"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.

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

Part I

System and Feasibility

System and Factors of Feasibility

System Analysis

great Library Management Software is a software that helps a library to manage 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.

The LMS comprises of

- Cataloging
- Retrieval
- Adding

as one of the core functionality.

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.

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

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

Usability Analysis

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.

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  import lms.sql_util
3  import lms.menu
4  # import getpass
5
6  i = 0
7
8  # ———————Login———————
9
10 while i < 3:
11     ask_name = input("Enter your name ").title().strip()
12     ask_pass = input("Enter your password ")
13     check_data = (ask_name, ask_pass)
14     # ——passwords retrieval
15     if lms.sql_util.pass_checker(check_data) is False:
16         print(" Invalid user, wrong password or name\nplease try
       again or register as a new user")
17         i += 1
18         print(f"you have {3 - i if 3 - i != 0 else exit()} tries")
19         lms.sql_util.logit(message='Login Failed')
20     else:
21         break
22
23
24 lms.sql_util.logit(message='Logged in!')
25
26 # Body of the program
27 # noinspection PyUnboundLocalVariable
28 lms.menu.menu(user=ask_name)
29 while True:
30     ask_option = input(" ==> ").strip().casefold()
31
32     if ask_option in ['browse', '1']:
33         # display all the isbn details and the books by them
34         lms.sql_util.display(table_name='books')
35         lms.sql_util.logit('displaying the books')
36
37     elif ask_option in ['search', 'find', '2']:
38         search_options = input("""
39         SEARCH mode
40         search by — ISBN(isbn), author(author) or name(name)
41         -> """).strip().casefold()
```

```
42
43          if search_options in ['isbn', '1']:
44              # searching the book using the books ISBN
45              ask_isbn = input("Enter the ISBN number of the book ")
46              # filtering the input
47              if ask_isbn.isnumeric():
48                  lms.sql_util.search_on_isbn(ask_isbn)
49              else:
50                  print("please enter a valid ISBN number")
51              lms.sql_util.logit('searching for a book by its ISBN')
52
53          elif search_options in ['author', '2']:
54              # searching using the author name
55              ask_author = input("Enter the author to search ").title()
    .strip()
56
57              lms.sql_util.search_on_author(ask_author)
58              lms.sql_util.logit("Searching on the basis of author ")
59
60          elif search_options in ['name', 'book name', '3']:
61              # searching using the books name
62              ask_title = input("Enter the Title of the book ").strip()
63
64              lms.sql_util.search_on_title(ask_title)
65              lms.sql_util.logit("searching for a book by title")
66
67      elif ask_option in ['add', 'contribute', 'add books']:
68          # adding the books by the user as a contribution
69          print("To Add books you have to verify that it's you!")
70          verify_user = input("Please enter your name ").strip().title
    ()
71          verify_pass = input("verify your password ")
72          # using the add_books function of the  sql_util package to
73          lms.sql_util.add_books((verify_user, verify_pass))
74          lms.sql_util.logit('Adding to the database')
75
76      elif ask_option in ['menu', 'options']:
77          # menu
78          lms.menu.menu()
79
80      elif ask_option in ['help', 'save me']:
81          # help regarding options
82          lms.menu.helpme()
83
84      elif ask_option in ['explore', '4']:
85          # explore for the library books
86
87          lms.sql_util.explore()
88
```

```
89      elif ask_option in ['exit', 'quit', '5', 'close']:
90          print("Exiting the program")
91          lms.sql_util.logit("Exiting the program ")
92          exit()
93
94      elif ask_option in ['version']:
95          lms.menu.version()
96
97      else:
98          print("I don't recognise that need help type help or menu")
99
100 #  using the logit function from lms.log  print(ls.logit("resenting
    by the user"))
```

main.py

SQL utility

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

```
1  """
2  mysql user credentials
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
14 USER_TABLE = 'lms_users'
15 BOOKS_TABLE = 'books'
16 DEBUG_TABLE = 'test_books'
17 ISSUE_TABLE = 'issue_list'
18
19
20 def main_cnx(user_id='user'):
21     """
22     function that returns the login connection using the
23     cnx_data.yml file
24     """
25     # changing to the data directory
26     if os.path.exists('cnx_data.yml') is False:
27         # os.chdir('..')
```

```
        os.chdir('data')
    with open('cnx_data.yml') as data_file:
        data = yaml.load(data_file, yaml.SafeLoader)

    cnx = mysql.connector.connect(**data[user_id])
    return cnx


def pass_checker(user_data):
    """
    checking the user input to the registered users
    in the database
    :return: boolean value
    """
    # starting the defined connection using the main_cnx() function
    cnx = main_cnx()

    cursor = cnx.cursor()
    # executing the command using execute statement

    cursor.execute(f'select * from {USER_TABLE}')
    # getting the data in the desired form
    database_data = cursor.fetchall()

    # checking the database from the file data
    if user_data in database_data:
        return True
    else:
        return False


def display(table_name='books'):
    """
    show the books, isbn author from the database
    :param table_name:
    :return:
    """
    # initiating the connection
    cnx = main_cnx()
    cursor = cnx.cursor()

    # executing the sql statement for the data
    cursor.execute(f"select * from {table_name}")

    # printing the data form stored in the cursor
    for lines in cursor:
        print(f'{lines[0]:14} {lines[1]:45}by {lines[2]}')


```

```
77 def search_on_isbn(isbn_number: str):
78     """
79     searching using the isbn of the book
80     :return:
81     """
82     cnx = main_cnx()
83     cursor = cnx.cursor()
84     if isbn_number.isnumeric():
85         cursor.execute(f"select * from {BOOKS_TABLE} where isbn = {
       isbn_number!r}")
86         # fetching the data from the database
87         data = cursor.fetchall()
88         # checking for empty data
89         if not data:
90             print(f"Sorry no book is found having ISBN {isbn_number}"
       )
91         else:
92             print('Found')
93             print(data)
94     else:
95         print("Please enter a number to search")
96
97
98 def search_on_author(author_name: str):
99     """
100    searching function using the author name
101    :return:
102    """
103
104    cnx = main_cnx()
105    cursor = cnx.cursor()
106    cursor.execute(f"SELECT book_name, published from {BOOKS_TABLE}
       where author = {author_name!r}")
107    data = cursor.fetchall()
108    # printing the data retrieved from database
109    # listing of the all the books from the author
110    if data:
111        print(f"Books by {author_name}")
112        print(f"Title {'-'*35}Publishing date")
113        for books in data:
114            print(f"{books[0]:40} {books[1]:5}")
115    else:
116        print(f"Author {author_name!r} not found\nPlease check for
       any typos in the author name and try again")
117
118
119 def search_on_title(book_name: str):
120     """
121     searching the books in the database using the
```

```
122         :param book_name:
123         :return:
124         """
125
126         cnx = main_cnx()
127         cursor = cnx.cursor()
128         cursor.execute(f"SELECT book_name, published, author from {
    BOOKS_TABLE} where book_name like {book_name+'%'!r}")
129         data = cursor.fetchall()
130         if data:
131             print("Found")
132             for books in data:
133                 print(f"{books[0]:40} {books[1]}, by {books[2]}")
134
135             return True
136         else:
137             print(f"Not Found with title {book_name!r}")
138             return False
139
140
141 def add_books(verify_user):
142     """
143     Adding the books by the user as a contribution to the project
    database
144     helping it to grow to a more vast book library
145     :param verify_user:
146     :return:
147     """
148     if pass_checker(verify_user) is False:
149         print("Sorry the credentials are wrong")
150     else:
151         cnx = main_cnx()
152         # making the cursor
153         cursor = cnx.cursor()
154         # asking the details of the books by the valid user
155         while True:
156             try:
157                 print("Enter the following details of the book exit
    to leave \n")
158                 ask_isbn = input("Enter the isbn number ").strip().
    casefold()
159                 if ask_isbn in ['exit', 'quit']:
160                     break
161                 ask_book_name = input("Enter the book name ").strip()
162                 ask_author = input(f"Enter the Author of the book {
    ask_book_name!r} ").title().strip()
163                 ask_year = input("Enter the year of publishing ")
164                 # if no exception occurs break the loop
165                 # ------tmp-----##
```

```
166                 cursor.execute(f"insert into {DEBUG_TABLE} values ({
    ask_isbn!r}, {ask_book_name!r}, {ask_author!r},"
167                                f" {ask_year})")
168                 # executing the changes to the table
169                 cnx.commit()
170                 print("*Successfully* added the book to the library
    thanks for the contribution \n"
171                       "help this project to grow.\n")
172 
173             except (mysql.connector.errors.DatabaseError, mysql.
    connector.errors.InterfaceError):
174                 print(f" {'*'*9}SORRY! there was an error, sorry for
    the inconvenience {'*'*9}")
175                 print(f"{'*'*9}Please enter a number value for the
    publishing year{'*'*9}")
176 
177 
178 def explore():
179     """
180     exploring the data
181     :return:
182     """
183 
184     cnx = main_cnx()
185 
186     cursor = cnx.cursor()
187 
188     # getting data for the author
189     cursor.execute(f"select author from {BOOKS_TABLE}")
190     author = cursor.fetchall()
191 
192     # getting the number of books in the database
193     cursor.execute(f'select count(*) from {BOOKS_TABLE}')
194     times = cursor.fetchall()
195 
196     # getting the old books in database
197     cursor.execute(f'select book_name, author from {BOOKS_TABLE}
    where published < 2000 ')
198     old = cursor.fetchall()
199 
200     # processing the retried values
201     classic_time = random.randint(0, len(old) - 1)
202     random_author = author[random.randint(0, len(author) - 1)][0]
203     classic_book = old[classic_time][0]
204     classic_author = old[classic_time][1]
205     total_books = times[0][0]
206 
207     print(fr"""
208     +{'-' * 30}LIBRARY MANAGEMENT SYSTEM{'-' * 30}+
```

```
    |{" "*85}|
    |    Read 'By Authors like{" "*61}|
    |    {random_author}{" "*(91 - (8 + 1 + len(random_author)))}|
    |    ``````` Total books in library {total_books} ```````{"
    "*(91- (49+len(str(total_books))))}|
    |    ~Time less classics{" "*63}|
    |    {classic_book}     by' {classic_author}{" "*(91 - (17+1+len(
    classic_author)+len(classic_book)))}|
    |{" "*85}|
    +{'-' * 30}{'*' * 25}{'-' * 30}+
        """)


def logit(message=''):
    """
    logging the events happened in the LMS
    :param message:
    :return:
    """

    if os.path.exists('logfile.log') is False:
        with open('logfile.log', 'x') as new_file:
            pass

    number_id = ' '.join(secrets.choice(string.digits) for _ in range
    (5))
    log_data = [time.asctime(time.localtime()), number_id, message]

    with open('logfile.log', 'a') as log_file:
        json.dump(log_data, log_file)
        log_file.write('\n')

    return number_id
```

sql_util.py

Menu and Help

Menu file stores the menus and helps

```
"""
menu, options and help for the file:main.py
"""


def menu(user=''):
    print(f"""
   +{'-'*60}+
```

```
    |             Library Management System                        |
    | Hi {user}{" "*(65-(1+8+len(user)))}|
    |     1.Browse books (browse)                                  |
    |     2.Search for the book (find)                             |
    |     3.Add Books (add)                                        |
    |     4.Explore (explore)                                      |
    |     5.exit (exit)                                            |
    +{'-'*60}+
    | For help enter help, for version information enter version |
    +{'-'*60}+
     """)


def helpme():
     print("""
     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
     """)


def version():
     print("""
```

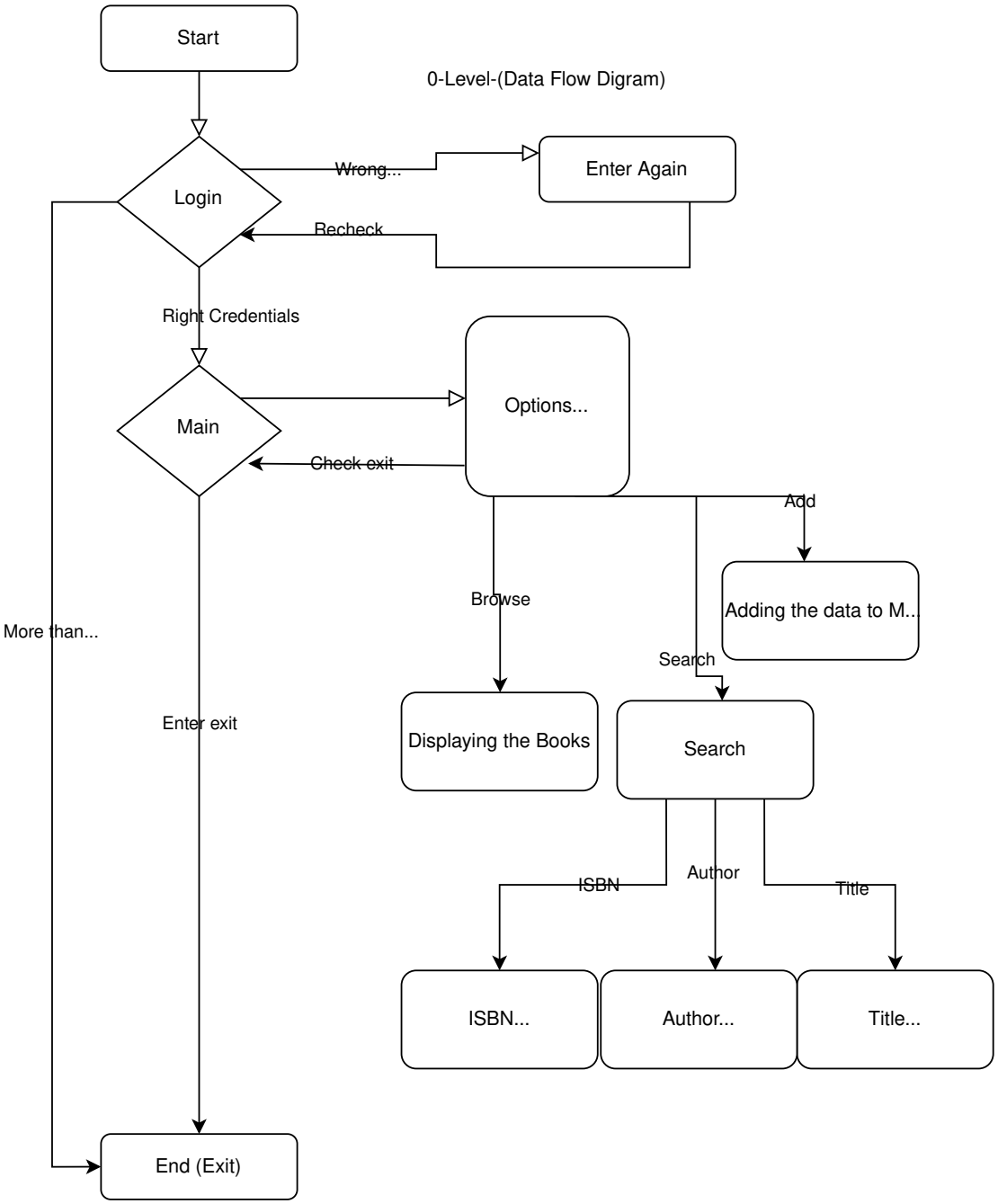
```
54      version information '0.5' 'Bloodymary'
55      """)
```

menu.py

Data Flow of the Program

Data flow diagram for the program is –

System Design



SQL Database Structure

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.

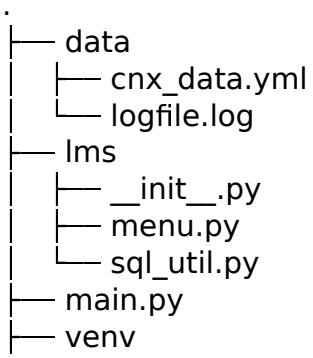


Figure 1: Directory Structure

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 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.

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 in 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 buy 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 . 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 tile 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 show casing 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

- ☐

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 recognise 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 program[1] □

```
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 |
```

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

```
   +-----------------------------------------------------------+

 ==> 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
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
```

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.*