"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]

[1]The 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 **5**
System and Factors of Feasibility 6
System Analysis 6
ISBN 6
Cataloging 6
Feasibility Study 7
Usability Analysis and features 7

II Source and Program Structure **9**
Source Code 10
main.py 10
SQL utility 13
Menu and Help 20
About Package `lms` 22
SQL Commands Imported 22
Users database 22
Books database 25
Data Flow of the Program 27
System Design 28
SQL Database Structure 29
Program Dependency tree 31

III Post Updates **32**
Future Updates 33

IV Case Study **34**
Login 35
Running 37
Working 41
Case 1: Adding the books to the books database 41
Case 2: Browsing and Exploring the library database 42
Case 3: Using the Different Searches 43
Case 4: using the help 45
Exception Handling 47
Overall Execution 49
Program Module information 51
Usage of Modules 52
Logging the Actions 54

Software CD **55**

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.

Some important terminology regarding the LMS

ISBN

ISBN stands for the International Standard Book Number is a commercial book identifier intended to be unique[2]. 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.

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 locations[3]. The conventional way of cataloging the books are using the card system. Card Catalog is the a method

[2]https://en.wikipedia.org/wiki/ISBN
[3]https://en.wikipedia.org/wiki/Library_catalog

Figure 1: ISBN number in a barcode

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.*

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.

The user

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

```
"""
main program file
"""

import lms.sql_util
import lms.menu
# import getpass

i = 0

# ———————Login———————
ask_name = None

while i < 3:
    ask_name = input("Enter your name ").title().strip()
    ask_pass = input("Enter your password ")
    check_data = (ask_name, ask_pass)
    # ——passwords retrieval
    if lms.sql_util.pass_checker(check_data) is False:
        print(" Invalid user, wrong password or name\nplease try again or register as a new user")
        i += 1
        print(f"you have {3 - i if 3 - i != 0 else exit()} tries")
        lms.sql_util.logit(message='Login Failed')
    else:
        # if the user is found in the database of the users
        break


lms.sql_util.logit(message='Logged in!')

# Body of the program
lms.menu.menu(user=ask_name)
while True:
    ask_option = input(" ⟹ ").strip().casefold()

    if ask_option in ['browse', '1']:
        # display all the isbn details and the books by them
        lms.sql_util.display(table_name='books')
        lms.sql_util.logit('displaying the books')

    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()
    .strip()
63
64              lms.sql_util.search_on_author(ask_author)
65              lms.sql_util.logit("Searching on the basis of author ")
66
67          elif search_options in ['name', 'book name', 'title', '3']:
68              # searching using the books name
69              ask_title = input("Enter the Title of the book ").strip()
70
71              lms.sql_util.search_on_title(ask_title)
72              lms.sql_util.logit("searching for a book by title")
73
74      elif ask_option in ['add', 'contribute', 'add books']:
75          # adding the books by the user as a contribution
76          print("To Add books you have to verify that it's you!")
77          verify_user = input("Please enter your name ").strip().title
    ()
78          verify_pass = input("verify your password ")
79          # using the add_books function of the  sql_util package to
80          lms.sql_util.add_books((verify_user, verify_pass))
81          lms.sql_util.logit('Adding to the database')
82
83      elif ask_option in ['menu', 'options']:
84          # main menu
85          lms.menu.menu()
86
87      elif ask_option in ['help', 'save me']:
88          # 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 loging 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

```
"""
mysql interaction and other main functions for the main.py file.
"""
import os
import yaml
import mysql.connector
import random
import secrets
import json
import time
import string


USER_TABLE = 'lms_users'
BOOKS_TABLE = 'books'
DEBUG_TABLE = 'test_books'
ISSUE_TABLE = 'issue_list'


def main_cnx(user_id='user'):
    """
    function that returns the login connection using the
    cnx_data.yml file
    """
    # changing to the data directory
    try:
        if os.path.exists('cnx_data.yml') is False:
            # 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
    except FileNotFoundError:
        # if the data directory is not found in the current directory
        # print that the
        print("FATAL ERROR :The directory 'data' does not exists
    please recover the data directory")
        exit()


def pass_checker(user_data):
    """
```

```
44      checking the user input to the registered users
45      in the database
46      :return: boolean value
47      """
48      # starting the defined connection using the main_cnx() function
49      cnx = main_cnx()
50
51      cursor = cnx.cursor()
52      # executing the command using execute statement
53
54      cursor.execute(f'select * from {USER_TABLE}')
55      # getting the data in the desired form
56      database_data = cursor.fetchall()
57
58      # checking the database from the file data
59      if user_data in database_data:
60          return True
61      else:
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          else:
98              print('Found')
99              print(f"""
100             ISBN: {data[0][0]}
101             Title: {data[0][1]}
102             Author: {data[0][2]}
103             Published: {data[0][3]}""")
104     else:
105         print("Please enter a number to search")
106
107
108 def search_on_author(author_name: str):
109     """
110     searching function using the author name
111     :return:
112     """
113
114     cnx = main_cnx()
115     cursor = cnx.cursor()
116     cursor.execute(f"SELECT book_name, published from {BOOKS_TABLE}
    where author = {author_name!r}")
117     data = cursor.fetchall()
118     # printing the data retrieved from database
119     # listing of the all the books from the author
120     if data:
121         print(f"Books by {author_name}")
122         print(f"Title {'-'*35}Publishing date")
123         for books in data:
124             print(f"{books[0]:40} {books[1]:5}")
125     else:
126         print(f"Author {author_name!r} not found\nPlease check for
    any typos in the author name and try again")
127
128
129 def search_on_title(book_name: str):
130     """
131     searching the books in the database using the sql query like
    functionality
132     :param book_name:
133     :return:
134     """
135
136     cnx = main_cnx()
```

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

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

```
223     classic_author = old[classic_time][1]
224     total_books = times[0][0]
225 
226     # printing the result in the Command line using the formatted
        string
227     print(f"""
228     +{'-' * 30}LIBRARY MANAGEMENT SYSTEM{'-' * 30}+
229     |{" "*85}|
230     |    Read `By Authors like{" "*61}|
231     |    {random_author}{" "*(91 - (8 + 1 + len(random_author)))}|
232     |    ``````` Total books in library {total_books} ```````{"
        "*(91- (49+len(str(total_books))))}|
233     |    ~Time less classics{" "*63}|
234     |    {classic_book}     by' {classic_author}{" "*(91 - (17+1+len(
        classic_author)+len(classic_book)))}|
235     |{" "*85}|
236     +{'-' * 30}{'*' * 25}{'-' * 30}+
237         """)
238 
239 
240 def logit(message=''):
241     """
242     logging the events happened in the LMS in the separate file
243     called logfile
244     :param message: str
245     :return: number_id -> str
246     """
247 
248     # if the file logfile.log does not exist create the new file
        named logfile.log
249     if os.path.exists('logfile.log') is False:
250         with open('logfile.log', 'x') as _:
251             pass
252 
253     # generating the random number
254     number_id = ' '.join(secrets.choice(string.digits) for _ in range
        (5))
255     # making the log data
256     log_data = [time.asctime(time.localtime()), number_id, message]
257 
258     # using the json to dump the list into a file and adding the new
        line after each dump
259     with open('logfile.log', 'a') as log_file:
260         # dumping the list of the log data to the log file
261         json.dump(log_data, log_file)
262         # adding the new line at the end of the file
263         log_file.write('\n')
264 
265     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

```

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

menu.py

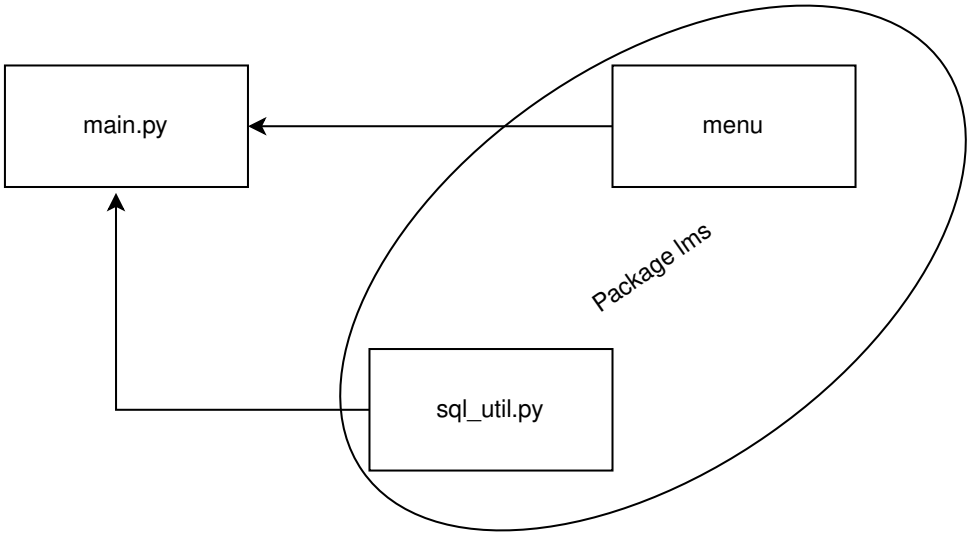


Figure 2: Dependency of main.py to the lms package

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

```
-- MySQL dump 10.13  Distrib 8.0.31, for Linux (x86_64)
--
-- Host: localhost    Database: people
-- ------------------------------------------------------
-- Server version    8.0.31-0ubuntu2

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!50503 SET NAMES utf8mb4 */;
/*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;
/*!40103 SET TIME_ZONE='+00:00' */;
/*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
    FOREIGN_KEY_CHECKS=0 */;
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='
    NO_AUTO_VALUE_ON_ZERO' */;
/*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;

--
-- Table structure for table `lms_users`
--

DROP TABLE IF EXISTS `lms_users`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `lms_users` (
  `name` varchar(50) DEFAULT NULL,
  `password` varchar(23) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `lms_users`
--

LOCK TABLES `lms_users` WRITE;
/*!40000 ALTER TABLE `lms_users` DISABLE KEYS */;
INSERT INTO `lms_users` VALUES ('Kate Stewart','kate123'),('Brian
    Smith','123'),('Sam Raimi','supersam'),('Monte Cue','python.org');
/*!40000 ALTER TABLE `lms_users` ENABLE KEYS */;
UNLOCK TABLES;
/*!40103 SET TIME_ZONE=@OLD_TIME_ZONE */;

/*!40101 SET SQL_MODE=@OLD_SQL_MODE */;
/*!40014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS */;
/*!40014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS */;
/*!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

```
-- MySQL dump 10.13  Distrib 8.0.31, for Linux (x86_64)
--
-- Host: localhost    Database: people
-- ------------------------------------------------------
-- Server version    8.0.31-0ubuntu2

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!50503 SET NAMES utf8mb4 */;
/*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;
/*!40103 SET TIME_ZONE='+00:00' */;
/*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
    FOREIGN_KEY_CHECKS=0 */;
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='
    NO_AUTO_VALUE_ON_ZERO' */;
/*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;

--
-- Table structure for table `books`
--

DROP TABLE IF EXISTS `books`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `books` (
  `isbn` varchar(20) NOT NULL,
  `book_name` varchar(200) DEFAULT NULL,
  `author` varchar(40) DEFAULT NULL,
  `published` int DEFAULT NULL,
  PRIMARY KEY (`isbn`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `books`
--

LOCK TABLES `books` WRITE;
/*!40000 ALTER TABLE `books` DISABLE KEYS */;
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);
/*!40000 ALTER TABLE `books` ENABLE KEYS */;
UNLOCK TABLES;
/*!40103 SET TIME_ZONE=@OLD_TIME_ZONE */;

/*!40101 SET SQL_MODE=@OLD_SQL_MODE */;
/*!40014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS */;
/*!40014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS */;
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
/*!40111 SET SQL_NOTES=@OLD_SQL_NOTES */;

-- Dump completed on 2022-11-12  0:08:09
```

books_database.sql

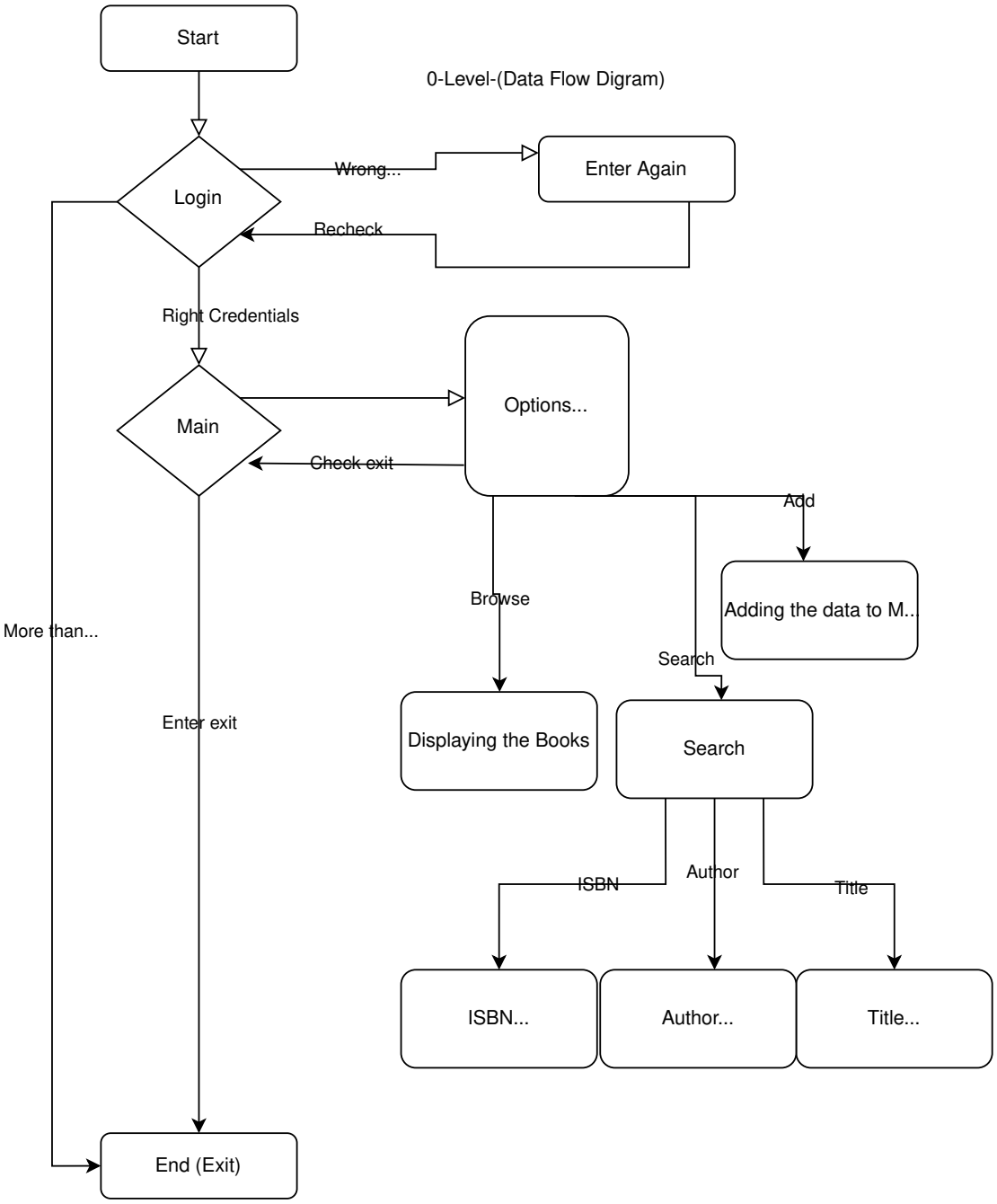
Data Flow of the Program

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.

System Design



SQL Database Structure

The SQL tables are arranges 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 the 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`.

```
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)
```

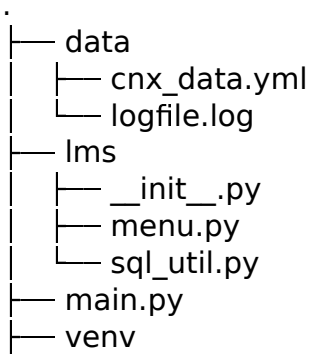


Figure 3: Directory Structure

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.

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 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 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

- ☐

```
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 explore[4] 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.

[4]The 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 program[5] □

```
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
```

[5]The 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
```

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 line[6]. 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-connetor
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.

[6]Python Software Foundation recommends using pip for installing the packages
[7]for 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 JSON[8] 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 bulitin 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()`

[8]Java 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 JSON[9] it is extremely human readable.

[9]JSON 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 `pip install PyYAML` or by visiting the Pypi.org website and doing a manual installation