function for printing the timing of the happening of the event.

string it is a very simple module providing with the strings of data like 123...or abcd...and symbols in both lowercase, uppercase or both cases mixed. This module's `string.digits()` for list of digits

mysql.connector This is a module that helps us to make our connection with the MySQL database by giving our credentials like user name, user's password, database to work with etc. Using the cursor we can initiate the query using the `cursor.execute(query)` command where `cursor` is a method of the connection. Using this we can execute multiple queries. Further `cursor.fetchall()` is used to get the result in a variable in the form of nested list of the given query by the programmer.

yaml YAML is a data serializing language used for configuration files in a program. In this program a very similar function `yaml.load()` is used to load the data from the YAML file to the python object. It returns a dictionary and is used for storing credentials for the user to load from. The user can configure the file accordingly to change the password of the database or the username or the database itself and unlike JSON⁹ it is extremely human readable.

Further more modules can be used to make it more functionality rich

⁹JSON is human readable but YAML has better human readability.

Logging the Actions

The program is also made with another feature other than all listed above to make a log of the actions that happen in the program by the user. The format of the log file is custom where it stores different like given below is a real log file from the programs directory

□

```
["Sun Nov 6 19:40:29 2022", "8 2 2 9 1", "Logged in!"]
["Sun Nov 6 19:41:31 2022", "7 6 3 1 2", "searching for a book by its ISBN"]
["Sun Nov 6 19:41:53 2022", "2 4 6 5 6", "Searching on the basis of author "]
["Sun Nov 6 19:42:10 2022", "4 8 1 6 3", "Exiting the program "]
["Mon Nov 7 16:18:28 2022", "2 5 3 5 7", "Logged in!"]
["Mon Nov 7 16:40:23 2022", "7 6 5 0 0", "Adding to the database"]
["Mon Nov 7 16:40:37 2022", "6 4 9 4 1", "Adding to the database"]
["Mon Nov 7 17:06:46 2022", "0 5 1 2 9", "displaying the books"]
["Mon Nov 7 17:20:55 2022", "1 3 9 3 7", "searching for a book by its ISBN"]
["Mon Nov 7 17:24:11 2022", "5 7 0 5 8", "Exiting the program "]
["Mon Nov 7 17:24:20 2022", "3 9 4 9 6", "Logged in!"]
["Mon Nov 7 17:25:24 2022", "8 0 2 2 0", "Logged in!"]
["Mon Nov 7 17:25:33 2022", "2 3 4 9 7", "searching for a book by its ISBN"]
["Mon Nov 7 17:35:52 2022", "5 6 6 6 7", "Searching on the basis of author "]
["Mon Nov 7 17:56:57 2022", "2 5 4 6 3", "searching for a book by title"]
["Mon Nov 7 17:57:23 2022", "8 6 9 2 2", "displaying the books"]
["Mon Nov 7 17:57:33 2022", "0 2 4 7 6", "searching for a book by title"]
["Mon Nov 7 17:58:52 2022", "8 3 3 5 4", "searching for a book by title"]
["Mon Nov 7 18:46:47 2022", "0 2 4 3 7", "Exiting the program "]
["Mon Nov 7 18:54:56 2022", "6 9 8 5 3", "Login Failed"]
["Mon Nov 7 18:56:58 2022", "9 6 0 9 9", "Logged in!"]
["Mon Nov 7 18:57:15 2022", "8 3 1 4 4", "searching for a book by its ISBN"]
["Mon Nov 7 18:57:25 2022", "4 7 3 3 9", "Searching on the basis of author "]
["Mon Nov 7 18:57:41 2022", "6 5 9 2 6", "Searching on the basis of author "]
```

The above is a log of a real file that can be seen to tell that the user logged in, actions done by the user and further telling the time and a unique id for a particular log is for *finding a particular log in the log file*. Further a unique random number is also made into the log script to make searching of a particular log easy. In the source code of this file the writing is done through `json` module.

Software CD

System Requirements –

System Require	Remark
OS	Any OS
Hardware Requirements	At least 2 GB RAM, 10 GB storage
Required Software	Python 3+ configured to path version, MySQL 5+ and Pypi's PyYAML installed using the pip by executing the following command <code>pip install PyYAML</code> or by visiting the Pypi.org website and doing a manual installation