**LIBRARY MANAGEMENT SYSTEM**

### …..INDEX……

| **S. No** | **Contents** | **Page No** |
| --- | --- | --- |
|  | **Introduction** | **01** |
|  | **Project Details** | **02** |
|  | **Objective of the Project** | **06** |
|  | **Technology Used: Hardware & Software Requirements** | **08** |
|  | **Requirement Specification** | **11** |
|  | **Design Specification** | **12** |
|  | **Possible Outcomes:**  **a. Front Pages**  **b. Data Flow Diagram (DFD)**  **c. Software Process Model**  **d. Functional Diagram** | **19**  **46**  **48**  **50** |
|  | **Conclusion** | **52** |
|  | **Bibliography** | **53** |

**INTRODUCTION**

**1.1 What is Library Management System:**

Libbrary management system is a system where keep all tracks of a library operation such as items, bill, paid, and also patrons record. By this software we can operate all the library operation easily. Integrated library system there is two interfaces one is patron and another one is for library staff. In library system the operation of a user/member/patron and staff of library is different. A user can search a book, view book list which are available in library, can issue book, renew book, can hold book, can print issue list, can edit his/her information (patron information). Online public excess catalogue, circulation, some function under serial control can operate by patron. But a staff can store bibliographic (book, CD, DVD, journal etc) record actually library materials record in database, can create patron in database, order a book, purchase a book. Acquisition, cataloging, some function under serial control, management all are operate by staff of a library.

A college library management is a project that manages and stores books information electronically according to students needs. The system helps both students and library manager to keep a constant track of all the books available in the library. It allows both the admin and the student to search for the desired book. It becomes necessary for colleges to keep a continuous check on the books issued and returned and even calculate fine. This task if carried out manually will be tedious and includes chances of mistakes. These errors are avoided by allowing the system to keep track of information such as issue date, last date to return the book and even fine information and thus there is no need to keep manual track of this information which thereby avoids chances of mistakes.

**Modules:**

* **Admin login:** Admin is the one who administers the system by adding or removing e-books into and from the system respectively.
* **User login:** Students have to register themselves into the system to create an account. After registering successfully, they can then login into the system by entering 10 digit mobile number and their email id.
* **Add and Update Books:** The admin can add books to the system by entering the details of the books and can even update the details.
* **View Order-**The admin can view order for the books.
* **Calculate Fine-** The student can view the issue and expiry date for the book issued and can even calculate fine.

**1.2 Requirement Analysis of Library Management System:**

In complete modern library management system there are different types of

modules or features, which are:

1. Acquisition

2. Cataloguing

3. Online Public Excess Catalogue (OPAC)

4. Circulation

5. Serial Control

6. Management or Report

7. System maintenance Facilities or System Parameters

**Project Overview**

This year the summer training project titled 'Library Management System' is a vast one with the usage of numerous resources and popular software like the Microsoft Visual Studio 2008 and SQL server management to build a website for the easiness of the library staff.

The project’s main aim is to build a management system more like a website which

* Can create a computerized management system for a library
* Has the capability to issue books
* Has the capability to return books
* Has an administrator account
* Can manage users through administrator account
* Has customized profile with photo of the user
* Tracks the books that users have issued
* Keeps the databases correct and up-to-date
* Can store all the book and user data in a proper manner

With the blooming of technology I thought it would be of a good sense to automate

some of our oldest systems. So I asked myself which system is

* Not up-to-date
* Requires management
* Has its components value very high in the education department

That’s when I came up with a plan to create an application ‘Library Management System’

By doing our practical ,where I learned

* Visual Python
* Using Microsoft Visual Studio
* Using SQL server management
* Various dynamics of creating pages and connecting them in order to get an application

Before going into any details I would like to explain exactly what this desktop application does

and why is it important

The application consists of several pages having different

* Views
* Functionality
* Purpose

The following pages constitute the ‘ Library Management System’ application

Main Page:

* Consisting of the WELCOME and REGISTRAION columns.

Login Page:

* Includes an interface to Login both for users and administrator.

Admin Page:

* Includes links to other pages: ‘Add Books’, ‘Remove Books’,
* ‘Manage Users’.
* Has a welcome page with profile options
* A Log out tab

User Page:

* Includes links to other pages: ‘Issue Books’, ‘Account’, ‘Return
* Books’
* And a welcome page with profile options
* A Log out tab

All these pages are interlinked using Microsoft visual studio 2008 and the coding which it uses is that of Python. The data is held in databases managed by the SQL server management 2005 which gives the facility to connect the database to the website so that the data can be viewed, updated or removed as required.

**Product Functions**

The Library Management System provides College/Organization real time information about the books available in the Library and the user information. The functions of the system include the system providing different type of services based on the type of users

[Member/Admin].

* The member should be provided with the updated information

about the books catalog.

* Provisions for the members to borrow the books they want, if all

the other required rules hold good.

* The member is given a provision to check his account information

and change the account information any time in the given valid

period.

* The members are provided with the books available roster and

allowed to choose the books, which they want to use in the coming

up days.

* The admin can get the information about the members who have

borrowed or returned the books.

The admin is provided with interfaces to add/delete the books available in the book catalog.

**User characteristics**

The users of the system are members and the administrator who maintains the system. The member is assumed to have basic knowledge of the computers and Internet browsing. The administrator of the system has more knowledge of the internals of the system and is able to rectify the small problems that may arise due to disk crashes, power failures and other catastrophes to maintain the system. The proper user interface, user’s manual, Colleges/organization help and the guide to install and maintain the system must be sufficient to educate the users on how to use the system without any problems.

**Constraints**

* The information of all the users must be stored in a database that is

accessible by the Library Management System. .

* The users must have their correct usernames and passwords to enter

into the Library Management System.

**Assumptions and dependencies**

* The users have sufficient knowledge of computers.
* The users know the English language, as the user interface will be

provided in English

* The product can access the student database

**FEASIBILITY STUDY**

In feasibility study phase various steps have been taken:

1. Identify information at different levels.

2. Identify the expectation of user from an automated

system.

3. Analyze the importance of automated system

**Feasibility study:**

In order to make sure that the project is feasible, following feasibility studies have been conducted: -

Technical Feasibility study:

It is possible to develop the system using simple platform. All the functions of a project for communication can be implemented in the new system. Hence the system is technically feasible.

Economic Feasibility study:

Library Management System is a worth making project. This project is economically feasible in every sense that it takes less effort, less time, and nominal cost of purchasing the tools for developing the software.

Legal feasibility study:

Since the project needs no copyright, patenting, and doesn’t intent to have any relation with anybody else’s intellectual property rights, it can be considered as a legally feasible project.

Time feasibility study:

As it has been more probable (as per the requirements, functions, and performance specifications of the system) that the project can be completed within the given time frame, it is considered that the undertaking this project is feasible in the context of time.

Operational feasibility study:

This system is completely operational and can be successfully implemented. Library Management System is easy to understand not only for any sophisticated users but for the naïve users as well .It provides simple ambience in which user can feel free to work faster, easier, and more accurately. Therefore, it can be socially and behaviorally accepted and is feasible too.

Project planning:

During Planning all the activities that are to be performed to create the system are planned. Following are some of the issues that are well devised so that proper monitoring and controlling of the project could be easily done: -

Project development:

To avoid being stuck in dilemma during the development of project, one need to be sure that the process model he’s using is right for the project. Since all the requirements about the problem can’t be easily understood and may not be stable during the development of the system.

Quality Planning:

To ensure that the final software for-“Library Management System” is of high quality, some quality control activities should be decided /planned in advance to perform them throughout the development of the software.

Following is a list of quality control activities that are used to identify and remove defects from the software, hence making it a high quality controlled system: -

Requirements Review

Design Reviews

Code Reviews

Testing

Risk control planning:

A risk is a probabilistic event – it may or may not occur. The aim of risk control planning is to minimize the impact of risks (if they occur) in the project. Following are some known risks that might occur and their mitigation plan: -

Unclear project requirements Keep in touch with the faculty in charge.

Data loss Use CD’s and/or pen drives to have some extra backups of the data.

Project delays Use proper scheduling of the project as soon as possible so that the

project could be completed.

Software size planning:

It has been taken into account that there are some functions in project that are indispensable .And these should not be excluded from project .Such functions/modules are like login, Sign up, admin add, admin remove, add user, remove user. So there are at least a minimum number of modules that have to be there in project.

Effort / cost Estimation:

Project scheduling:

During early stages of project planning, a kind of macroscopic schedule was already planned which gave a rough idea about activities that should be carried out for developing the project .In project scheduling , those sets of activities are refined into a detailed schedule.

Product Perspective:

Library Management System is a product which does not intent to have any relation with any other product. It is a complete system in itself. It is an exclusive product which is to be concerned with the optimization.

Product Features: The project mainly use the concepts of .Net, simple tools of programming and SQL server for database storage User Classes and Characteristics.

There are number of functions that the system/product is supposed to perform which

is as follows: -

* Issue books
* Remove books
* Admin login
* User login
* Add books
* Welcome page
* New user

The user of this product need not be computer expert. Even a naïve user can also operate the system. The user interfaces are to be made so simple that anybody can be comfortable in working with the system in just a few minutes. The basic things which are required in a user are:

* User should know what the computer is.
* User should understand English.
* User must know which key (button or keyboard) does what.
* User must have an experience of library management.

**3. OBJECTIVE**

With the growing information technology industry, automation of their system and management is desired by all kind of commercial enterprises. As the name suggest Library Management System will deal in the entire requirement needed for managing the activities of College/Organization Library. It will deal with the process of maintaining data about the books and many other things as well as transactions which are taking place in the library with respect to the Issue, Cataloguing, Searching and Return of the books

.

Library Management System maintains the record of books in the library, issue, purchasing and return process of the books in the library. Here we are primarily concerned with management of books of library. In this project we identify the need for computer based Library Management Systems.

Hence this covers the following issues:

* + Maintains data about the books of the library.
  + Arranging data in logical order for easy maintenance.
  + Collection of data about books which are issued and requirement for purchasing.
  + To provide various search options to know the availability of books in the Library.
  + Data about books which are lost.
  + Generation of various reports according to the management request i.e. Cataloguing, Searching etc.

Library is a growing organism that requires constant positive changes to meet the need of its user. The invention of computer has brought in a rapid change in the society. Therefore, automation has become the need of the hour. Library automation not only improves the image of the library staff but also provides additional services to the users with the existing staff. The impact of automation on the library is quite obvious; it creates new environment where each function redefines the traditional organizational structure and transforms it into new institutional entries. In this unit a brief overview is given about library automation.

Automation is defined as a technique, a process, or a system which operates automatically. According to the Encyclopedia of Library and Information Science, “Automation is the technology concerned with a design and development of the process and systems that minimize the necessity of human intervention in their operation.

Swihart Stanley S and Hefley Beryl F have defined the term ‘library automation’ as “the processing of certain routine clerical function in the library with the assistance of computer or other mechanized or semi automatic equipment”. It may also be defined as a process of mechanization of all the housekeeping operation of a library which is repetitive in nature. The housekeeping operation includes acquisition, cataloguing, circulation, serial control, references and administration work.

Automation is a technique to make a system automated, i.e. self active. For this the electronic machines are used to automate the libraries. By automation, libraries activities such as acquisition, circulation, serial control, information retrieval, cataloguing and indexing can be mechanized by using library software’s.

The Library Automation means:-

* + Computerization of the entire house keeping operation of the library.
  + Operate a computerization library management system.
  + Offer new services based on the technologies and also integrate the traditional library operations in the era.

**Scope**

This application can be used by any Library to automate the process of manually maintaining the records related to the subject of purchasing of books, selecting and ordering items for the collection and maintaining the accounts, Creating records for material held in the collection and providing access to the catalogue-Via an Online Public Access Catalogue (OPAC).

**4 Technology Used Hardware & Software Requirement**

**Technology Used:**

To build this project we have used python programming language . Which is a High Level Programming language. It is a dynamic and interpreter type programming language.

## What is Python?

**Python** is a very popular general-purpose interpreted, interactive, object-oriented, and high-level programming language. Python is dynamically-typed and garbage-collected programming language. It was created by Guido van Rossum during 1985- 1990. Like Perl, Python source code is also available under the GNU General Public License (GPL).

Python supports multiple programming paradigms, including Procedural, Object Oriented and Functional programming language. Python design philosophy emphasizes code readability with the use of significant indentation.

This tutorial gives a complete understanding of Python programming language starting from basic conceopts to advanced concepts. This tutorial will take you through simple and practical approaches while learning Python Programming language.

**Python** is a MUST for students and working professionals to become a great Software Engineer specially when they are working in Web Development Domain. I will list down some of the key advantages of learning Python:

* **Python is Interpreted** − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
* **Python is Interactive** − You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
* **Python is Object-Oriented** − Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
* **Python is a Beginner's Language** − Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

## Applications of Python

The latest release of Python is 3.x. As mentioned before, Python is one of the most widely used language over the web. I'm going to list few of them here:

* **Easy-to-learn** − Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
* **Easy-to-read** − Python code is more clearly defined and visible to the eyes.
* **Easy-to-maintain** − Python's source code is fairly easy-to-maintain.
* **A broad standard library** − Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
* **Interactive Mode** − Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
* **Portable** − Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
* **Extendable** − You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
* **Databases** − Python provides interfaces to all major commercial databases.
* **GUI Programming** − Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
* **Scalable** − Python provides a better structure and support for large programs than shell scripting.

**Python Tkinter:**

Tkinter is the most commonly used library for developing GUI (Graphical User Interface) in Python. It is a standard Python interface to the Tk GUI toolkit shipped with Python. As Tk and Tkinter are available on most of the Unix platforms as well as on the Windows system, developing GUI applications with Tkinter becomes the fastest and easiest.



This Tkinter Tutorial will help learning Tkinter from Basics to advance GUI development, including all the necessary functions explained in detail.

Tkinter tutorial provides basic and advanced concepts of Python Tkinter. Our Tkinter tutorial is designed for beginners and professionals.

Python provides the standard library Tkinter for creating the graphical user interface for desktop based applications.

Developing desktop based applications with python Tkinter is not a complex task. An empty Tkinter top-level window can be created by using the following steps.

1. import the Tkinter module.
2. Create the main application window.
3. Add the widgets like labels, buttons, frames, etc. to the window.
4. Call the main event loop so that the actions can take place on the user's computer screen.

**Software Used:-**

There are two application software are use to build this project they are-

1. Microsoft Visual Studio (VS Code)
2. MySQL
   * 1. **Microsoft Visual Studio(VS Code):**

Visual Studio Code, also commonly referred to as VS Code, is a [source-code editor](https://en.wikipedia.org/wiki/Source-code_editor) made by [Microsoft](https://en.wikipedia.org/wiki/Microsoft) with the [Electron Framework](https://en.wikipedia.org/wiki/Electron_(software_framework)), for [Windows](https://en.wikipedia.org/wiki/Windows), [Linux](https://en.wikipedia.org/wiki/Linux) and [macOS](https://en.wikipedia.org/wiki/MacOS). Features include support for [debugging](https://en.wikipedia.org/wiki/Debugging), [syntax highlighting](https://en.wikipedia.org/wiki/Syntax_highlighting), [intelligent code completion](https://en.wikipedia.org/wiki/Intelligent_code_completion), [snippets](https://en.wikipedia.org/wiki/Snippet_(programming)), [code refactoring](https://en.wikipedia.org/wiki/Code_refactoring), and embedded [Git](https://en.wikipedia.org/wiki/Git). Users can change the [theme](https://en.wikipedia.org/wiki/Theme_(computing)), [keyboard shortcuts](https://en.wikipedia.org/wiki/Keyboard_shortcut), preferences, and install [extensions](https://en.wikipedia.org/wiki/Plug-in_(computing)) that add additional functionality.

In the [Stack Overflow](https://en.wikipedia.org/wiki/Stack_Overflow) 2021 Developer Survey, Visual Studio Code was ranked the most popular developer environment tool among 82,000 respondents, with 70% reporting that they use it.

**Features:**

Visual Studio Code is a source-code editor that can be used with a variety of programming languages, including [C](https://en.wikipedia.org/wiki/C_(programming_language)), [C#](https://en.wikipedia.org/wiki/C_Sharp_(programming_language)), [C++](https://en.wikipedia.org/wiki/C%2B%2B), [Fortran](https://en.wikipedia.org/wiki/Fortran), [Go](https://en.wikipedia.org/wiki/Go_(programming_language)), [Java](https://en.wikipedia.org/wiki/Java_(programming_language)), [JavaScript](https://en.wikipedia.org/wiki/JavaScript), [Node.js](https://en.wikipedia.org/wiki/Node.js), [Python](https://en.wikipedia.org/wiki/Python_(programming_language)), [Rust](https://en.wikipedia.org/wiki/Rust_(programming_language)). It is based on the [Electron](https://en.wikipedia.org/wiki/Electron_(software_framework)) framework which is used to develop [Node.js](https://en.wikipedia.org/wiki/Node.js) [web applications](https://en.wikipedia.org/wiki/Web_application) that run on the [Blink layout engine](https://en.wikipedia.org/wiki/Blink_layout_engine). Visual Studio Code employs the same editor component (codenamed "Monaco") used in [Azure DevOps](https://en.wikipedia.org/wiki/Azure_DevOps_Server) (formerly called Visual Studio Online and Visual Studio Team Services).

Out of the box, Visual Studio Code includes basic support for most common programming languages. This basic support includes [syntax highlighting](https://en.wikipedia.org/wiki/Syntax_highlighting), [bracket matching](https://en.wikipedia.org/wiki/Bracket_matching), [code folding](https://en.wikipedia.org/wiki/Code_folding), and configurable snippets. Visual Studio Code also ships with [IntelliSense](https://en.wikipedia.org/wiki/Intelligent_code_completion) for JavaScript, TypeScript, [JSON](https://en.wikipedia.org/wiki/JSON), [CSS](https://en.wikipedia.org/wiki/CSS), and [HTML](https://en.wikipedia.org/wiki/HTML), as well as debugging support for Node.js. Support for additional languages can be provided by freely available extensions on the VS Code Marketplace.

Instead of a project system, it allows users to open one or more directories, which can then be saved in workspaces for future reuse. This allows it to operate as a [language-agnostic](https://en.wikipedia.org/wiki/Language-agnostic) code editor for any language. It supports many programming languages and a set of features that differs per language.

* + 1. **MySQL:-**

**MySQL** is an [open-source](https://en.wikipedia.org/wiki/Open-source_software) [relational database management system](https://en.wikipedia.org/wiki/Relational_database_management_system) (RDBMS). Its name is a combination of "My", the name of co-founder [Michael Widenius](https://en.wikipedia.org/wiki/Michael_Widenius)'s daughter My, and "SQL", the acronym for [Structured Query Language](https://en.wikipedia.org/wiki/Structured_Query_Language). A [relational database](https://en.wikipedia.org/wiki/Relational_database) organizes data into one or more data tables in which data may be related to each other; these relations help structure the data. SQL is a language programmers use to create, modify and extract data from the relational database, as well as control user access to the database. In addition to relational databases and SQL, an RDBMS like MySQL works with an [operating system](https://en.wikipedia.org/wiki/Operating_system) to implement a relational database in a computer's storage system, manages users, allows for network access and facilitates testing database integrity and creation of backups.

MySQL is [free and open-source software](https://en.wikipedia.org/wiki/Free_and_open-source_software) under the terms of the [GNU General Public License](https://en.wikipedia.org/wiki/GNU_General_Public_License), and is also available under a variety of [proprietary](https://en.wikipedia.org/wiki/Proprietary_software) licenses. MySQL was owned and sponsored by the [Swedish](https://en.wikipedia.org/wiki/Sweden) company [MySQL AB](https://en.wikipedia.org/wiki/MySQL_AB), which was bought by [Sun Microsystems](https://en.wikipedia.org/wiki/Sun_Microsystems) (now [Oracle Corporation](https://en.wikipedia.org/wiki/Oracle_Corporation)). In 2010, when [Oracle acquired Sun](https://en.wikipedia.org/wiki/Acquisition_of_Sun_Microsystems_by_Oracle_Corporation), Widenius [forked](https://en.wikipedia.org/wiki/Fork_(software_development)) the [open-source](https://en.wikipedia.org/wiki/Open-source) MySQL project to create [MariaDB](https://en.wikipedia.org/wiki/MariaDB).

**Features:**

MySQL is offered under two different editions: the [open source](https://en.wikipedia.org/wiki/Open-source_software) MySQL Community Server and the proprietary [Enterprise Server](https://en.wikipedia.org/wiki/MySQL_Enterprise). MySQL Enterprise Server is differentiated by a series of proprietary extensions which install as server plugins, but otherwise shares the version numbering system and is built from the same code base.

Major features as available in MySQL 5.6:

* A broad subset of [ANSI SQL 99](https://en.wikipedia.org/wiki/SQL:1999), as well as extensions
* Cross-platform support
* [Stored procedures](https://en.wikipedia.org/wiki/Stored_procedure), using a procedural language that closely adheres to [SQL/PSM](https://en.wikipedia.org/wiki/SQL/PSM)
* [Triggers](https://en.wikipedia.org/wiki/Database_trigger)
* [Cursors](https://en.wikipedia.org/wiki/Cursor_(databases))
* Updatable [views](https://en.wikipedia.org/wiki/View_(SQL))
* Online [Data Definition Language](https://en.wikipedia.org/wiki/Data_Definition_Language) (DDL) when using the InnoDB Storage Engine.
* [Information schema](https://en.wikipedia.org/wiki/Information_schema)
* Performance Schema that collects and aggregates statistics about server execution and query performance for monitoring purposes.
* A set of SQL Mode options to control [runtime](https://en.wikipedia.org/wiki/Run_time_(program_lifecycle_phase)) behavior, including a strict mode to better adhere to SQL standards.

**Hardware Requirements**

Processor : i3 7th generation

RAM : 4GB RAM

Monitor. : 15” Color Monitor

Keyboard

Mouse

Operating system : Windows 7

Developing Tool : MS Visual Studio Code

Database : MySQL

Python is a high-level, general-purpose programming language. Its design philosophy emphasizes code readability with the use of significant indentation.

Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly procedural), object-oriented and functional programming. It is often described as a "batteries included" language due to its comprehensive standard library.

Guido van Rossum began working on Python in the late 1980s as a successor to the ABC programming language and first released it in 1991 as Python 0.9.0.[36] Python 2.0 was released in 2000. Python 3.0, released in 2008, was a major revision not completely backward-compatible with earlier versions. Python 2.7.18, released in 2020, was the last release of Python 2.

Python consistently ranks as one of the most popular programming languages.

Python is meant to be an easily readable language. Its formatting is visually uncluttered and often uses English keywords where other languages use punctuation. Unlike many other languages, it does not use curly brackets to delimit blocks, and semicolons after statements are allowed but rarely used. It has fewer syntactic exceptions and special cases than C or Pascal.

**Output**

After the processing and validation of data in LMS to provide the following outputs:-

* + Report of available books.
  + To view the account details of purchase book.
  + To view the requirement of library.
  + To view report of publisher.
  + To view the list of purchase book.

**System Design**

**Table Design:-**

There are various tables are present to store information-

In this project to store the data we have use database management system for we have use MySQL.

## What is Database?

The database is a collection of inter-related data which is used to retrieve, insert and delete the data efficiently. It is also used to organize the data in the form of a table, schema, views, and reports, etc.

**For example:** The college Database organizes the data about the admin, staff, students and faculty etc.

Using the database, you can easily retrieve, insert, and delete the information.

## Database Management System

* Database management system is a software which is used to manage the database. For example: [MySQL](https://www.javatpoint.com/mysql-tutorial), [Oracle](https://www.javatpoint.com/oracle-tutorial), etc are a very popular commercial database which is used in different applications.
* DBMS provides an interface to perform various operations like database creation, storing data in it, updating data, creating a table in the database and a lot more.
* It provides protection and security to the database. In the case of multiple users, it also maintains data consistency.

**DBMS allows users the following tasks:**

* **Data Definition:** It is used for creation, modification, and removal of definition that defines the organization of data in the database.
* **Data Updation:** It is used for the insertion, modification, and deletion of the actual data in the database.
* **Data Retrieval:** It is used to retrieve the data from the database which can be used by applications for various purposes.
* **User Administration:** It is used for registering and monitoring users, maintain data integrity, enforcing data security, dealing with concurrency control, monitoring performance and recovering information corrupted by unexpected failure.

## Characteristics of DBMS

* It uses a digital repository established on a server to store and manage the information.
* It can provide a clear and logical view of the process that manipulates data.
* DBMS contains automatic backup and recovery procedures.
* It contains ACID properties which maintain data in a healthy state in case of failure.
* It can reduce the complex relationship between data.
* It is used to support manipulation and processing of data.
* It is used to provide security of data.
* It can view the database from different viewpoints according to the requirements of the user.

## Advantages of DBMS

* **Controls database redundancy:** It can control data redundancy because it stores all the data in one single database file and that recorded data is placed in the database.
* **Data sharing:** In DBMS, the authorized users of an organization can share the data among multiple users.
* **Easily Maintenance:** It can be easily maintainable due to the centralized nature of the database system.
* **Reduce time:** It reduces development time and maintenance need.
* **Backup:** It provides backup and recovery subsystems which create automatic backup of data from [hardware](https://www.javatpoint.com/hardware) and [software](https://www.javatpoint.com/software) failures and restores the data if required.
* **multiple user interface:** It provides different types of user interfaces like graphical user interfaces, application program interfaces

## Disadvantages of DBMS

* **Cost of Hardware and Software:** It requires a high speed of data processor and large memory size to run DBMS software.
* **Size:** It occupies a large space of disks and large memory to run them efficiently.
* **Complexity:** Database system creates additional complexity and requirements.
* **Higher impact of failure:** Failure is highly impacted the database because in most of the organization, all the data stored in a single database and if the database is damaged due to electric failure or database corruption then the data may be lost forever.

**ER-Diagram:**

the ER Model is represented by means of an ER diagram. Any object, for example, entities, attributes of an entity, relationship sets, and attributes of relationship sets, can be represented with the help of an ER diagram.

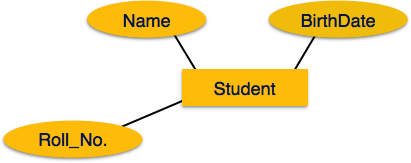
## Entity

Entities are represented by means of rectangles. Rectangles are named with the entity set they represent.

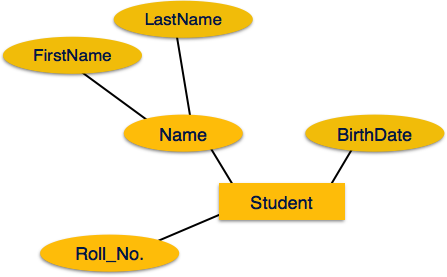
Entities in a school database

## Attributes

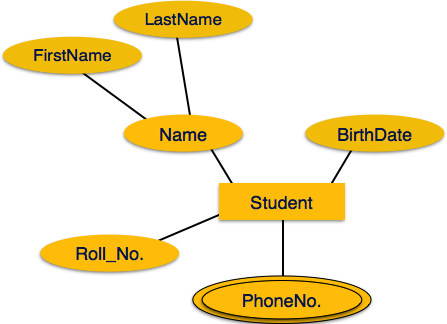
Attributes are the properties of entities. Attributes are represented by means of ellipses. Every ellipse represents one attribute and is directly connected to its entity (rectangle).



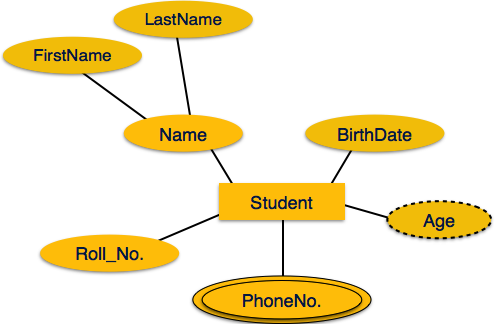
If the attributes are **composite**, they are further divided in a tree like structure. Every node is then connected to its attribute. That is, composite attributes are represented by ellipses that are connected with an ellipse.



**Multivalued** attributes are depicted by double ellipse.



**Derived** attributes are depicted by dashed ellipse.



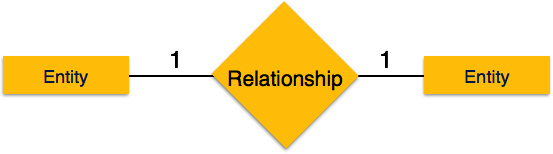
## Relationship

Relationships are represented by diamond-shaped box. Name of the relationship is written inside the diamond-box. All the entities (rectangles) participating in a relationship, are connected to it by a line.

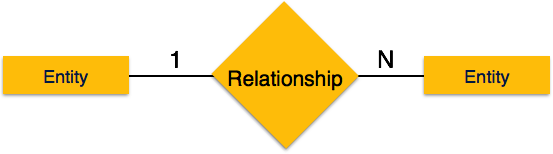
### Binary Relationship and Cardinality

A relationship where two entities are participating is called a **binary relationship**. Cardinality is the number of instance of an entity from a relation that can be associated with the relation.

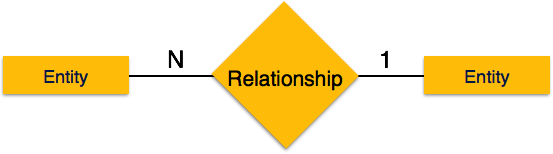
* **One-to-one** − When only one instance of an entity is associated with the relationship, it is marked as '1:1'. The following image reflects that only one instance of each entity should be associated with the relationship. It depicts one-to-one relationship.



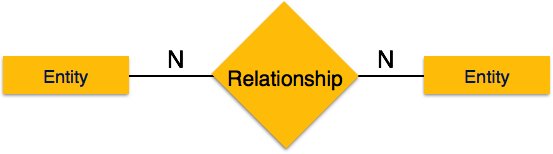
* **One-to-many** − When more than one instance of an entity is associated with a relationship, it is marked as '1:N'. The following image reflects that only one instance of entity on the left and more than one instance of an entity on the right can be associated with the relationship. It depicts one-to-many relationship.



* **Many-to-one** − When more than one instance of entity is associated with the relationship, it is marked as 'N:1'. The following image reflects that more than one instance of an entity on the left and only one instance of an entity on the right can be associated with the relationship. It depicts many-to-one relationship.

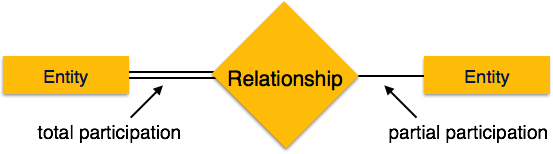


* **Many-to-many** − The following image reflects that more than one instance of an entity on the left and more than one instance of an entity on the right can be associated with the relationship. It depicts many-to-many relationship.



### Participation Constraints

* **Total Participation** − Each entity is involved in the relationship. Total participation is represented by double lines.
* **Partial participation** − Not all entities are involved in the relationship. Partial participation is represented by single lines.



* **Registration Table From Database.**

In this table there are 15 columns there names and type are –

fname VARCHAR(45) – This column is for first name of the user.

mname VARCHAR(45) – This column is for middle name of the user.

lname VARCHAR(45) – This column is for last name of the user.

gender VARCHAR(45) – This column is for selecting the gender of the user.

contact VARCHAR(45) – This column is for entering contact mobile number.

alternateNo VARCHAR(45) – This column is for entering alternate mobile number.

security VARCHAR(45) – This column is for selecting the security question through which the user can forget the password.

answer VARCHAR(45) – This column is for entering the answer of security question.

address1 VARCHAR(45) – This column is for entering the address detail of the user.

address2 VARCHAR(45) – This column is for second line of address of the user.

email VARCHAR(45) – This column is for receiving the email address of the user.

alternateEmail VARCHAR(45) - This column is for another email address of the user for the security purpose.

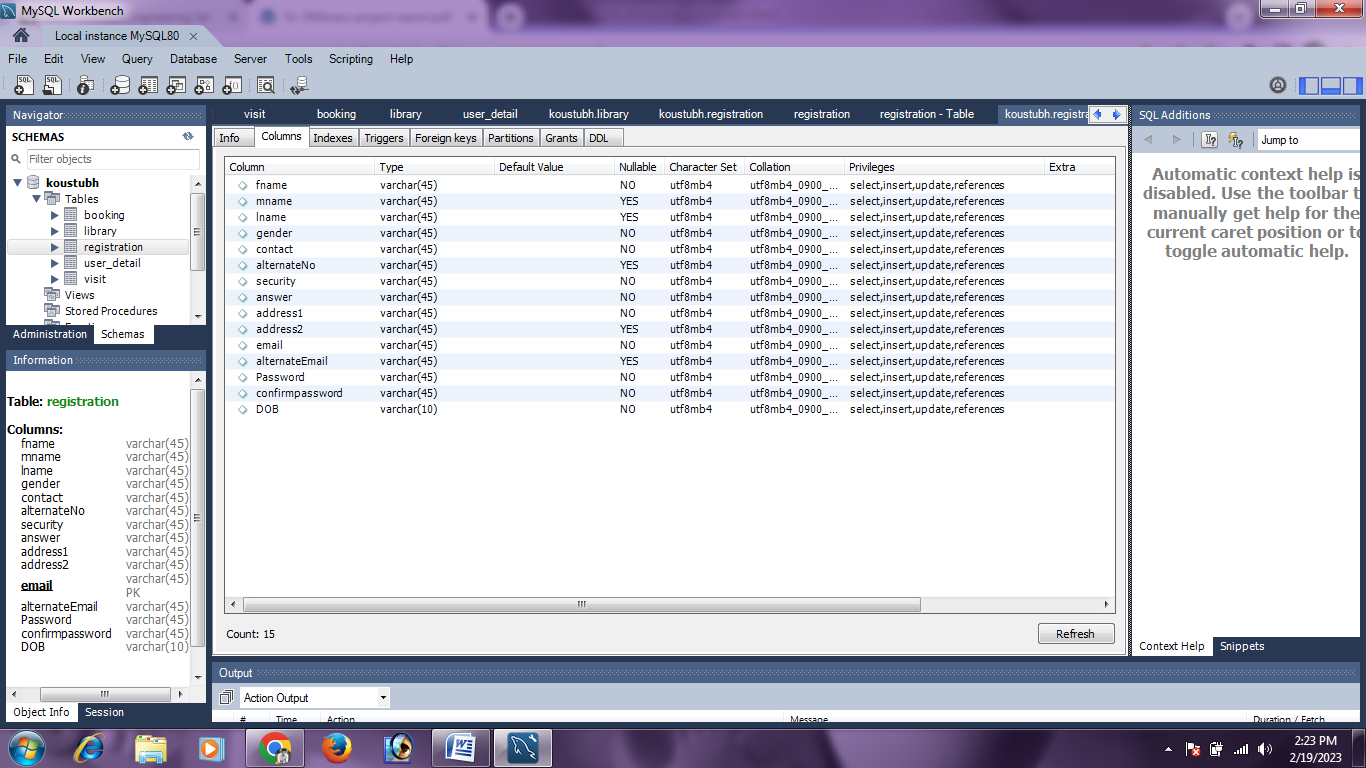
Password VARCHAR(45) – This column is for password of the user to login the page.

confirmpassword VARCHAR(45) - This is for confirming the password of the user.

DOB VARCHAR(10) – This column is to receive the date of birth of the user.

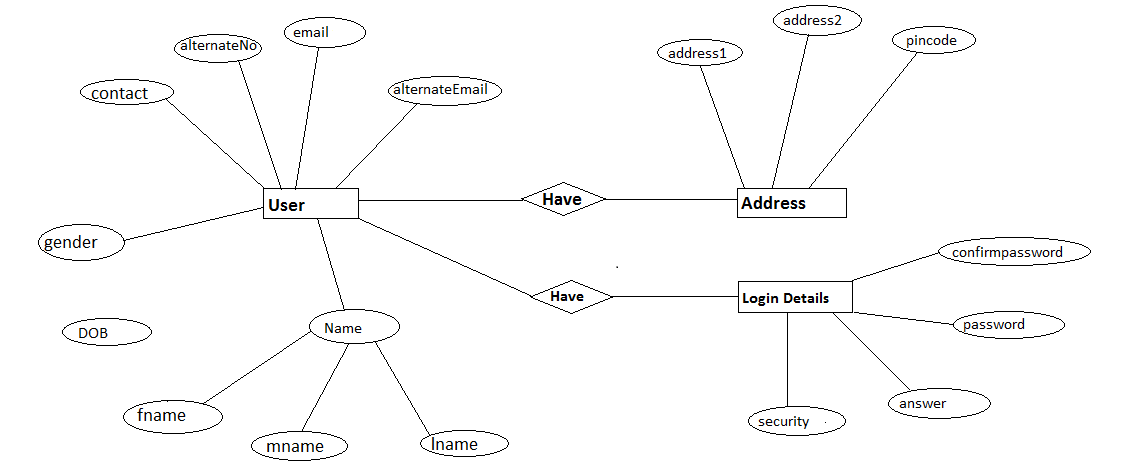
In this table the value of mname, lname, alternateNo, address2, alternateEmail can be Null, But the value of all the other columns cannot be Null.

In this table the Primary Key is email column. That means the there cannot be the duplicate value of email, It is always be unique.



**ER\_Diagram:**

The ER-Diagram for this database Table is given below:



* **Library Table from Database.**

In this table there are 18 columns there names and type are –

Member VARCHAR(45) - This column is for selecting type of user(admin, staff).

PRN\_NO VARCHAR(45) – This column is for pnr number of the user.

ID\_NO VARCHAR(45) – This column is for identity number of the user.

Firstname VARCHAR(45) – This column is for first name of the user.

Lastname VARCHAR(45) – This column is for last name of the user.

Address1 VARCHAR(45) – This column is for entering the address detail of the user.

Address2 VARCHAR(45) – This column is for second line of address of the user.

PostID VARCHAR(45) – This column is for selecting the post ID of the user.

Mobile VARCHAR(45) – This column is for entering contact mobile number.

BookID VARCHAR(45) – This column is for entering book id number.

BookTitle VARCHAR(45) – This column is for entering the Title of the book.

Auther VARCHAR(45) – This column is for entering the name of the auther.

DateBorrowed VARCHAR(45) – This column is for entering the the date of book borrowed.

DateDue VARCHAR(45) - This column is for entering the due date of the book reversal.

DaysOnBook VARCHAR(45) – This column is for entering for how many days the book issued from the library.

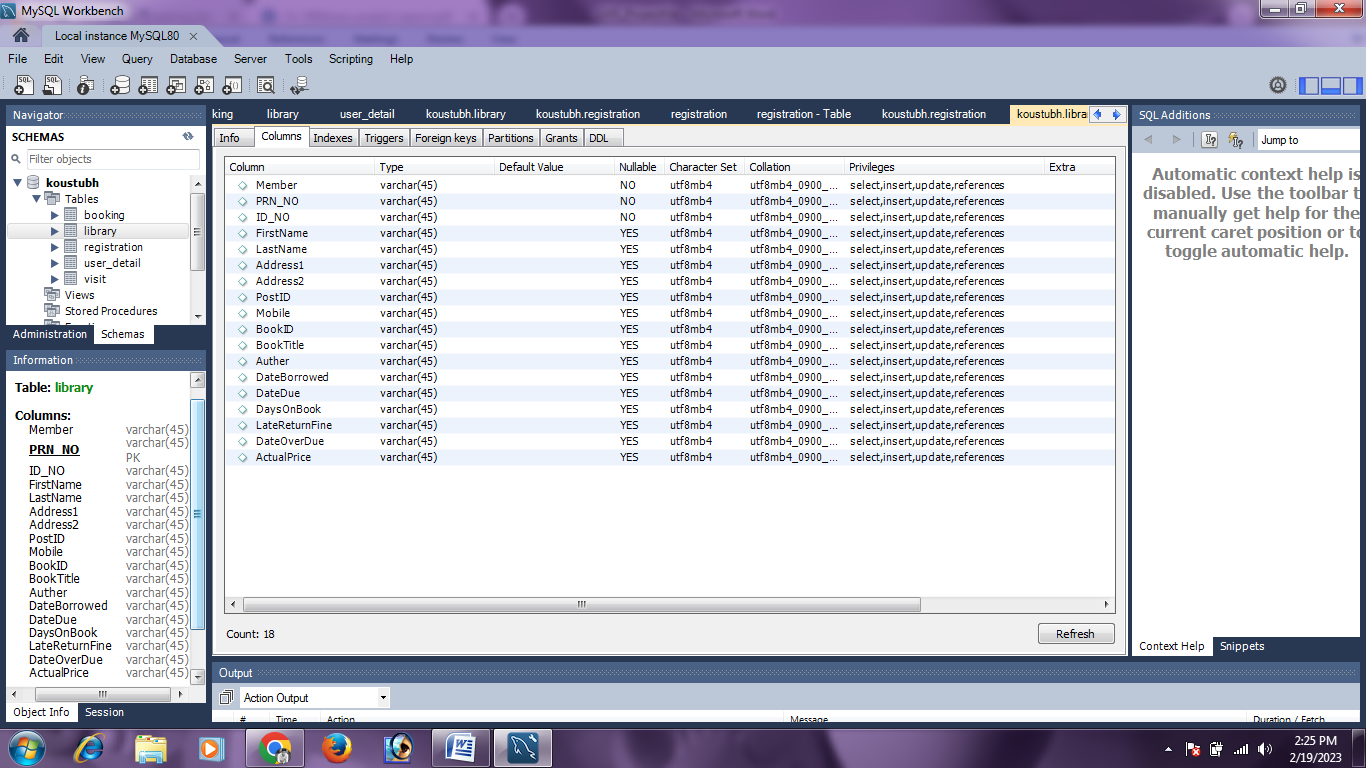
LateReturnFine VARCHAR(45) - This is for entering the late returning fine in rupees .

DateOverDue VARCHAR(45) – This column is to enter the date of due submit.

ActualPrice VARCHAR(45) – This column is for showing the price of the book.

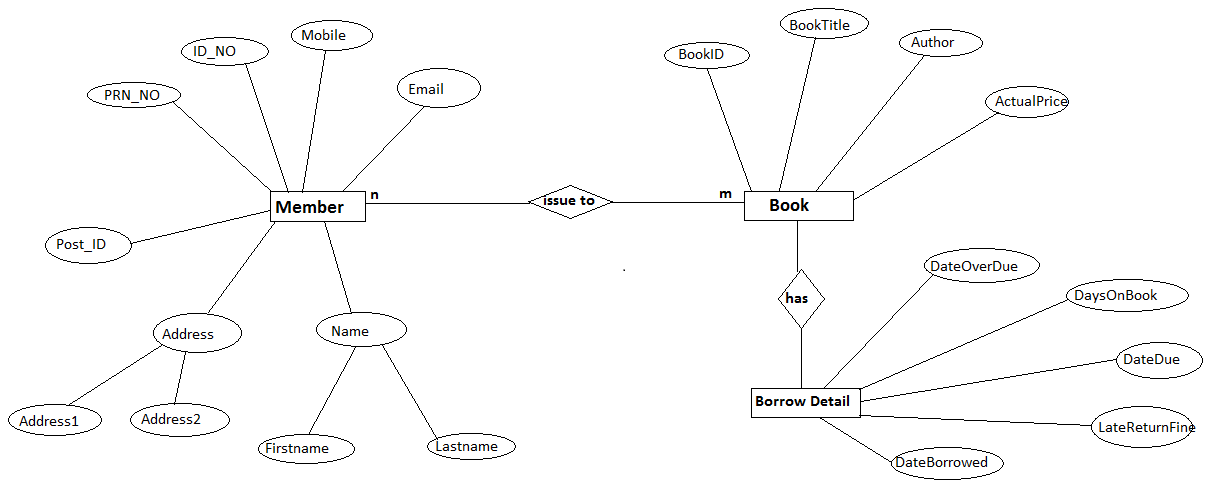
In this table the value of Member, PRN\_NO, ID\_NO cannot be Null, But the value of all the other columns can be Null.

In this table the Primary Key is PRN\_NO column. That means the there cannot be the duplicate value of PNR, It is always be unique.



**ER Diagram :**

ER-Diagram for this database table is figured below:



**Possible Outcomes:**

**a). Front Pages:**

In this Project there are three pages in which first page is for registration , second page is for login page and the last page is the main welcome page .We are sharing the screenshots of thee pages through which you can see all the features and functionality of this project.

1. **Registration Page:-**

First of all the user have to register itself through this page.

By filling these details the user can register himself/herself on this application.

After filling all the details the user have to click the submit button but before clicking on submit button the user have to check the checkbox without confirming it user cannot be process further.

If the user already have an account then he/she can click on Sign In button and then login to the application.

**Source Code for the creating the Registration Page:**

import tkinter

from tkinter import\*

from tkinter import ttk

from tkinter import messagebox

import mysql.connector

# from PIL import ImageTk

class RegistrationForm:

def \_\_init\_\_(self,root):

self.root=root

self.root.title("Registration Form")

self.root.geometry("1350x700+0+0")

# ==================================== VARIABLES ====================================================================================

self.FirstName\_var=StringVar()

self.MiddleName\_var=StringVar()

self.LastName\_var=StringVar()

self.Gender\_var=StringVar()

self.Contact\_var=StringVar()

self.AlternateNo\_var=StringVar()

self.Security\_var=StringVar()

self.Answer\_var=StringVar()

self.Address1\_var=StringVar()

self.Address2\_var=StringVar()

self.email\_var=StringVar()

self.AlternateEmail\_var=StringVar()

self.Password\_var=StringVar()

self.ConfirmPassword\_var=StringVar()

self.DOB\_var=StringVar()

self.chk\_var=IntVar()

# ==========================================FRAME=======================================================================================

lbltitle=Label(self.root,text="Registration Form",bg="salmon",fg="black",bd=0,relief=RIDGE,font=("times new roman",30,"bold"),padx=2,pady=6)

lbltitle.pack(side=TOP,fill=X)

frame=Frame(self.root,bd=0,relief=RIDGE,padx=20,bg="salmon")

frame.place(x=0,y=55,width=1350,height=700)

Insideframe=Frame(self.root,bd=10,relief=RIDGE,padx=20,bg="wheat")

Insideframe.place(x=70,y=80,width=1220,height=580)

# ==========================================LABELS======================================================================================

lblFirstName=Label(Insideframe,text="First Name:",bg="wheat",font=("times new roman",15,"bold"),padx=4,pady=10)

lblFirstName.grid(row=0,column=0,sticky=W)

txtFirstName=Entry(Insideframe,font=("times new roman",15,"bold"),textvariable=self.FirstName\_var,width=32)

txtFirstName.grid(row=0,column=1)

lblMiddleName=Label(Insideframe,text="Middle Name:",bg="wheat",font=("times new roman",15,"bold"),padx=4,pady=10)

lblMiddleName.grid(row=1,column=0,sticky=W)

txtMiddleName=Entry(Insideframe,font=("times new roman",15,"bold"),textvariable=self.MiddleName\_var,width=32)

txtMiddleName.grid(row=1,column=1)

lblLastName=Label(Insideframe,text="Last Name:",bg="wheat",font=("times new roman",15,"bold"),padx=4,pady=10)

lblLastName.grid(row=2,column=0,sticky=W)

txtLastName=Entry(Insideframe,font=("times new roman",15,"bold"),textvariable=self.LastName\_var,width=32)

txtLastName.grid(row=2,column=1)

lblGender=Label(Insideframe,text="Gender:",bg="wheat",font=("times new roman",15,"bold"),padx=4,pady=10)

lblGender.grid(row=3,column=0,sticky=W)

comGender=ttk.Combobox(Insideframe,font=("times new roman",15,"bold"),textvariable=self.Gender\_var,width=30,state="readonly",justify=CENTER)

comGender["value"]=("--------------select---------------","Male","Female","Transgender")

comGender.grid(row=3,column=1)

comGender.current(0)

lblContact=Label(Insideframe,text="Contact No:",bg="wheat",font=("times new roman",15,"bold"),padx=4,pady=10)

lblContact.grid(row=4,column=0,sticky=W)

txtContact=Entry(Insideframe,font=("times new roman",15,"bold"),textvariable=self.Contact\_var,width=32)

txtContact.grid(row=4,column=1)

lblAlternateNo=Label(Insideframe,text="Alternate Contact No:",bg="wheat",font=("times new roman",15,"bold"),padx=4,pady=10)

lblAlternateNo.grid(row=5,column=0,sticky=W)

txtAlternateNo=Entry(Insideframe,font=("times new roman",15,"bold"),textvariable=self.AlternateNo\_var,width=32)

txtAlternateNo.grid(row=5,column=1)

lblSecurity=Label(Insideframe,text="Security Question:",bg="wheat",font=("times new roman",15,"bold"),padx=4,pady=10)

lblSecurity.grid(row=6,column=0,sticky=W)

comSecurity=ttk.Combobox(Insideframe,font=("times new roman",15,"bold"),textvariable=self.Security\_var,width=30,state="readonly",justify=CENTER)

comSecurity["value"]=("--------------Select-------------","What is your favourite color?","What is your pet name?","What is your born city name?")

comSecurity.grid(row=6,column=1)

comSecurity.current(0)

lblAnswer=Label(Insideframe,text="Answer:",bg="wheat",font=("times new roman",15,"bold"),padx=4,pady=10)

lblAnswer.grid(row=7,column=0,sticky=W)

txtAnswer=Entry(Insideframe,font=("times new roman",15,"bold"),textvariable=self.Answer\_var,width=32)

txtAnswer.grid(row=7,column=1)

lblAddress1=Label(Insideframe,text="Address1:",bg="wheat",font=("times new roman",15,"bold"),padx=4,pady=10)

lblAddress1.grid(row=0,column=2,sticky=W)

txtAddress1=Entry(Insideframe,font=("times new roman",15,"bold"),textvariable=self.Address1\_var,width=32)

txtAddress1.grid(row=0,column=3)

lblAdress2=Label(Insideframe,text="Address2:",bg="wheat",font=("times new roman",15,"bold"),padx=4,pady=10)

lblAdress2.grid(row=1,column=2,sticky=W)

txtAdress2=Entry(Insideframe,font=("times new roman",15,"bold"),textvariable=self.Address2\_var,width=32)

txtAdress2.grid(row=1,column=3)

lblemail=Label(Insideframe,text="Email:",bg="wheat",font=("times new roman",15,"bold"),padx=4,pady=10)

lblemail.grid(row=2,column=2,sticky=W)

txtemail=Entry(Insideframe,font=("times new roman",15,"bold"),textvariable=self.email\_var,width=32)

txtemail.grid(row=2,column=3)

lblAlternateEmail=Label(Insideframe,text="Alternate Email:",bg="wheat",font=("times new roman",15,"bold"),padx=4,pady=10)

lblAlternateEmail.grid(row=3,column=2,sticky=W)

txtAlternateEmail=Entry(Insideframe,font=("times new roman",15,"bold"),textvariable=self.AlternateEmail\_var,width=32)

txtAlternateEmail.grid(row=3,column=3)

lblPassword=Label(Insideframe,text="Password:",bg="wheat",font=("times new roman",15,"bold"),padx=4,pady=10)

lblPassword.grid(row=4,column=2,sticky=W)

txtPassword=Entry(Insideframe,font=("times new roman",15,"bold"),textvariable=self.Password\_var,width=32)

txtPassword.grid(row=4,column=3)

lblConfirmPassword=Label(Insideframe,text="Confirm Password:",bg="wheat",font=("times new roman",15,"bold"),padx=4,pady=10)

lblConfirmPassword.grid(row=5,column=2,sticky=W)

txtConfirmPassword=Entry(Insideframe,font=("times new roman",15,"bold"),textvariable=self.ConfirmPassword\_var,width=32)

txtConfirmPassword.grid(row=5,column=3)

lblDOB=Label(Insideframe,text="DOB(DD/MM/YYYY):",bg="wheat",font=("times new roman",15,"bold"),padx=4,pady=10)

lblDOB.grid(row=6,column=2,sticky=W)

txtDOB=Entry(Insideframe,font=("times new roman",15,"bold"),textvariable=self.DOB\_var,width=32)

txtDOB.grid(row=6,column=3)

# ======================================BUTTON WORK==========================================================================================

chk=Checkbutton(Insideframe,text="I Agree The Terms & Conditions",bg="wheat",font=("times new roman",12,"bold"),variable=self.chk\_var,onvalue=1,offvalue=0).place(x=200,y=430)

btnSubmitData=Button(Insideframe,command=self.register\_data,text="Submit",font=("times new roman",12,"bold"),width=26,bg="chocolate",fg="white")

btnSubmitData.place(x=200,y=460)

btnResetData=Button(Insideframe,command=self.resetData,text="Reset",font=("times new roman",12,"bold"),width=26,bg="chocolate",fg="white")

btnResetData.place(x=500,y=460)

btnCancel=Button(Insideframe,command=self.cancel,text="Cancel",font=("times new roman",12,"bold"),width=26,bg="chocolate",fg="white")

btnCancel.place(x=800,y=460)

btnSignin=Button(Insideframe,command=self.Sign\_In,text="Sign In",font=("times new roman",12,"bold"),width=26,bg="navy",fg="white")

btnSignin.place(x=500,y=520)

# =============================================== MYSQL ======================================================================

def register\_data(self):

if (self.FirstName\_var.get()=="" or self.Address1\_var.get()=="" or self.Gender\_var.get()=="" or self.Contact\_var.get()=="" or self.Security\_var.get()=="--------------Select-------------" or self.email\_var.get()=="" or self.Password\_var.get()=="" or self.ConfirmPassword\_var.get()=="" or self.Answer\_var.get()=="" or self.DOB\_var.get()==""):

messagebox.showerror("Error","All Fields Are Mandatory")

elif self.Password\_var.get()!=self.ConfirmPassword\_var.get():

messagebox.showerror("Error","Password Not Match!")

elif self.chk\_var.get()==0:

messagebox.showerror("Error","Please Check The Terms and Condition")

else:

cnx=mysql.connector.connect(user="root",password="parsaiji",host="localhost",database="koustubh")

my\_cursor=cnx.cursor()

my\_cursor.execute("insert into registration values(%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s)",(

self.FirstName\_var.get(),

self.MiddleName\_var.get(),

self.LastName\_var.get(),

self.Gender\_var.get(),

self.Contact\_var.get(),

self.AlternateNo\_var.get(),

self.Security\_var.get(),

self.Answer\_var.get(),

self.Address1\_var.get(),

self.Address2\_var.get(),

self.email\_var.get(),

self.AlternateEmail\_var.get(),

self.Password\_var.get(),

self.ConfirmPassword\_var.get(),

self.DOB\_var.get(),

))

cnx.commit()

cnx.close()

messagebox.showinfo("success","Data Register Successfully")

def resetData(self):

self.FirstName\_var.set(""),

self.MiddleName\_var.set(""),

self.LastName\_var.set(""),

self.Contact\_var.set(""),

self.AlternateNo\_var.set(""),

self.Security\_var.set(""),

self.Answer\_var.set(""),

self.Address1\_var.set(""),

self.Address2\_var.set(""),

self.email\_var.set(""),

self.AlternateEmail\_var.set(""),

self.Password\_var.set(""),

self.ConfirmPassword\_var.set(""),

self.DOB\_var.set(""),

self.textBox.delete("1.0",END)

def cancel(self):

cancel=tkinter.messagebox.askyesno("Registration Form","Do You Want To Cancel?")

if cancel>0:

self.root.destroy()

return

def Sign\_In(self):

self.root.destroy()

import login

root = Tk()

obj=RegistrationForm(root)

root.mainloop()

**The Output Of The Following Source Code Is Shown below :-**

****

**After Inserting The Data To All The Fields:-**

****

1. **Login Page:-**

In this page the user can login by entering the login details the user name is same as the email address and the password is same that the user provide it at the time of registration. After filling the details user have to click on the login button. If the user don’t have login details then he/she can first register itself.

**Source Code for the creating the Registration Page:**

from msilib.schema import CheckBox

from tkinter import \*

from tkinter import ttk

import tkinter

from tkinter import messagebox

import mysql.connector

class LogIn:

def \_\_init\_\_(self,root):

self.root=root

self.root.title("Login Page")

self.root.geometry("1350x700+0+0")

# ============================================== VARIABLES ===============================================================================

self.UserName\_var=StringVar()

self.Password\_var=StringVar()

self.chk\_var=IntVar()

# ============================================ FRAME + LABEL =============================================================================

lbltitle=Label(self.root,text="Welcome !",bg="gray",fg="black",bd=0,relief=RIDGE,font=("italian",35,"bold"),padx=2,pady=6)

lbltitle.pack(side=TOP,fill=X)

frame=Frame(self.root,bd=0,relief=RIDGE,padx=20,bg="gray")

frame.place(x=0,y=55,width=1350,height=700)

Insideframe=Frame(self.root,bd=10,relief=RIDGE,padx=20,bg="tan")

Insideframe.place(x=70,y=80,width=1220,height=580)

lbltitle=Label(self.root,text="If Already Have An Account Please Login",bg="tan",fg="black",bd=0,relief=RIDGE,font=("italian",15,"bold"),padx=2,pady=4)

lbltitle.pack(side=TOP,fill=X)

lbltitle.place(x=400,y=120,width=500)

# =================================================== LABELS =============================================================================

lblUserName=Label(Insideframe,text="User Name:",bg="tan",font=("times new roman",15,"bold"),padx=4,pady=10)

lblUserName.place(x=320,y=170)

txtUserName=Entry(Insideframe,textvariable=self.UserName\_var,font=("times new roman",15,"bold"),width=32)

txtUserName.place(x=450,y=180)

lblPassword=Label(Insideframe,text="Password:",bg="tan",font=("times new roman",15,"bold"),padx=4,pady=10)

lblPassword.place(x=320,y=230)

txtPassword=Entry(Insideframe,show='\*',textvariable=self.Password\_var,font=("times new roman",15,"bold"),width=32)

txtPassword.place(x=450,y=240)

# =================================================== BUTTONS ============================================================================

btnSignUp=Button(Insideframe,command=self.Sign\_Up,text="Sign Up",font=("times new roman",15,"bold"),width=15,bg="chocolate",fg="white")

btnSignUp.place(x=400,y=350)

btnLogin=Button(Insideframe,command=self.login,text="Login",font=("times new roman",15,"bold"),width=15,bg="chocolate",fg="white")

btnLogin.place(x=600,y=350)

# ==================================================== FUNCTIONING OF BUTTONS =============================================================

def Sign\_Up(self):

self.root.destroy()

import registration

def login(self):

if self.UserName\_var.get()=="" or self.Password\_var.get()=="":

messagebox.showerror("Error","Please Enter User Name and Password",parent=self.root)

else:

cnx=mysql.connector.connect(user="root",password="parsaiji",host="localhost",database="koustubh")

my\_cursor=cnx.cursor()

my\_cursor.execute("select \* from registration where email=%s and Password=%s",(self.UserName\_var.get(),self.Password\_var.get()))

row=my\_cursor.fetchone()

if row==None:

messagebox.showerror("Error","Invalid User Name or Password",parent=self.root)

else:

self.root.destroy()

import project

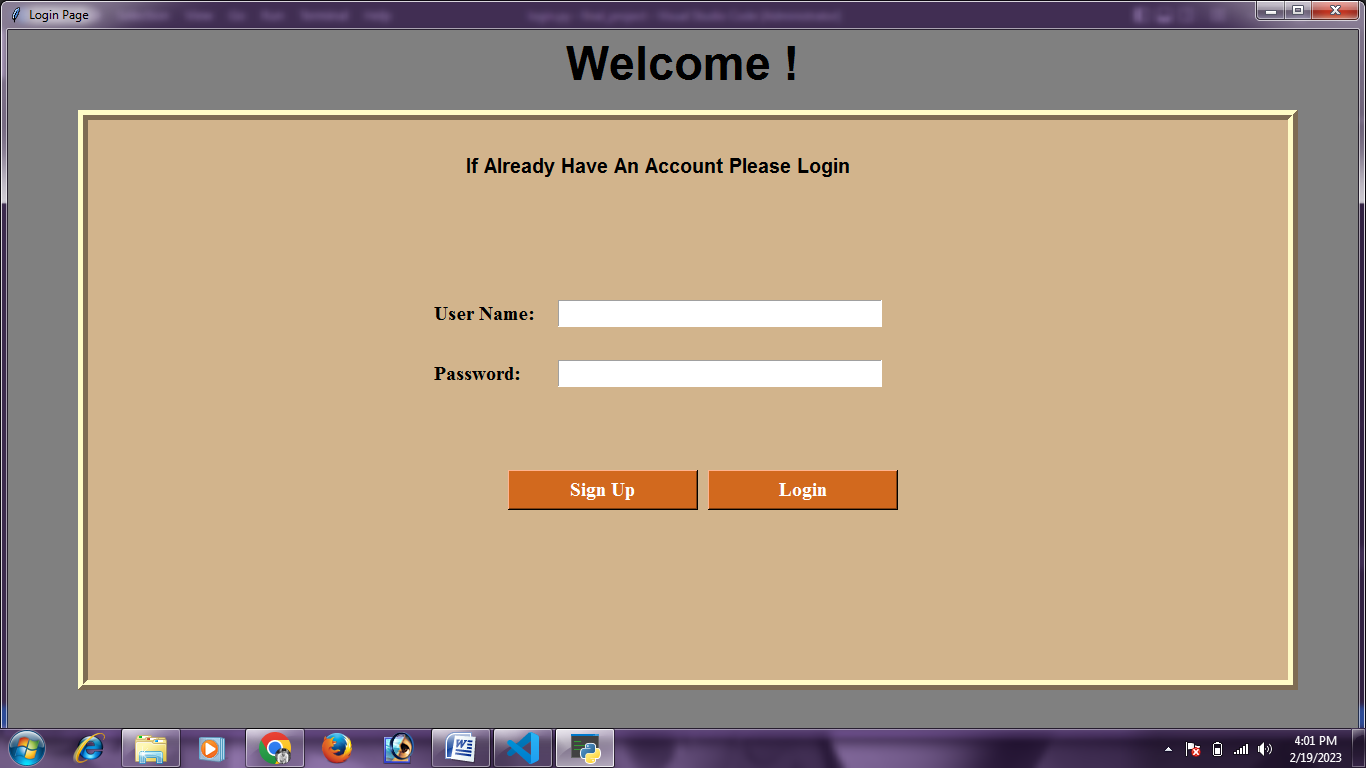
cnx.close()

root = Tk()

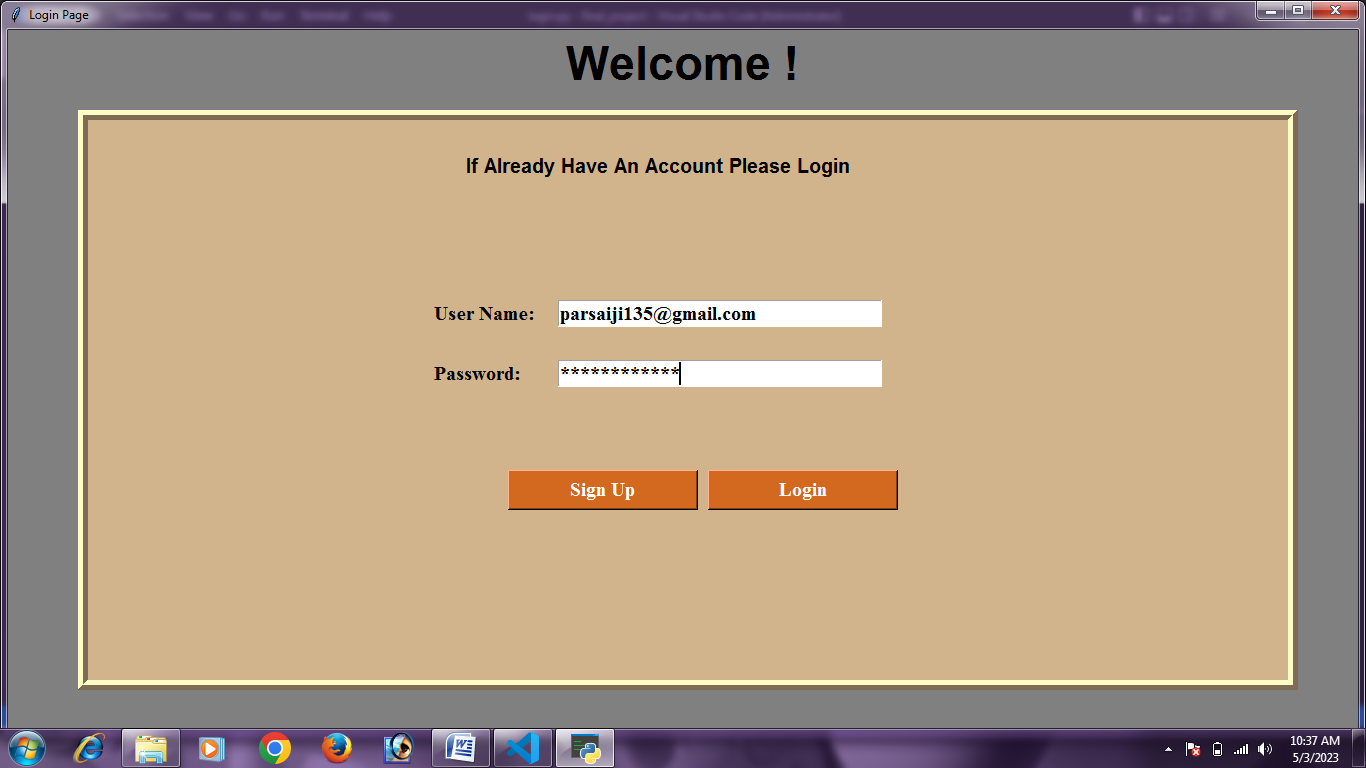
obj=LogIn(root)

root.mainloop()

**The Output Of The Following Source Code Is Shown below :-**

****

**After Inserting The Data To All The Fields:-**

****

1. **Main Page:-**

In this page the user have all the details of book like title of books, author name, price of book etc. through this page application the user can borrow any book from the library. All the details of the user are saved in the database through which it is easy to get the details of users and the detail like which user borrow which book and how many days before he/she borrow that book and when will the last date to return that book.

**Source Code for the creating the Registration Page:**

# from msilib.schema import ListBox

from msilib.schema import ListBox

from tkinter import\*

from tkinter import ttk

import mysql.connector

from tkinter import messagebox

import datetime

import tkinter

class LibraryManagementSystem:

#==========================OUT BOX=======================================

def \_\_init\_\_(self,root):

self.root=root

self.root.title("Library Management System")

self.root.geometry("1350x700+0+0")

#=============================VARIABLE=======================================================================

self.member\_var=StringVar()

self.prn\_var=StringVar()

self.id\_var=StringVar()

self.firstname\_var=StringVar()

self.lastname\_var=StringVar()

self.address1\_var=StringVar()

self.address2\_var=StringVar()

self.postcode\_var=StringVar()

self.mobile\_var=StringVar()

self.bookid\_var=StringVar()

self.booktitle\_var=StringVar()

self.auther\_var=StringVar()

self.dateborrowed\_var=StringVar()

self.datedue\_var=StringVar()

self.daysonbook\_var=StringVar()

self.latereturnfine\_var=StringVar()

self.dateoverdue\_var=StringVar()

self.actualprice\_var=StringVar()

#==========================TITLE BOX=====================================

lbltitle=Label(self.root,text="LIBRARY MANAGEMENT SYSTEM",bg="sky blue",fg="red",bd=20,relief=RIDGE,font=("times new roman",50,"bold"),padx=2,pady=6)

lbltitle.pack(side=TOP,fill=X)

frame=Frame(self.root,bd=10,relief=RIDGE,padx=20,bg="sky blue")

frame.place(x=0,y=130,width=1350,height=400)

#==========================DATA FRAME LEFT=========================

DataFrameLeft=LabelFrame(frame,text="Library Membership Information",bg="sky blue",fg="red",bd=12,relief=RIDGE,font=("times new roman",20,"bold"))

DataFrameLeft.place(x=0,y=0,width=750,height=360)

lblMember=Label(DataFrameLeft,text="Member Type",bg="sky blue",font=("times new roman",12,"bold"),padx=2,pady=6)

lblMember.grid(row=0,column=0,sticky=W)

comMember=ttk.Combobox(DataFrameLeft,font=("times new roman",12,"bold"),textvariable=self.member\_var,width=25,state="readonly",justify=CENTER)

comMember["value"]=("-------------Select------------","Admin Staff","Student","Teachers")

comMember.grid(row=0,column=1)

comMember.current(0)

lblPRN\_No=Label(DataFrameLeft,text="PRN No:",bg="sky blue",font=("times new roman",12,"bold"),padx=2)

lblPRN\_No.grid(row=1,column=0,sticky=W)

txtPRN\_No=Entry(DataFrameLeft,font=("times new roman",12,"bold"),textvariable=self.prn\_var,width=27)

txtPRN\_No.grid(row=1,column=1)

lblTitle=Label(DataFrameLeft,text="ID No:",bg="sky blue",font=("times new roman",12,"bold"),padx=2,pady=4)

lblTitle.grid(row=2,column=0,sticky=W)

txtTitle=Entry(DataFrameLeft,font=("times new roman",12,"bold"),textvariable=self.id\_var,width=27)

txtTitle.grid(row=2,column=1)

lblFirstName=Label(DataFrameLeft,text="First Name:",bg="sky blue",font=("times new roman",12,"bold"),padx=2,pady=6)

lblFirstName.grid(row=3,column=0,sticky=W)

txtFirstName=Entry(DataFrameLeft,font=("times new roman",12,"bold"),textvariable=self.firstname\_var,width=27)

txtFirstName.grid(row=3,column=1)

lblLastName=Label(DataFrameLeft,text="Last Name:",bg="sky blue",font=("times new roman",12,"bold"),padx=2,pady=6)

lblLastName.grid(row=4,column=0,sticky=W)

txtLastName=Entry(DataFrameLeft,font=("times new roman",12,"bold"),textvariable=self.lastname\_var,width=27)

txtLastName.grid(row=4,column=1)

lblAddress1=Label(DataFrameLeft,text="Address1:",bg="sky blue",font=("times new roman",12,"bold"),padx=2,pady=6)

lblAddress1.grid(row=5,column=0,sticky=W)

txtAddress1=Entry(DataFrameLeft,font=("times new roman",12,"bold"),textvariable=self.address1\_var,width=27)

txtAddress1.grid(row=5,column=1)

lblAddress2=Label(DataFrameLeft,text="Address2:",bg="sky blue",font=("times new roman",12,"bold"),padx=2,pady=6)

lblAddress2.grid(row=6,column=0,sticky=W)

txtAddress2=Entry(DataFrameLeft,font=("times new roman",12,"bold"),textvariable=self.address2\_var,width=27)

txtAddress2.grid(row=6,column=1)

lblPostCode=Label(DataFrameLeft,text="Post Code:",bg="sky blue",font=("times new roman",12,"bold"),padx=2,pady=6)

lblPostCode.grid(row=7,column=0,sticky=W)

txtPostCode=Entry(DataFrameLeft,font=("times new roman",12,"bold"),textvariable=self.postcode\_var,width=27)

txtPostCode.grid(row=7,column=1)

lblMobile=Label(DataFrameLeft,text="Mobile:",bg="sky blue",font=("times new roman",12,"bold"),padx=2,pady=6)

lblMobile.grid(row=8,column=0,sticky=W)

txtMobile=Entry(DataFrameLeft,font=("times new roman",12,"bold"),textvariable=self.mobile\_var,width=27)

txtMobile.grid(row=8,column=1)

lblBookId=Label(DataFrameLeft,text="Book ID:",bg="sky blue",font=("times new roman",12,"bold"),padx=2,pady=6)

lblBookId.grid(row=0,column=2,sticky=W)

txtBookId=Entry(DataFrameLeft,font=("times new roman",12,"bold"),textvariable=self.bookid\_var,width=27)

txtBookId.grid(row=0,column=3)

lblBookTitle=Label(DataFrameLeft,text="Book Title:",bg="sky blue",font=("times new roman",12,"bold"),padx=2,pady=6)

lblBookTitle.grid(row=1,column=2,sticky=W)

txtBookTitle=Entry(DataFrameLeft,font=("times new roman",12,"bold"),textvariable=self.booktitle\_var,width=27)

txtBookTitle.grid(row=1,column=3)

lblAuthor=Label(DataFrameLeft,text="Author Name:",bg="sky blue",font=("times new roman",12,"bold"),padx=2,pady=6)

lblAuthor.grid(row=2,column=2,sticky=W)

txtAuthor=Entry(DataFrameLeft,font=("times new roman",12,"bold"),textvariable=self.auther\_var,width=27)

txtAuthor.grid(row=2,column=3)

lblDateBorrowed=Label(DataFrameLeft,text="Date Borrowed:",bg="sky blue",font=("times new roman",12,"bold"),padx=2,pady=6)

lblDateBorrowed.grid(row=3,column=2,sticky=W)

txtDateBorrowed=Entry(DataFrameLeft,font=("times new roman",12,"bold"),textvariable=self.dateborrowed\_var,width=27)

txtDateBorrowed.grid(row=3,column=3)

lblDateDue=Label(DataFrameLeft,text="Date Due:",bg="sky blue",font=("times new roman",12,"bold"),padx=2,pady=6)

lblDateDue.grid(row=4,column=2,sticky=W)

txtDateDue=Entry(DataFrameLeft,font=("times new roman",12,"bold"),textvariable=self.datedue\_var,width=27)

txtDateDue.grid(row=4,column=3)

lblDaysOnBook=Label(DataFrameLeft,text="Days On Book:",bg="sky blue",font=("times new roman",12,"bold"),padx=2,pady=6)

lblDaysOnBook.grid(row=5,column=2,sticky=W)

txtDaysOnBook=Entry(DataFrameLeft,font=("times new roman",12,"bold"),textvariable=self.daysonbook\_var,width=27)

txtDaysOnBook.grid(row=5,column=3)

lblLateReturnFine=Label(DataFrameLeft,text="Late Return Fine:",bg="sky blue",font=("times new roman",12,"bold"),padx=2,pady=6)

lblLateReturnFine.grid(row=6,column=2,sticky=W)

txtLateReturnFine=Entry(DataFrameLeft,font=("times new roman",12,"bold"),textvariable=self.latereturnfine\_var,width=27)

txtLateReturnFine.grid(row=6,column=3)

lblDateOverDue=Label(DataFrameLeft,text="Date Over Due:",bg="sky blue",font=("times new roman",12,"bold"),padx=2,pady=6)

lblDateOverDue.grid(row=7,column=2,sticky=W)

txtDateOverDue=Entry(DataFrameLeft,font=("times new roman",12,"bold"),textvariable=self.dateoverdue\_var,width=27)

txtDateOverDue.grid(row=7,column=3)

lblActualPrice=Label(DataFrameLeft,text="Actual Price:",bg="sky blue",font=("times new roman",12,"bold"),padx=2,pady=6)

lblActualPrice.grid(row=8,column=2,sticky=W)

txtActualPrice=Entry(DataFrameLeft,font=("times new roman",12,"bold"),textvariable=self.actualprice\_var,width=27)

txtActualPrice.grid(row=8,column=3)

#==========================DATA FRAME RIGHT==========================

DataFrameRight=LabelFrame(frame,text="Book Details",bg="sky blue",fg="red",bd=12,relief=RIDGE,font=("times new roman",20,"bold"))

DataFrameRight.place(x=755,y=0,width=540,height=360)

self.textBox=Text(DataFrameRight,font=("times new roman",12,"bold"),width=30,height=15,padx=2,pady=6)

self.textBox.grid(row=0,column=2)

listScrollBar=Scrollbar(DataFrameRight)

listScrollBar.grid(row=0,column=1,sticky="ns")

listBooks=["Ulysses","The Great Gateway","Lolita","Brave New World","The Sound and The Fury","Catch-22",

"Under The Volcano","The Way Of All Flesh","Hello World","Evil Return",

"Bahubali:The Beginning","Bahubali:The Conclusion","RRR","Pushpa",

"KGF Chapter1","KGF Chapter2","Adipurush"]

def SelectBook(event=""):

value=str(listBox.get(listBox.curselection()))

x=value

if (x=="Ulysses"):

self.bookid\_var.set("1020UL40")

self.booktitle\_var.set("English Poem")

self.auther\_var.set("Alfred Tennyson")

d1=datetime.date.today()

d2=datetime.timedelta(days=15)

d3=d1+d2

self.dateborrowed\_var.set(d1)

self.datedue\_var.set(d3)

self.daysonbook\_var.set(d2)

self.latereturnfine\_var.set("Rs.30")

self.dateoverdue\_var.set("NO")

self.actualprice\_var.set("Rs.499")

elif (x=="The Great Gateway"):

self.bookid\_var.set("1020TGG40")

self.booktitle\_var.set("Historical Story")

self.auther\_var.set("DayaRam Sahani")

d1=datetime.date.today()

d2=datetime.timedelta(days=15)

d3=d1+d2

self.dateborrowed\_var.set(d1)

self.datedue\_var.set(d3)

self.daysonbook\_var.set(d2)

self.latereturnfine\_var.set("Rs.30")

self.dateoverdue\_var.set("NO")

self.actualprice\_var.set("Rs.549")

elif (x=="Lolita"):

self.bookid\_var.set("1020L40")

self.booktitle\_var.set("English Story")

self.auther\_var.set("Lalit Vyas")

d1=datetime.date.today()

d2=datetime.timedelta(days=15)

d3=d1+d2

self.dateborrowed\_var.set(d1)

self.datedue\_var.set(d3)

self.daysonbook\_var.set(d2)

self.latereturnfine\_var.set("Rs.30")

self.dateoverdue\_var.set("NO")

self.actualprice\_var.set("Rs.339")

elif (x=="Brave New World"):

self.bookid\_var.set("1020BNW40")

self.booktitle\_var.set("Thriller Nobel")

self.auther\_var.set("Vishal Prajapati")

d1=datetime.date.today()

d2=datetime.timedelta(days=15)

d3=d1+d2

self.dateborrowed\_var.set(d1)

self.datedue\_var.set(d3)

self.daysonbook\_var.set(d2)

self.latereturnfine\_var.set("Rs.30")

self.dateoverdue\_var.set("NO")

self.actualprice\_var.set("Rs.429")

elif (x=="The Sound and The Fury"):

self.bookid\_var.set("1020TSATF40")

self.booktitle\_var.set("Horrer Story")

self.auther\_var.set("Premlata Pandit")

d1=datetime.date.today()

d2=datetime.timedelta(days=15)

d3=d1+d2

self.dateborrowed\_var.set(d1)

self.datedue\_var.set(d3)

self.daysonbook\_var.set(d2)

self.latereturnfine\_var.set("Rs.30")

self.dateoverdue\_var.set("NO")

self.actualprice\_var.set("Rs.299")

elif (x=="Catch-22"):

self.bookid\_var.set("1020C2240")

self.booktitle\_var.set("Spy Real Story")

self.auther\_var.set("Shivam Sharma")

d1=datetime.date.today()

d2=datetime.timedelta(days=15)

d3=d1+d2

self.dateborrowed\_var.set(d1)

self.datedue\_var.set(d3)

self.daysonbook\_var.set(d2)

self.latereturnfine\_var.set("Rs.30")

self.dateoverdue\_var.set("NO")

self.actualprice\_var.set("Rs.289")

elif (x=="Under The Volcano"):

self.bookid\_var.set("1020UTV40")

self.booktitle\_var.set("Science Fiction")

self.auther\_var.set("Rohit Shrivastava")

d1=datetime.date.today()

d2=datetime.timedelta(days=15)

d3=d1+d2

self.dateborrowed\_var.set(d1)

self.datedue\_var.set(d3)

self.daysonbook\_var.set(d2)

self.latereturnfine\_var.set("Rs.30")

self.dateoverdue\_var.set("NO")

self.actualprice\_var.set("Rs.586")

elif (x=="The Way Of All Flesh"):

self.bookid\_var.set("1020TWOAF40")

self.booktitle\_var.set("Japanise Book")

self.auther\_var.set("Kartik Jain")

d1=datetime.date.today()

d2=datetime.timedelta(days=15)

d3=d1+d2

self.dateborrowed\_var.set(d1)

self.datedue\_var.set(d3)

self.daysonbook\_var.set(d2)

self.latereturnfine\_var.set("Rs.30")

self.dateoverdue\_var.set("NO")

self.actualprice\_var.set("Rs.426")

elif (x=="Hello World"):

self.bookid\_var.set("1020HW40")

self.booktitle\_var.set("Comedy")

self.auther\_var.set("Shivanshu Bhurtiya")

d1=datetime.date.today()

d2=datetime.timedelta(days=15)

d3=d1+d2

self.dateborrowed\_var.set(d1)

self.datedue\_var.set(d3)

self.daysonbook\_var.set(d2)

self.latereturnfine\_var.set("Rs.30")

self.dateoverdue\_var.set("NO")

self.actualprice\_var.set("Rs.228")

elif (x=="Evil Return"):

self.bookid\_var.set("1020ER40")

self.booktitle\_var.set("Horrer")

self.auther\_var.set("Sharad Sharma")

d1=datetime.date.today()

d2=datetime.timedelta(days=15)

d3=d1+d2

self.dateborrowed\_var.set(d1)

self.datedue\_var.set(d3)

self.daysonbook\_var.set(d2)

self.latereturnfine\_var.set("Rs.30")

self.dateoverdue\_var.set("NO")

self.actualprice\_var.set("Rs.486")

elif (x=="Bahubali:The Beginning"):

self.bookid\_var.set("1020BTB40")

self.booktitle\_var.set("Historical")

self.auther\_var.set("S.S.Rajamouli")

d1=datetime.date.today()

d2=datetime.timedelta(days=15)

d3=d1+d2

self.dateborrowed\_var.set(d1)

self.datedue\_var.set(d3)

self.daysonbook\_var.set(d2)

self.latereturnfine\_var.set("Rs.30")

self.dateoverdue\_var.set("NO")

self.actualprice\_var.set("Rs.999")

elif (x=="Bahubali:The Conclusion"):

self.bookid\_var.set("1020BTC40")

self.booktitle\_var.set("Greatest Of 2019")

self.auther\_var.set("S.S.Rajamouli")

d1=datetime.date.today()

d2=datetime.timedelta(days=15)

d3=d1+d2

self.dateborrowed\_var.set(d1)

self.datedue\_var.set(d3)

self.daysonbook\_var.set(d2)

self.latereturnfine\_var.set("Rs.30")

self.dateoverdue\_var.set("NO")

self.actualprice\_var.set("Rs.1299")

elif (x=="RRR"):

self.bookid\_var.set("1020RRR40")

self.booktitle\_var.set("Action Nobel")

self.auther\_var.set("V.Vijayandra Prasad")

d1=datetime.date.today()

d2=datetime.timedelta(days=15)

d3=d1+d2

self.dateborrowed\_var.set(d1)

self.datedue\_var.set(d3)

self.daysonbook\_var.set(d2)

self.latereturnfine\_var.set("Rs.30")

self.dateoverdue\_var.set("NO")

self.actualprice\_var.set("Rs.899")

elif (x=="Pushpa"):

self.bookid\_var.set("1020P40")

self.booktitle\_var.set("Action+Comedy")

self.auther\_var.set("Sukumar")

d1=datetime.date.today()

d2=datetime.timedelta(days=15)

d3=d1+d2

self.dateborrowed\_var.set(d1)

self.datedue\_var.set(d3)

self.daysonbook\_var.set(d2)

self.latereturnfine\_var.set("Rs.30")

self.dateoverdue\_var.set("NO")

self.actualprice\_var.set("Rs.789")

elif (x=="KGF Chapter1"):

self.bookid\_var.set("1020KGF140")

self.booktitle\_var.set("Action+Suspence")

self.auther\_var.set("Prashanth Neel")

d1=datetime.date.today()

d2=datetime.timedelta(days=15)

d3=d1+d2

self.dateborrowed\_var.set(d1)

self.datedue\_var.set(d3)

self.daysonbook\_var.set(d2)

self.latereturnfine\_var.set("Rs.30")

self.dateoverdue\_var.set("NO")

self.actualprice\_var.set("Rs.1599")

elif (x=="KGF Chapter2"):

self.bookid\_var.set("1020KGF240")

self.booktitle\_var.set("Action+Historical")

self.auther\_var.set("Prashanth Neel")

d1=datetime.date.today()

d2=datetime.timedelta(days=15)

d3=d1+d2

self.dateborrowed\_var.set(d1)

self.datedue\_var.set(d3)

self.daysonbook\_var.set(d2)

self.latereturnfine\_var.set("Rs.30")

self.dateoverdue\_var.set("NO")

self.actualprice\_var.set("Rs.228")

elif (x=="Adipurush"):

self.bookid\_var.set("1020AD40")

self.booktitle\_var.set("Epic")

self.auther\_var.set("Prabhas")

d1=datetime.date.today()

d2=datetime.timedelta(days=15)

d3=d1+d2

self.dateborrowed\_var.set(d1)

self.datedue\_var.set(d3)

self.daysonbook\_var.set(d2)

self.latereturnfine\_var.set("Rs.30")

self.dateoverdue\_var.set("NO")

self.actualprice\_var.set("Rs.1987")

listBox=Listbox(DataFrameRight,font=("times new roman",12,"bold"),width=20,height=15)

listBox.bind("<<ListboxSelect>>",SelectBook)

listBox.grid(row=0,column=0,padx=4)

listScrollBar.config(command=listBox.yview)

for item in listBooks:

listBox.insert(END,item)

#==========================BUTTONS FRAME=================================

FrameButton=Frame(self.root,bd=8,relief=RIDGE,padx=0,bg="sky blue")

FrameButton.place(x=0,y=530,width=1350,height=50)

btnAddData=Button(FrameButton,command=self.add\_data,text="Add Data",font=("times new roman",12,"bold"),width=24,bg="blue",fg="white")

btnAddData.grid(row=0,column=0)

btnShowData=Button(FrameButton,command=self.showData,text="Show Data",font=("times new roman",12,"bold"),width=23,bg="blue",fg="white")

btnShowData.grid(row=0,column=1)

btnUpdateData=Button(FrameButton,command=self.updateData,text="Update",font=("times new roman",12,"bold"),width=23,bg="blue",fg="white")

btnUpdateData.grid(row=0,column=2)

btnDeleteData=Button(FrameButton,command=self.deleteData,text="Delete",font=("times new roman",12,"bold"),width=24,bg="blue",fg="white")

btnDeleteData.grid(row=0,column=3)

btnResetData=Button(FrameButton,command=self.resetData,text="Reset",font=("times new roman",12,"bold"),width=23,bg="blue",fg="white")

btnResetData.grid(row=0,column=4)

btnExit=Button(FrameButton,command=self.iExit,text="Exit",font=("times new roman",12,"bold"),width=24,bg="blue",fg="white")

btnExit.grid(row=0,column=5)

#==========================INFORMATION FRAME=================================

FrameDetails=Frame(self.root,bd=10,relief=RIDGE,padx=20,bg="sky blue")

FrameDetails.place(x=0,y=580,width=1350,height=165)

Table\_frame=Frame(FrameDetails,bd=5,relief=RIDGE,bg="sky blue")

Table\_frame.place(x=0,y=2,width=1290,height=100)

xscroll=ttk.Scrollbar(Table\_frame,orient=HORIZONTAL)

yscroll=ttk.Scrollbar(Table\_frame,orient=VERTICAL)

self.library\_table=ttk.Treeview(Table\_frame,column=("membertype","prnno","idno","firstname","lastname","address1",

"address2","postcode","mobile","bookid","booktitle","auther",

"dateborrowed","datedue","days","latereturnfine","dateoverdue","actualprice"),

xscrollcommand=xscroll.set,yscrollcommand=yscroll.set)

xscroll.pack(side=BOTTOM,fill=X)

yscroll.pack(side=RIGHT,fill=Y)

xscroll.config(command=self.library\_table.xview)

yscroll.config(command=self.library\_table.yview)

self.library\_table.heading("membertype",text="Member Type")

self.library\_table.heading("prnno",text="PRN No")

self.library\_table.heading("idno",text="ID No")

self.library\_table.heading("firstname",text="First Name")

self.library\_table.heading("lastname",text="Last Name")

self.library\_table.heading("address1",text="Address1")

self.library\_table.heading("address2",text="Address2")

self.library\_table.heading("postcode",text="Post Code")

self.library\_table.heading("mobile",text="Mobile")

self.library\_table.heading("bookid",text="Book ID")

self.library\_table.heading("booktitle",text="Book Title")

self.library\_table.heading("auther",text="Auther Name")

self.library\_table.heading("dateborrowed",text="Date Borrowed")

self.library\_table.heading("datedue",text="Date Due")

self.library\_table.heading("days",text="Days")

self.library\_table.heading("latereturnfine",text="Late Return Fine")

self.library\_table.heading("dateoverdue",text="Date Over Due")

self.library\_table.heading("actualprice",text="Actual Price")

self.library\_table["show"]="headings"

self.library\_table.pack(fill=BOTH,expand=1)

self.library\_table.column("membertype",width=90)

self.library\_table.column("prnno",width=90)

self.library\_table.column("idno",width=90)

self.library\_table.column("firstname",width=90)

self.library\_table.column("lastname",width=90)

self.library\_table.column("address1",width=90)

self.library\_table.column("address2",width=90)

self.library\_table.column("postcode",width=90)

self.library\_table.column("mobile",width=90)

self.library\_table.column("bookid",width=90)

self.library\_table.column("booktitle",width=90)

self.library\_table.column("auther",width=90)

self.library\_table.column("dateborrowed",width=90)

self.library\_table.column("datedue",width=90)

self.library\_table.column("days",width=90)

self.library\_table.column("latereturnfine",width=90)

self.library\_table.column("dateoverdue",width=90)

self.library\_table.column("actualprice",width=90)

self.fatch\_data()

self.library\_table.bind("<ButtonRelease-1>",self.get\_cursor)

def add\_data(self):

cnx=mysql.connector.connect(user="root",password="parsaiji",host="localhost",database="koustubh")

my\_cursor=cnx.cursor()

my\_cursor.execute("insert into library values(%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s)",(

self.member\_var.get(),

self.prn\_var.get(),

self.id\_var.get(),

self.firstname\_var.get(),

self.lastname\_var.get(),

self.address1\_var.get(),

self.address2\_var.get(),

self.postcode\_var.get(),

self.mobile\_var.get(),

self.bookid\_var.get(),

self.booktitle\_var.get(),

self.auther\_var.get(),

self.dateborrowed\_var.get(),

self.datedue\_var.get(),

self.daysonbook\_var.get(),

self.latereturnfine\_var.get(),

self.dateoverdue\_var.get(),

self.actualprice\_var.get(),

))

#=====================================CLOSE CONNECTION====================================================

cnx.commit()

self.fatch\_data()

cnx.close()

messagebox.showinfo("Success","Member Has Been Inserted Successfully")

def fatch\_data(self):

cnx=mysql.connector.connect(user="root",password="parsaiji",host="localhost",database="koustubh")

my\_cursor=cnx.cursor()

my\_cursor.execute("select \* from library")

rows=my\_cursor.fetchall()

if len(rows)!=0:

self.library\_table.delete(\*self.library\_table.get\_children())

for i in rows:

self.library\_table.insert("",END,values=i)

cnx.commit()

cnx.close()

def get\_cursor(self,event=""):

cursor\_row=self.library\_table.focus()

content=self.library\_table.item(cursor\_row)

row=content['values']

self.member\_var.set(row[0]),

self.prn\_var.set(row[1]),

self.id\_var.set(row[2]),

self.firstname\_var.set(row[3]),

self.lastname\_var.set(row[4]),

self.address1\_var.set(row[5]),

self.address2\_var.set(row[6]),

self.postcode\_var.set(row[7]),

self.mobile\_var.set(row[8]),

self.bookid\_var.set(row[9]),

self.booktitle\_var.set(row[10]),

self.auther\_var.set(row[11]),

self.dateborrowed\_var.set(row[12]),

self.datedue\_var.set(row[13]),

self.daysonbook\_var.set(row[14]),

self.latereturnfine\_var.set(row[15]),

self.dateoverdue\_var.set(row[16]),

self.actualprice\_var.set(row[17])

def showData(self):

self.textBox.insert(END,"Member Type\t\t"+self.member\_var.get() + "\n")

self.textBox.insert(END,"PRN No:\t\t"+self.prn\_var.get() + "\n")

self.textBox.insert(END,"ID No:\t\t"+self.id\_var.get() + "\n")

self.textBox.insert(END,"First Name:\t\t"+self.firstname\_var.get() + "\n")

self.textBox.insert(END,"Last Name\t\t"+self.lastname\_var.get() + "\n")

self.textBox.insert(END,"Address1:\t\t"+self.address1\_var.get() + "\n")

self.textBox.insert(END,"Address2:\t\t"+self.address2\_var.get() + "\n")

self.textBox.insert(END,"Post Code:\t\t"+self.postcode\_var.get() + "\n")

self.textBox.insert(END,"Mobile No:\t\t"+self.mobile\_var.get() + "\n")

self.textBox.insert(END,"Book ID:\t\t"+self.bookid\_var.get() + "\n")

self.textBox.insert(END,"Book Title:\t\t"+self.booktitle\_var.get() + "\n")

self.textBox.insert(END,"Auther:\t\t"+self.auther\_var.get() + "\n")

self.textBox.insert(END,"DateBorrowed:\t\t"+self.dateborrowed\_var.get() + "\n")

self.textBox.insert(END,"DateDue:\t\t"+self.datedue\_var.get() + "\n")

self.textBox.insert(END,"DaysOnBook:\t\t"+self.daysonbook\_var.get() + "\n")

self.textBox.insert(END,"LateReturnFine:\t\t"+self.latereturnfine\_var.get() + "\n")

self.textBox.insert(END,"DateOverDue:\t\t"+self.dateoverdue\_var.get() + "\n")

self.textBox.insert(END,"ActualPrice:\t\t"+self.actualprice\_var.get() + "\n")

def updateData(self):

cnx=mysql.connector.connect(user="root",password="parsaiji",host="localhost",database="koustubh")

my\_cursor=cnx.cursor()

my\_cursor.execute("update library set Member=%s,ID\_NO=%s,FirstName=%s,LastName=%s,Address1=%s,Address2=%s,PostID=%s,Mobile=%s,BookID=%s,BookTitle=%s,Auther=%s,DateBorrowed=%s,DateDue=%s,DaysOnBook=%s,LateReturnFine=%s,DateOverDue=%s,ActualPrice=%s where PRN\_NO=%s",(

self.member\_var.get(),

self.id\_var.get(),

self.firstname\_var.get(),

self.lastname\_var.get(),

self.address1\_var.get(),

self.address2\_var.get(),

self.postcode\_var.get(),

self.mobile\_var.get(),

self.bookid\_var.get(),

self.booktitle\_var.get(),

self.auther\_var.get(),

self.dateborrowed\_var.get(),

self.datedue\_var.get(),

self.daysonbook\_var.get(),

self.latereturnfine\_var.get(),

self.dateoverdue\_var.get(),

self.actualprice\_var.get(),

self.prn\_var.get()

))

cnx.commit()

self.fatch\_data()

self.resetData()

cnx.close()

messagebox.showinfo("success","Member Has Been Updated")

def deleteData(self):

if self.prn\_var.get()=="" or self.id\_var.get()=="":

messagebox.showerror("Error","First Select The Member")

else:

deleteData=tkinter.messagebox.askyesno("Library Management System","Do You Want To Delete?")

if deleteData>0:

cnx=mysql.connector.connect(user="root",password="parsaiji",host="localhost",database="koustubh")

my\_cursor=cnx.cursor()

query="delete from library where PRN\_NO=%s"

value=(self.prn\_var.get(),)

my\_cursor.execute(query,value)

cnx.commit()

self.fatch\_data()

self.resetData()

cnx.close()

messagebox.showinfo("Success","Member Has Been Deleted")

def resetData(self):

self.member\_var.set(""),

self.prn\_var.set(""),

self.id\_var.set(""),

self.firstname\_var.set(""),

self.lastname\_var.set(""),

self.address1\_var.set(""),

self.address2\_var.set(""),

self.postcode\_var.set(""),

self.mobile\_var.set(""),

self.bookid\_var.set(""),

self.booktitle\_var.set(""),

self.auther\_var.set(""),

self.dateborrowed\_var.set(""),

self.datedue\_var.set(""),

self.daysonbook\_var.set(""),

self.latereturnfine\_var.set(""),

self.dateoverdue\_var.set(""),

self.actualprice\_var.set(""),

self.textBox.delete("1.0",END)

def iExit(self):

iExit=tkinter.messagebox.askyesno("Library Management System","Do You Want To Exit?")

if iExit>0:

self.root.destroy()

return

root=Tk()

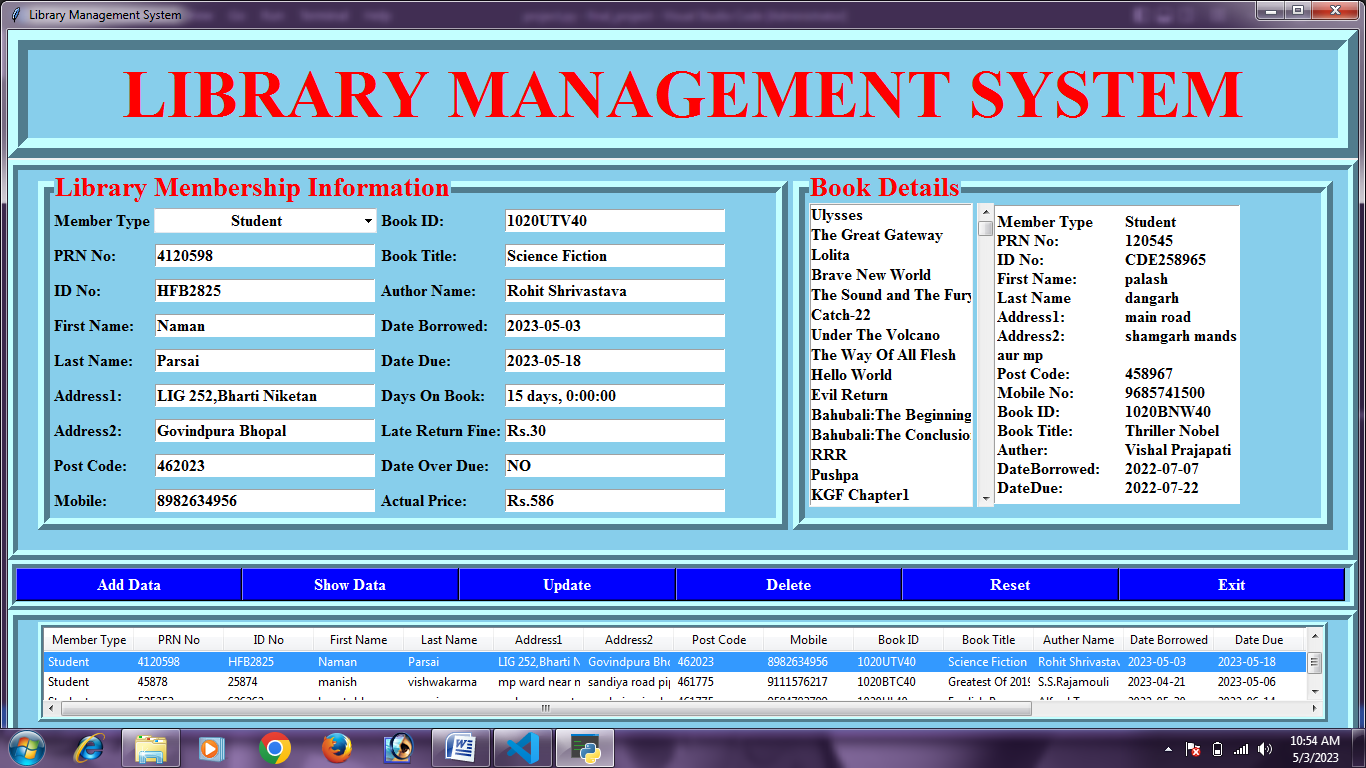
obj=LibraryManagementSystem(root)

root.mainloop()

**The Output Of The Following Source Code Is Shown below :-**



**After Inserting The Data To All The Fields:-**

****

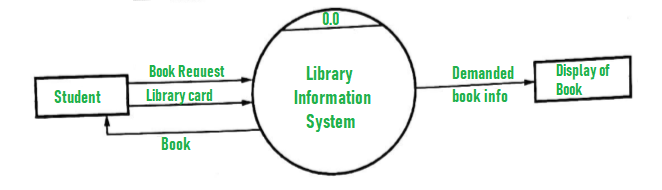
**b). Data Flow Diagram(DFD):-**

Data Flow Diagram(DFD) depicts the flow of information and the transformation applied when a data moves in and out from a system. The overall system is represented and described using input, processing and output in the DFD. The inputs can be:

* **Book request** when a student requests for a book.
* **Library card** when the student has to show or submit his/her identity as a proof.

The overall processing unit will contain the following output that a system will produce or generate:

* Book will be the output as the book demanded by the student will be given to them.
* Information of demanded book should be displayed by the library information system that can be used by the student while selecting the book which makes it easier for the student.
  + - 1. **Level 0 DFD –**



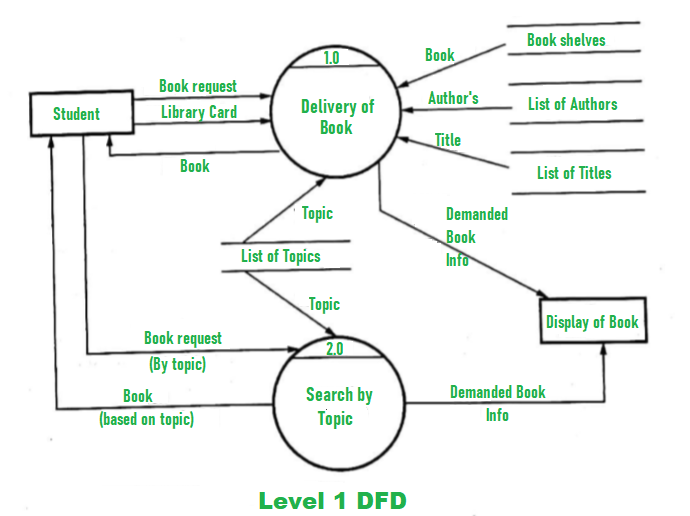
* 1. **Level 1 DFD –**

At this level, the system has to show or exposed with more details of processing.

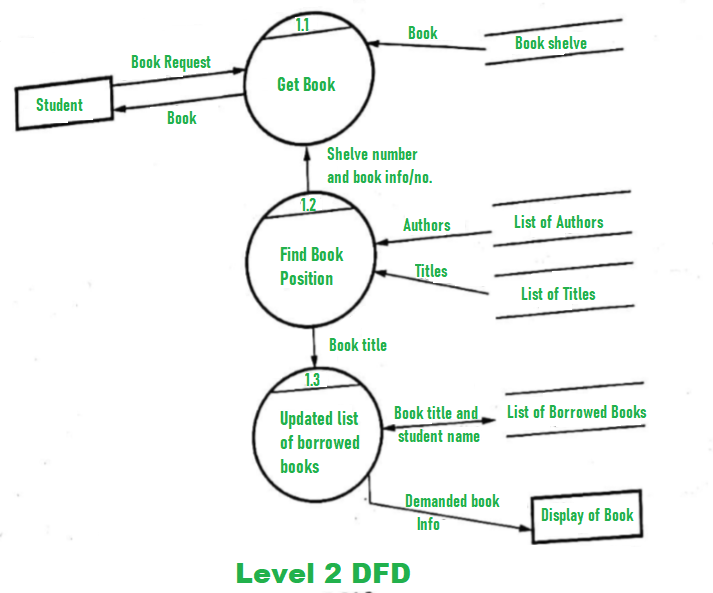
The processes that are important to be carried out are:

* Book delivery
* Search by topic

List of authors, List of Titles, List of Topics, the bookshelves from which books can be located are some information that is required for these processes. **Data store** is used to represent this type of information.



1. **Level 2 DFD –**



**Out of scope:**

Other activities like purchasing of new books, replacement of old books or charging a fine are not considered in the above system.

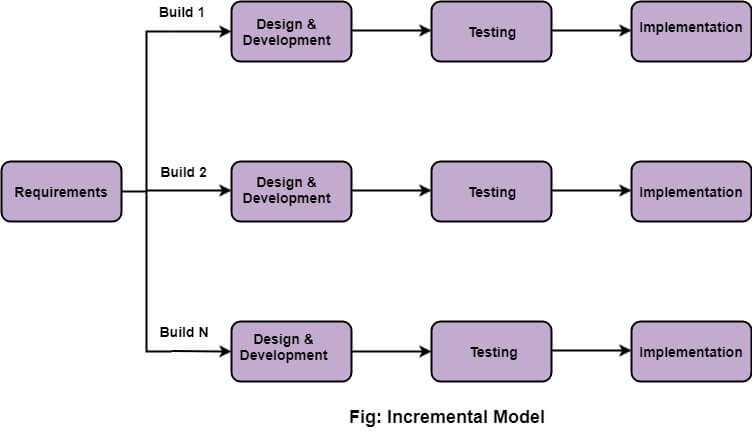
**c). Software Process Model:**

In software engineering, a system development methodology refers to the framework that is used to structure, plan, and control the process of developing an information system. Software development methodologies define the processes we use to build software. These methodologies are also referred to as Software Development Process Models. Each methodology follows a series of steps unique to its type, to ensure success in the process of software development. A software process is a set of related activities that leads to the production of a software product. There are a lot of software processes but they all include four activities:

* + - * Specification
      * Development
      * Validation
      * Evolution←

These are fundamental to software engineering. A wide variety of Software Development methodology has evolved over the years. Each of these methodologies has its own recognized strengths and weaknesses. The following are the most widely used methodologies for software development.

1. **Incremental Method** : The Incremental method of development is based on the idea of developing an initial implementation, exposing this model to user to receive feedback. This model evolves in the form of versions as the requirements change until an acceptable system has been developed.Incremental Model is a process of software development where requirements divided into multiple standalone modules of the software development cycle. In this model, each module goes through the requirements, design, implementation and testing phases. Every subsequent release of the module adds function to the previous release. The process continues until the complete system achieved.



## The various phases of incremental model are as follows:

**1. Requirement analysis:** In the first phase of the incremental model, the product analysis expertise identifies the requirements. And the system functional requirements are understood by the requirement analysis team. To develop the software under the incremental model, this phase performs a crucial role.

**2. Design & Development:** In this phase of the Incremental model of SDLC, the design of the system functionality and the development method are finished with success. When software develops new practicality, the incremental model uses style and development phase.

**3. Testing:** In the incremental model, the testing phase checks the performance of each existing function as well as additional functionality. In the testing phase, the various methods are used to test the behavior of each task

1. **Implementation:** Implementation phase enables the coding phase of the development system. It involves the final coding that design in the designing and development phase and tests the functionality in the testing phase. After completion of this phase, the number of the product working is enhanced and upgraded up to the final system product

## When we use the Incremental Model?

* When the requirements are superior.
* A project has a lengthy development schedule.
* When Software team are not very well skilled or trained.
* When the customer demands a quick release of the product.
* You can develop prioritized requirements first.

## Advantage of Incremental Model

* Errors are easy to be recognized.
* Easier to test and debug
* More flexible.
* Simple to manage risk because it handled during its iteration.
* The Client gets important functionality early.

## Disadvantage of Incremental Model

* Need for good planning
* Total Cost is high.
* Well defined module interfaces are needed.

**d) Functional Diagram:-**

The structural relationship of the system and elements. component diagrams generally simplify the interactions within more complex systems.

In this functionality diagram shows that there are three types of users they – 1). Admin

2). Staff

3). Student

Each have different type of access to this project. Main work among them is of admin every task (maintenance) of the application.

Admin :- They have all the access to this project. Some of them are-

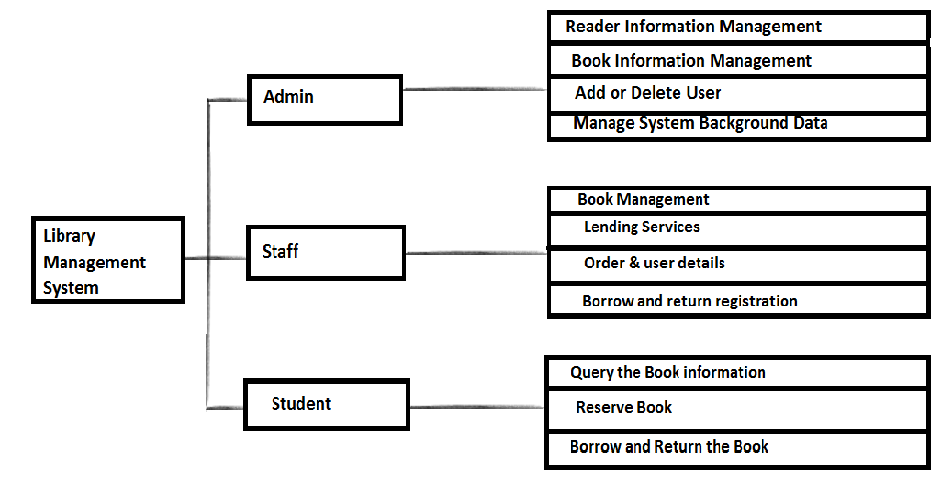
1. Reader Information Management
2. Book Information Management
3. Add or Delete User
4. Manage System Background Data
5. Database Management

Staff :- They have access to student detail verification. Some of them are-

1. Book Management
2. Lending Services
3. Order and User details
4. Borrow and return registration
5. Verify the details of student
6. Give permission to issue the book

Student :- They have only access of reading the book and some others are-

1. Query the book Information
2. Reserve Book
3. Borrow and return the book
4. Registration for Library Management System



**Conclusion**

This application provides a computerized version of library management system which will benefit the students as well as the staff of the library. It makes entire process online where student can search books, staff can generate reports and do book transactions. It also has a facility for student login where student can login and can see status of books issued as well request for book or give some suggestions. It has a facility of teacher’s login where teachers can add lectures notes and also give necessary suggestion to library and also add info about workshops or events happening in our college or nearby college in the online notice board.

There is a future scope of this facility that many more features such as online lectures video tutorials can be added by teachers as well as online assignments submission facility , a feature Of group chat where students can discuss various issues of engineering can be added to this project thus making it more interactive more user friendly and project which fulfills each users need in the best way possible.

The purpose of the library management system is to automate and digitize this traditional way of managing the library work. The Library Management System is much more user-friendly, faster in operation and easy to manage than the manual one. Through the use of it, the librarian can manage the whole data of the library in a single database in different tables with a much more security than the traditional way. In a library, tasks like issue/return/add new students/add new books/ checking any discrepancy in stock, calculating fine for overdue books etc. are performed on a daily basis and suppose a student asks for a particular book from a librarian then he has to search the book manually which takes a lot of time and there are chances of human error in that process as well. But with the help of LMS, the searching of books has become quite easy and fast and the librarian only needs to enter the book id in the search section of the application. Similarly, if a new student has to be added as a registered member of the library or an existing student has to be removed then this process has become very easy to perform. So, all these kinds of operations can be digitized and performed efficiently by using a library management system.

An online library management system is a practical solution for the existing issues of the traditional library system. It is basically a windows application that is built mainly on Java technology and relational database (SQL). The similar application can also be built using the web technologies like Python , Tkinter and a corresponding database , that can be a relational database like sql.

Using this application software, the librarian can search any book by using the issued book id of that particular book in just a second. He can also add new students, new books to the library database, can issue books, return books by making the necessary changes to the database part from the application user-interface. The whole application is divided into different section depending upon its usage. The different sections are explained in the architecture part of this paper.

**BIBLIOGRAPHY**

1. Han J. and Kamber M. (2003): “***Data Mining, Concepts and Techniques***”, Academic Press, 2003.
2. Hand D., Mannila H. and Smyth P. (2001): “***Principle of Data Mining***”. MIT Press, Cambridge, Massachusetts, USA, 2001.
3. J. Han and M. Kamber (2011): “***Data Mining: Concepts and Techniques***”, Morgan Kaufmann Pub., 3rd ed., 2011.
4. Berson (1997): “***Data warehousing, Data Mining & DLAP, Computing***” Mcgraw-Hill, 1997.
5. W. H. Inmon (2005): “***Building the Data Warehouse***”, 3ed, Wiley India, 2005.
6. B. W. Kernighan and R. Pike (1995): “***The UNIX Programming Environment***”, Prentice Hall of India, 1995.
7. Buyya and Selvi (2013): ”***Mastering Cloud Computing*** “,TMH Pub., 2013.
8. Law and Kelton (2000): “***Simulation Modelling and Analysis***” 3rd Ed., McGraw Hill, 2000.
9. Jeffrey R. Shapiro (2002): “***The Complete Reference Visual Basic .NET***”, Tata Mcgraw Hill, 2002.
10. Chris Ullman and Kauffman (2003): *“****Beg ASP.NET1.1 with VC#.NET 2003****”,* Wiley Dream tech., 2003.
11. Han J., Pei J. and Yin Y. (2000): “***Mining Frequent Patterns without Candidate Generation***”. In proceedings of International Conference on Management of Data (ACM SIGMOD’00), pages 1-12, ACM Press Dallas, TX, United States, May 2000.
12. Andrew S. Tanenbaum and Maarten Van Steen (2002): “***Distributed Systems Principles and Paradigms***” Pearson Education Inc. 2002.
13. Eric Maiwald (2004): “***Fundamentals of network security***”, New Delhi, Wiley Dreamtech India Ltd, 2004 Edition.
14. Elaine Rich, Kevin Knight and Shivashankar B. Nair(2009): “***Artifical Intelligence***”, Tata McGraw-HillEducation Private Limited, 2009.
15. G. Somasundaram and Alok Shrivastava (EMC Education Services)(2002): “***Information Storage and Management: Storing, Managing, and Protecting Digital Information***”,2nd Ed., Wiley India, 2002.
16. S. Harrington (1987): “***Computer Graphics - a Programming approach***”, 2nd ed. McGrawhill, New York, 1987.
17. Ullman (2008): "***Analysis and Design of Algorithm***", McGrawhill, New York, 2008.
18. Goodman (2002): “***Introduction to the Design & Analysis of Algorithms”***, TMH-2002.
19. Naughton and Schildt (1999): “***The Complete Reference Java 2***”, Tata McGraw Hill 1999.
20. Deitel (2004): "**Java How to Program**", Fifth Edition. PearsonEducation, 2004.
21. Alfred V. Aho, Ravi Sethi and J.D. Ullman (1986): “***Compilers- Principles, Techniques and tools***”, Addison Wesley, 1986.
22. Elmarsi (2004): “***Fundamentals of Database Systems***”, 6 th Edition, Pearson Education, 2004.
23. R. Ramakrishnan (1998): “***Database Management Systems***”, McGraw Hill, International Editions, 1998.
24. Gillet B.E. (1979): “***Introduction to Operation Research, Computer Oriented Algorithmic approach”,*** Tata McGraw, Hill Publising Co. Ltd., New Delhi, 1979.
25. R. S. Pressman (2002): “***Software Engineering – A practitioner’s approach***”, 6th ed., McGraw Hill Int. Ed., 2002.
26. Lafore R. (2000): “***Object Oriented Programming in C++***”, 3rd. ed., Galgotia Pub, 2000.
27. Balagurusawmy (2008): “***Object Orienter Programming with C++”,*** Tata McGraw-Hill Education, 2008.
28. Hopcroft & Ullman (1979): “***Introduction to Automata theory, languages & Computation***”, Narosha Publishing house, 1979.
29. A. S. Tanenbaum (1989): “***Computer Network***”, 4th addition, PHI, 1989.
30. A. S. Tanenbaum (1995): "***Modern Operating System***", Prentice Hall of India Pvt. Ltd., 1995.
31. H. M. Deitel (1984): "***An Introduction to Operating Systems***". Addison Wesley Publishing Company, 1984.
32. A Silberschatz, H.F Korth and Sudersan (2002): “***Database System Concepts***”, 4th ed. MGH Publication, 2002.
33. C. J. Date (1997): “***An introduction to Database Systems***”, Addison-Wesley, 6th edition, 1997.
34. Iyengyr M. K. Jain & R. K. Jain (1995): “***Numerical Methods for scientific and engineering computation***”, Wiley Eastern (New Age), 1995.
35. Hipp J., Guntzer U. and Nakhaeizadeh G. (2000): “***Algorithms for Association Rule Mining: A General Survey and Comparison***”. SIGKDD Explorations, Vol. 2, No. 1, pages 58-64, July 2000.
36. Bhattacharya S.K. and Deardan John (1986 ): “***Accounting for Management***” PHI, South Asia Books, Jan 1, 1986.
37. Rajaraman V. (2004): “***Fundamental of Computers***”, 4nd edition, Prentice Hall of India, New Delhi, 2004.
38. J. P. Trembley & R. P. Manohar (1989): “***Discrete Mathematical Structure with applications to Computer Science***”. McGraw Hill, 1989;
39. Brian W. Kernighan and Dennis M. Ritchie(1990): “***The C programming language***”, PHI, 284 pages; **PHI**; 2 edition, 1990.
40. Schildt (2000 ): “***C The Complete reference***”, McGraw-Hill Osborne Media; 4 edition, April 26, 2000.
41. M. Morris Mano (1993 ): “***Computer System Architecture***”, PHI, 3rd edition, 1993.
42. Website URL: https://www.w3schools.com/java/