**Book Store Database Application**

***A project Report***

***submitted in partial fulfillment of the***

***requirements for the award of the degree of***

**Bachelor of Technology (B. Tech)**

***Submitted by***

***ANURAG DEV***

***(60314803116)***

******

***Under the supervision of***

Supervisors name and affiliation

**DEPARTMENT OF INFORMATION TECHNOLOGY**

**MAHARAJA AGRASEN INSTITUTE OF TECHNOLOGY**

**SEC-22, ROHINI, DELHI - 110086, INDIA**

**SEPTEMBER, 2018**

**ACKNOWLEDGEMENTS**

I have taken keen efforts in this project. However, it would have not been possible without the kind support and help of many individuals and organizations. I would like to extend my sincere thanks to all of them.

I am highly indebted to **MR. DIVANSHU** for their guidance and constant supervision as well as providing quintessential information regarding the project and also for their support in completing the project.

I would like to express my gratitude towards my parents and members of Kyrion Technologies Ltd.for their co-operation and enlivenment which helped me in completion of this project.

I would like to express my special gratitude and thanks to our institution Maharaja Agrasen Institute of Technology (MAIT) for giving me the opportunity to work on this project.

My appreciations also go to sites like Youtube and Google that provided me with plenty study materials and information for implementation of idea of this project.

**Anurag Dev**

**CANDIDATE'S DECLARATION**

I hereby declare that the work presented in this project report titled, “**BOOKSTORE DATABASE APPLICATION**" submitted by me in the partial fulfillment of the requirement of the award of the degree of **Bachelor of Technology (B.Tech)** Submitted in the Department of **Information Technology,** Maharaja Agrasen Institute of Technology is an authentic record of my project work carried during 2 months of Summer Training Period.

Date: 12 September, 2018 Anurag Dev

Place: Delhi Roll Number: 60314803116

**ABOUT THE ORGANIZATION**

**Kyrion Digital Securities**, Flagship Company of Kyrion Group is working towards the vision to develop India as a **Cyber Secured Country**. It is a leading player in India to provide Ethical Hacking & IT Security Training to all possible audience. With an outreach to over eighty thousand students and over a thousand major colleges, Kyrion Digital Securities is a well known and trusted brand in the education and information security business.

Kyrion Digital Securities has a wide and vibrant profile which caters to a range of professional courses that are customized to suit the aspirations of different graduate streams. The USP of the company lies in its ability to clean and sustain itself, it is supported by a clear and strong business model that focuses of improvising to adjust with the trends and needs of a volatile market.

**A brief tabular description of Kyrion Digital Securities is stated:**

* Digital Security is the need of an hour. Kyrion provides training and education in the field of Ethical Hacking & Information Security to the students from schools and colleges as well as to corporate executives. Trainings can be availed at client location (Onsite Training), at Kyrion Training Center (Offsite Training) or through distance mode.
* The IT sector is currently facing a shortage in Security Solutions. Kyrion's home-grown concept of IT with Security Plugins has proved to be an exemplary solution. Kyrion aims to create a better workforce which could support and shape the evolving sector of IT security. In this way, Kyrion helps the IT sector by providing efficient and cost-effective manpower for their businesses.
* Engineering & MCA colleges are aplenty, however a flip side of this truth is that there is a dearth of trained and motivated faculty, because of which students are unable to meet the corporate requirements of the market. Kyrion with its initiative Campus2Corporate aims to bridge the gap between theoretical knowledge acquired from university and the ability to tackle the day-to-day problems faced by the industry.

**Head Offices are located at all Major Locations such as:**

* Connaught Place Delhi, India
* Pennsylvania, United State of America (USA)
* London, United Kingdom (UK)
* Perth, Western Australia

Team at Kyrion believes that opportunities are borne out of imperfections. Our **Education Entrepreneurship Firm** aims to deliver high-end security solutions and training to students and corporates to prepare them to meet the challenges of IT Security Industry. Kyrion also set the benchmark in the industry by being the premier entity that introduced the concept of IT with Security Plugins in India and since then, the Indian IT sector has soared higher.

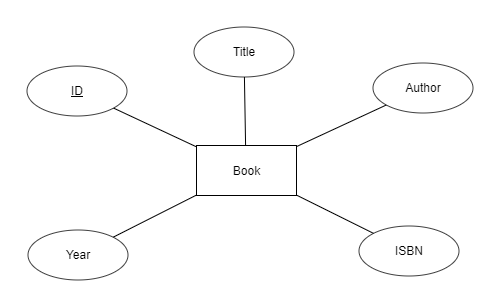
One of our major challenges was to sensitize the professionals and students about the need to master IT Security skill set. Kyrion aimed at building a sustainable culture of trained security professionals that would help the industries reduce their dependency upon foreign experts and solutions. An extensive network of Indian Defense (Navy and Air Force), 1000 engineering colleges, 102 schools and 62 corporate, government & defense organizations in this short span of time makes us feel proud of our endeavors and motivate us to continue on the chosen path. We are one of the fastest growing and globally recognized IT Securities firm for conducting workshops and corporate training in India and world-wide.

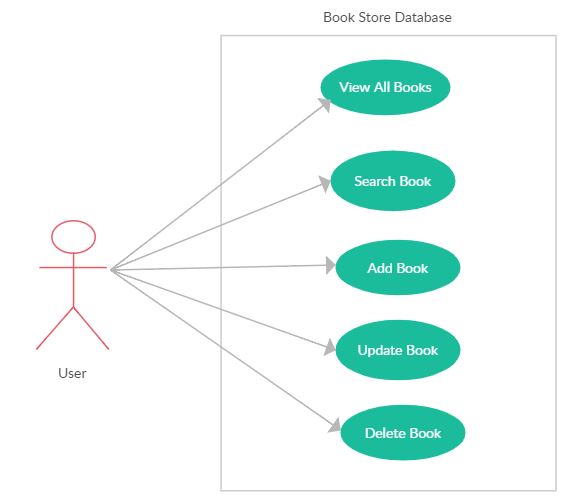
**ABSTRACT**

Small bookstores owners are generally facing problems for keeping the track and record of books that are available in their stores. That’s why I thought of making this database application so that the store owners can keep the record of books and its related information (like ISBN number, Year, etc.) readily available with them.

This project contains both front-end and back-end code for a Bookstore Database Application with a simple graphical user interface (GUI) built with Tkinter library. While I personally might not choose Tkinter if I were to build a real-world app with modern UI, Tkinter serves me well in terms of getting me familiar with how to connect to databases (SQLite in this case), manipulate data, and reflect the changes on the front end with Python.

The extensive use of programming language like Python and knowledge of other import field like database connectivity using SQL, GUI implementation makes it a very interactive and feature rich database connectivity application.

**ER Diagram and Use Case Diagram**

****

**TABLE OF CONTENTS**

**Acknowledgement i**

**Candidate Declaration ii**

**Training Letter iii**

**Certificates iv**

**Abstract v**

**ER Diagram and Use Case Diagram vi**

CHAPTER 1: INTRODUCTION

1.1 Introduction

1.2 Python

1.2.1 Features

1.2.2 Libraries

1.2.3 Uses

1.3 SQL

1.3.1 Uses

1.3.2 Alternatives of SQL

CHAPTER 2: GUI AND WIDGETS

2.1 What is GUI?

2.2 What are Widgets?

2.3 GUI Implementation in Python

2.3.1 Using Tkinter

2.3.2 Standard Attributes

2.3.3 Methods of Geometry Management

CHAPTER 3: DBMS IN PYTHON

3.1 What is Database and DBMS?

3.2 SQL Using Python

3.3 SQL Statements

3.3.1 DML

3.3.2 DDL

3.3.3 DCL

3.3.4 TCL

3.4 SQLite3 in Python

CHAPTER 4: ABOUT THE PROJECT

4.1 Parts of Project

4.1.1 Front End

4.1.2 Back End

4.2 Functionalities of Project

4.3 System Requirements

**CHAPTER 1**

**INTRODUCTION**

**1.1 INTRODUCTION**

The project introduces the problem faced by small bookstore owners for keeping the track and record of books available in their stores so that they can provide customer with the information about books that are sold by them and that are available in their stores.

With the digitization of world the computers are made available to the reach of common people at an affordable price. Hence, computers can be used as a tool for keeping the record of books available on these bookstores due to their high speed, reliability, accuracy and efficient memory.

The project contains code for both front end and back end part of the application.

Front end part of the application contains GUI implementation to make it more interactive and user friendly through the use of various widgets like textboxes, buttons, scrollbars, list boxes, etc. These functionalities make it easier to use this application even by some naïve user.

Back end part is the heart of this application and contains code for creating the database and connecting to the database for querying it. Database contains the information like title, Author, etc. about any book record in the relational (or Tabular) form. Basically back end part contains the action to be performed when any button on this application is clicked by the user. When a user is clicking on a particular button of this application then he/she is querying the database basically.

This project is developed with the help of two computer languages:

* Python
* SQL

Let’s know in brief about each of them.

**1.2 PYTHON**

**Python** is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace. It provides constructs that enable clear programming on both small and large scales. In July 2018, Van Rossum stepped down as the leader in the language community after 30 years.

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

Python interpreters are available for many operating systems. CPython, the reference implementation of Python, is open source software and has a community-based development model, as do nearly all of Python's other implementations. Python and CPython are managed by the non-profit Python Software Foundation.

**1.2.1 FEATURES OF PYTHON**

Python is a multi-paradigm programming language. Object-oriented programming and structured programming are fully supported, and many of its features support functional programming and aspect-oriented programming (including by metaprogramming and metaobjects (magic methods)). Many other paradigms are supported via extensions, including design by contract and logic programming.

Python uses dynamic typing, and a combination of reference counting and a cycle-detecting garbage collector for memory management. It also features dynamic name resolution ([late binding](https://en.wikipedia.org/wiki/Late_binding)), which binds method and variable names during program execution.

Python's design offers some support for functional programming in the Lisp tradition. It has filter( ), map( ), and reduce( ) functions; list comprehensions, dictionaries, and sets; and generator expressions. The standard library has two modules (itertools and functools) that implement functional tools borrowed from Haskell and Standard ML.

Rather than having all of its functionality built into its core, Python was designed to be highly extensible. This compact modularity has made it particularly popular as a means of adding programmable interfaces to existing applications. Van Rossum's vision of a small core language with a large standard library and easily extensible interpreter stemmed from his frustrations with ABC, which espoused the opposite approach.

**1.2.2 LIBRARIES OF PYTHON**

Python's large standard library, commonly cited as one of its greatest strengths, provides tools suited to many tasks. For Internet-facing applications, many standard formats and protocols such as MIME and HTTP are supported. It includes modules for creating graphical user interfaces, connecting to relational databases, generating pseudorandom numbers, arithmetic with arbitrary precision decimals, manipulating [regular expressions](https://en.wikipedia.org/wiki/Regular_expression), and [unit testing](https://en.wikipedia.org/wiki/Unit_testing).

Some parts of the standard library are covered by specifications (for example, the Web Server Gateway Interface (WSGI) implementation wsgiref follows PEP 333), but most modules are not. They are specified by their code, internal documentation, and test suites (if supplied). However, because most of the standard library is cross-platform Python code, only a few modules need altering or rewriting for variant implementations.

As of March 2018, the Python Package Index (PyPI), the official repository for third-party Python software, contains over 130,000packages with a wide range of functionality, including:

* Graphical user interfaces
* Web frameworks
* Multimedia
* Databases
* Networking
* Test frameworks
* Automation
* Web scraping
* Documentation
* System administration
* Scientific computing
* Text processing
* Image processing

**1.2.3 USES OF PYTHON**

Since 2003, Python has consistently ranked in the top ten most popular programming languages in the TIOBE Programming Community Index where, as of January 2018, it is the fourth most popular language (behind Java, C, and C++).It was selected Programming Language of the Year in 2007 and 2010.

An empirical study found that scripting languages, such as Python, are more productive than conventional languages, such as C and Java, for programming problems involving string manipulation and search in a dictionary, and determined that memory consumption was often "better than Java and not much worse than C or C++".

Large organizations that use Python include Wikipedia, Google, Yahoo!, CERN, NASA, Facebook, Amazon,  Instagram, Spotify and some smaller entities like ILM and ITA. The social news networking site Reddit is written entirely in Python.

Python can serve as a scripting language for web applications, e.g., via mod\_wsgi for the Apache web server. With Web Server Gateway Interface, a standard API has evolved to facilitate these applications. Web frameworks like Django, Pylons, Pyramid, TurboGears, web2py, Tornado, Flask, Bottle and Zope support developers in the design and maintenance of complex applications. Pyjs and IronPython can be used to develop the client-side of Ajax-based applications. SQLAlchemy can be used as data mapper to a relational database. Twisted is a framework to program communications between computers, and is used (for example) by Dropbox.

Libraries such as NumPy, SciPy and Matplotlib allow the effective use of Python in scientific computing, with specialized libraries such as Biopython and Astropy providing domain-specific functionality. SageMath is mathematical software with a "notebook" programmable in Python: its library covers many aspects of mathematics, including algebra, combinatorics, numerical mathematics, number theory, and calculus.

Python has been successfully embedded in many software products as a scripting language, including in finite element method software such as Abaqus, 3D parametric modeler like FreeCAD, 3D animation packages such as 3ds Max, Blender, Cinema 4D, Lightwave, Houdini, Maya, modo, MotionBuilder, Softimage, the visual effects compositor Nuke, 2D imaging programs like GIMP, Inkscape, Scribus and Paint Shop Pro, and musical notation programs like scorewriter and capella. GNU Debugger uses Python as a pretty printer to show complex structures such as C++ containers. Esri promotes Python as the best choice for writing scripts in ArcGIS. It has also been used in several video games,and has been adopted as first of the three available programming languages in Google App Engine, the other two being Java and Go. Python is also used in algorithmic trading and quantitative finance. Python can also be implemented in APIs of online brokerages that run on other languages by using wrappers.

Python has been used in artificial intelligence projects.As a scripting language with modular architecture, simple syntax and rich text processing tools, Python is often used for natural language processing.

Many operating systems include Python as a standard component. It ships with most Linux distributions, AmigaOS 4, FreeBSD, NetBSD, OpenBSD and macOS, and can be used from the command line (terminal). Many Linux distributions use installers written in Python: Ubuntu uses the Ubiquity installer, while Red Hat Linux and Fedora use the Anaconda installer. Gentoo Linux uses Python in its package management system, Portage.

Python is used extensively in the information security industry, including in exploit development.

Most of the Sugar software for the One Laptop per Child XO, now developed at Sugar Labs, is written in Python.

The Raspberry Pi single-board computer project has adopted Python as its main user-programming language.

LibreOffice includes Python, and intends to replace Java with Python. Its Python Scripting Provider is a core feature since Version 4.0 from 7 February 2013.

**1.3 SQL**

**SQL** (**Structured Query Language**) is a domain-specific language used in programming and designed for managing data held in a relational database management system (RDBMS), or for stream processing in a relational data stream management system (RDSMS). It is particularly useful in handling structured data where there are relations between different entities/variables of the data. SQL offers two main advantages over older read/write APIs like ISAM or VSAM: first, it introduced the concept of accessing many records with one single command; and second, it eliminates the need to specify *how* to reach a record, e.g. with or without an index.

Originally based upon relational algebra and tuple relational calculus, SQL consists of many types of statements, which may be informally classed as sublanguages, commonly: a data query language (DQL), a data definition language (DDL), a data control language (DCL), and a data manipulation language (DML). The scope of SQL includes data query, data manipulation (insert, update and delete), data definition (schema creation and modification), and data access control. Although SQL is often described as, and to a great extent is, a declarative language (4GL), it also includes procedural elements.

**1.3.1 USES OF SQL**

Essentially, Structured Query Language (SQL) is used to retrieve data or otherwise interface with a relational database. As a standard going back to the 1970s, SQL is a popular way to get information out of relational database systems. Relational databases are set up with a particular structure – each record has a series of keys that are linked to one another in consistent ways, and placed in a "table" represented visually in a grid.

The SQL language is written to comb the contents of tables in a conventional database. SQL is widely used in business and in other types of database administration. It is the default tool for “operating” on the conventional database, to alter tabled data, retrieve data or otherwise manipulate an existing data set.

Simple SQL commands like SELECT, ORDER BY and INSERT (all of which are typically rendered in all capital letters) help administrators to route data in and out of a database table. This goes on over all sorts of platforms, and is a major part of delivering data results in today’s cloud and hybrid distributed systems. In the API economy, where so many pieces of “middleware” or connecting pieces join parts of IT architecture, having SQL as a consistent database language has been central to porting data to all of those places that it needs to go. Because of the fairly straightforward syntax and ease of use, administrators can then focus on the theory of database construction and the logistical aspect of getting data into and out of systems.

Over time, an alternative to SQL has emerged called NoSQL. The concept is that data that is not tabled in a relational database may not need SQL as a query language. So the biggest uses of SQL are in a spectrum that might be called “smaller” database systems. Another way to explain this is that SQL does not “scale infinitely.” So by that principle, the SQL is used for traditional DB systems and other methods are used for larger NoSQL database systems where checks on data are not strictly enforced

**1.3.2 ALTERNATIVES OF SQL**

A distinction should be made between alternatives to SQL as a language, and alternatives to the relational model itself. Below are proposed relational alternatives to the SQL language. See navigational database and NoSQL for alternatives to the relational model.

* **.QL**: object-oriented Datalog
* **4D Query Language (4D QL)**
* [**BQL**](http://tech.pro/blog/1917/a-better-query-language-bql-language-specification): a superset that compiles down to SQL
* **Datalog**: critics suggest that [Datalog](https://en.wikipedia.org/wiki/Datalog) has two advantages over SQL: it has cleaner semantics, which facilitates program understanding and maintenance, and it is more expressive, in particular for recursive queries.
* **HTSQL**: URL based query method
* **IBM Business System 12 (IBM BS12)**: one of the first fully relational database management systems, introduced in 1982
* **jOOQ**: SQL implemented in Java as an internal domain-specific language
* **Java Persistence Query Language (JPQL)**: The query language used by the Java Persistence API and Hibernate persistence library
* **JavaScript**: MongoDB implements its query language is a JavaScript API.
* **LINQ**: Runs SQL statements written like language constructs to query collections directly from inside .Net code.
* **QBE (Query By Example)** created by Moshè Zloof, IBM 1977
* **Quel** introduced in 1974 by the U.C. Berkeley Ingres project.
* **Tutorial D**
* **XQuery**

**CHAPTER 2**

**GUI AND WIDGETS**

**2.1 WHAT IS GUI?**

The **graphical user interface** (**GUI**), is a type of user interface that allows users to interact with electronic devices through graphical icons and visual indicators such as secondary notation, instead of text-based user interfaces, typed command labels or text navigation. GUIs were introduced in reaction to the perceived steep learning curve of command-line interfaces (CLIs), which require commands to be typed on a computer keyboard.

The actions in a GUI are usually performed through direct manipulation of the graphical elements. Beyond computers, GUIs are used in many handheld mobile devices such as MP3 players, portable media players, gaming devices, smartphones and smaller household, office and industrial controls. The term *GUI* tends not to be applied to other lower-display resolution types of interfaces, such as video games (where *head-up display* (HUD) is preferred), or not including flat screens, like volumetric displays because the term is restricted to the scope of two-dimensional display screens able to describe generic information, in the tradition of the computer science research at the Xerox Palo Alto Research Center.

The visible graphical interface features of an application are sometimes referred to as *chrome* or *GUI* (pronounced *gooey*). Typically, users interact with information by manipulating visual widgets that allow for interactions appropriate to the kind of data they hold. The widgets of a well-designed interface are selected to support the actions necessary to achieve the goals of users. A model–view–controller allows a flexible structure in which the interface is independent from and indirectly linked to application functions, so the GUI can be customized easily. This allows users to select or design a different *skin* at will, and eases the designer's work to change the interface as user needs evolve. Good user interface design relates to users more, and to system architecture less.

**2.2 WHAT ARE WIDGETS?**

**A control element** (sometimes called a **control** or **widget**) in a graphical user interface is an element of interaction, such as a button or a scroll bar. Controls are software components that a computer user interacts with through direct manipulation to read or edit information about an application. User interface libraries such as Windows Presentation Foundation, GTK+, and Cocoa, contain a collection of controls and the logic to render these.

Each widget facilitates a specific type of user-computer interaction, and appears as a visible part of the application's GUI as defined by the theme and rendered by the rendering engine. The theme makes all widgets adhere to a unified aesthetic design and creates a sense of overall cohesion. Some widgets support interaction with the user, for example labels, buttons, and check boxes. Others act as containers that group the widgets added to them, for example windows, panels, and tabs.

Structuring a user interface with widget toolkits allows developers to reuse code for similar tasks, and provides users with a common language for interaction, maintaining consistency throughout the whole information system.

Graphical user interface builders facilitate the authoring of GUIs in a WYSIWYG manner employing a user interface markup language. They automatically generate all the source code for a widget from general descriptions provided by the developer, usually through direct manipulation.

**2.3 GUI IMPLEMENTATION IN PYTHON**

Python provides several different options for writing GUI based programs. These are listed below:

* **Tkinter**: It is the easiest among all to get started with. It is Python's standard GUI (Graphical User Interface) package. It is the most commonly used toolkit for GUI Programming in Python.
* **JPython**: It is the Python platform for Java that is providing Python scripts seamless access o Java class Libraries for the local machine.
* **wxPython**: It is open-source, cross-platform GUI toolkit written in C++. It one of the alternatives to Tkinter, which is bundled with Python.

There are many other interfaces available for GUI. But these are the most commonly used ones. We will be using Tkinter in this project so let’s learn about it in brief.

## 2.3.1 USING Tkinter

It is the standard GUI toolkit for Python. Fredrik Lundh wrote it. For modern Tk binding, Tkinter is implemented as a Python wrapper for the Tcl Interpreter embedded within the interpreter of Python. Tk provides the following widgets:

* button
* canvas
* combo-box
* frame
* level
* check-button
* entry
* level-frame
* menu
* list - box
* menu button
* message
* progress-bar
* radio button
* scroll bar
* separator
* tree-view and many more.

Creating a GUI program using this Tkinter is simple. For this programmers need to follow the steps mentioned below:

1. Import the module Tkinter
2. Build a GUI application (as a window)
3. Add those widgets that are mentioned above
4. Enter the primary, i.e., main event's loop for taking action when the user triggered the event.

**2.3.2 STANDARD ATTRIBUTES FOR GUI**

* Dimensions
* Fonts
* Colors
* Cursors
* Anchors
* Bitmaps

**2.3.3 METHODS FOR GEOMETRY MANAGEMENT**

* **pack()**: This method manages the geometry of widgets in blocks
* **grid()**: This method organizes widgets in a tabular structure
* **place()**: This method organizes the widgets to place them in a specific position

**CHAPTER 3**

**DBMS IN PYTHON**

**3.1 WHAT IS A DATABASE AND DBMS?**

**A database** is an organized collection of data, stored and accessed electronically. Database designers typically organize the data to model aspects of reality in a way that supports processes requiring information, such as (for example) modeling the availability of rooms in hotels in a way that supports finding a hotel with vacancies.

**The database management system (DBMS)** is the software that interacts with end users, applications, and the database itself to capture and analyze data. A general-purpose DBMS allows the definition, creation, querying, update, and administration of databases. A database is generally stored in a DBMS-specific format which is not portable, but different DBMSs can share data by using standards such as SQL and ODBC or JDBC. The sum total of the database, the DBMS and its associated applications can be referred to as a "database system". Often the term "database" is used to loosely refer to any of the DBMS, the database system or an application associated the database.

Computer scientists may classify database-management systems according to the database models that they support. Relational databases became dominant in the 1980s. These model data as rows and columns in a series of tables, and the vast majority use SQL for writing and querying data. In the 2000s, non-relational databases became popular, referred to as NoSQL because they use different query languages.

Formally, a "database" refers to a set of related data and the way it is organized. Access to this data is usually provided by a "database management system" (DBMS) consisting of an integrated set of computer software that allows users to interact with one or more databases and provides access to all of the data contained in the database (although restrictions may exist that limit access to particular data). The DBMS provides various functions that allow entry, storage and retrieval of large quantities of information and provides ways to manage how that information is organized.

Because of the close relationship between them, the term "database" is often used casually to refer to both a database and the DBMS used to manipulate it.

Outside the world of professional information technology, the term *database* is often used to refer to any collection of related data (such as a spreadsheet or a card index) as however size and usage requirements typically necessitate use of a database management system.

Existing DBMSs provide various functions that allow management of a database and its data which can be classified into four main functional groups:

* **Data definition** – Creation, modification and removal of definitions that define the organization of the data.
* **Update** – Insertion, modification, and deletion of the actual data.
* **Retrieval** – Providing information in a form directly usable or for further processing by other applications. The retrieved data may be made available in a form basically the same as it is stored in the database or in a new form obtained by altering or combining existing data from the database.
* **Administration** – Registering and monitoring users, enforcing data security, monitoring performance, maintaining data integrity, dealing with concurrency control, and recovering information that has been corrupted by some event such as an unexpected system failure.

Both a database and its DBMS conform to the principles of a particular database model.[]](https://en.wikipedia.org/wiki/Database#cite_note-FOOTNOTETsitchizrisLochovsky1982-5) "Database system" refers collectively to the database model, database management system, and database.[]](https://en.wikipedia.org/wiki/Database#cite_note-FOOTNOTEBeynon%E2%80%93Davies2003-6)

Physically, database servers are dedicated computers that hold the actual databases and run only the DBMS and related software. Database servers are usually multiprocessor computers, with generous memory and RAID disk arrays used for stable storage. RAID is used for recovery of data if any of the disks fail. Hardware database accelerators, connected to one or more servers via a high-speed channel, are also used in large volume transaction processing environments. DBMSs are found at the heart of most database applications. DBMSs may be built around a custom multitasking kernel with built-in networking support, but modern DBMSs typically rely on a standard operating system to provide these functions.

Since DBMSs comprise a significant market, computer and storage vendors often take into account DBMS requirements in their own development plans.

Databases and DBMSs can be categorized according to the database model(s) that they support (such as relational or XML), the type(s) of computer they run on (from a server cluster to a mobile phone), the query language(s) used to access the database (such as SQL or XQuery), and their internal engineering, which affects performance, scalability, resilience, and security.

**3.2 SQL USING PYTHON**

SQL is used to communicate with a database. According to ANSI (American National Standards Institute), it is the standard language for relational database management systems. **SQL statements** are used to perform tasks such as **update** data on a database, or retrieve data from a database.

Connecting a program with a database is considered a tough task in any programming language. It is used to connect the front-end of your application with the back-end database. Python with its native built-in modules made this thing easy too.  
This needs the basic understanding of SQL.

Here, we are going to connect SQLite with Python. Python has a native library for SQLite. Let us explain how it works.

1. To use SQLite, we must import sqlite3.
2. Then create a connection using connect() method and pass the name of the database you want to access if there is a file with that name, it will open that file. Otherwise, Python will create a file with the given name.
3. After this, a cursor object is called to be capable to send commands to the SQL. Cursor is a control structure used to traverse and fetch the records of the database. Cursor has a major role in working with Python. All the commands will be executed using cursor object only.
4. To create a table in the database, create an object and write the SQL command in it with being commented. Example:- sql\_comm = ”SQL statement”
5. And executing the command is very easy. Call the cursor method execute and pass the name of the sql command as a parameter in it. Save a number of commands as the sql\_comm and execute them. After you perform all your activities, save the changes in the file by committing those changes and then lose the connection.

**3.3 SQL STATEMENTS**

SQL statements are categorized into four different type of statements, which are

1. DML (DATA MANIPULATION LANGUAGE)
2. DDL (DATA DEFINITION LANGUAGE)
3. DCL (DATA CONTROL LANGUAGE)
4. TCL (TRANSACTION CONTROL LANGUAGE)

**3.3.1 DML**  
In Data Manipulation Language (DML), we have four different SQL statements.

1. **SELECT**  
     
   Select statement is used to select the collection of records from the table, which is based on some condition.  
     
   E.g. select \* from student – Get all the records of student table.  
     
   Select \* from student where rank>5 – Get the records with the condition where students' rank is greater than 5.
2. **INSERT**Insert statement is used to insert the set of values into the table.  
     
   E.g. Insert into Student (Rank, StudentName, Mark) Values(1,’Kumar’,450)
3. **UPDATE**Update statement is used to update the existing values in the table, which is based on some condition.  
     
   E.g., update student set StudentName=’Manoj’ where StudentName=’Kumar’   
     
   The query given above will update the studentName from Manoj to Kumar where student Name Kumar.
4. **DELETE**Delete statement is used to delete the existing record in the table, which is based on some condition.  
     
   Eg., Delete from Student where StudentName=’Manoj’  
     
   The query given above will delete the record which has StudentName Manoj.

**3.3.2 DDL**  
In Data Definition Language (DDL), we have three different SQL statements.

1. **CREATE**  
     
   Create statement is used to create the new table in an existing database.  
     
   Eg., Create Table Student (Rank Int,StudentName varchar(50),Mark Float)
2. **ALTER**  
     
   Alter statement can add a column, modify a column, drop a column, rename a column or rename a table.  
     
   Eg., Alter Table Student Add (StudentAddress varchar (100))
3. **DROP**  
     
   SQL DROP TABLE statement is used to remove a table definition and all the data, indexes, triggers, constraints and permission specifications for the table.  
     
   Eg, Drop Student

**3.3.3 DCL**  
In Data Control Language (DCL), it defines the control over the data in the database. We have two different commands, which are

1. **GRANT**  
     
   Grant is allowed to do the specified user to the specified tasks.  
     
   **Syntax**  
     
   *GRANT privilege\_name  
   ON object\_name  
   TO {user\_name |PUBLIC |role\_name}   
   [WITH GRANT OPTION];*
2. **REVOKE**  
   It is used to cancel previously granted or denied permissions.  
     
   **Syntax**  
     
   *REVOKE privilege\_name  
   ON object\_name  
   FROM {user\_name |PUBLIC |role\_name}*

**3.3.4 TCL**  
In Transaction Control Language (TCL), the commands are used to manage the transactions in the database. These are used to manage the changes made by DML statements. It also allows the statements to be grouped together into logical transactions.

1. **COMMIT**Commit command is used to permanently save any transaction into the database.  
     
   Syntax Commit;
2. **ROLLBACK**  
     
   Rollback command is used to restore the database for the last committed state. It’s also used with save point to jump to the save point.  
     
   **Syntax**  
     
   Rollback to save point name
3. **SAVEPOINT**  
     
   Save point command is used to temporarily save a transaction, so that you can roll back to that point whenever necessary.  
     
   **Syntax**  
   savepointsavepoint-name;

**3.4 SQLite3 IN PYTHON**

SQLite3 is a very easy to use database engine. It is self-contained, server less, zero-configuration and transactional. It is very fast and lightweight, and the entire database is stored in a single disk file. It is used in a lot of applications as internal data storage. The Python Standard Library includes a module called "sqlite3" intended for working with this database. This module is a SQL interface compliant with the DB-API 2.0 specification.

**CHAPTER 4**

**ABOUT THE PROJECT**

**4.1 PARTS OF PROJECT**

This book store database application has been divided into following 2 parts:

1. Back End
2. Front End

These parts are combined together to from the complete project. For running this application user has to double click on front\_end.py file.

On clicking the front\_end.py file the application a window will open which provide the user with functionalities like:

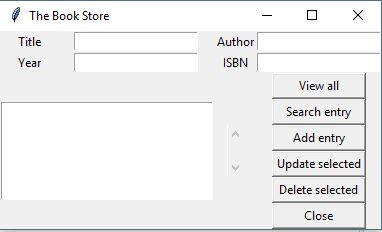
* Adding a new book entry
* Searching a book
* Updating the details of book
* Deleting a book record permanently
* Viewing all books at once
* Closing the window

These functionalities are available to user in form of GUI which will make it simpler to use and also make it appear more interactive and user friendly. This is only possible with the help of Tkinter module of python which provide all the functionalities needed for GUI. So it forms the **front end** of application.

Also due to Sqlite3 module provided by python the implementation of all these functionalities provided in GUI is possible. Thus implementation part of this application is hidden and so it forms the **back end** of the application. Sqlite3 module allows creating a database which will store all the information needed by user.

**4.1.1 FRONT END**

This is how the front end of this application will look like:



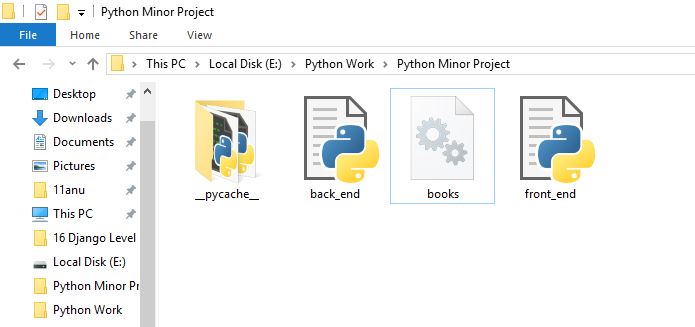
So it consists of many widgets like:

* Entry or Text boxes
* Labels
* List boxes
* Buttons
* Scrollbar

These widgets make it easier for a user to use it and they form the part of Graphical User Interface (GUI).

**4.1.2 BACK END**

The back end part of this application is the database containing a relation having information about the stored books. Information about database is present in a file called books.db which is shown below in the folder.



Here, the file books.db is a file of type database and it contains the information about the books in a relational form which is having attributes like:

1. Title
2. Author
3. Year
4. ISBN
5. ID

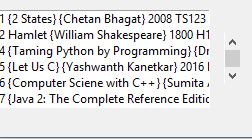
So tuples of this relation contains information about each book stored in this database. Thus, the file books.db forms the back end part of this application.

**4.2 FUNCTIONALITIES OF THIS APPLICATION**

This application is used to enter information about books and store it in a database for later reference. This application performs following functionalities:

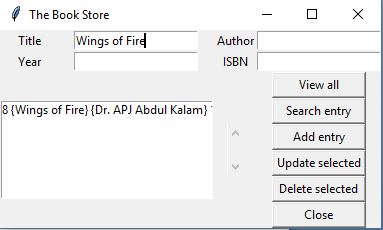
1. **View all Button:**

This button is used to view at once record of all the books stored in the database of this application. On clicking this button record of entries of books stored will be shown in a list box. Its output is shown below. The scroll bar is used to see more entries present in database.



1. **Search Entry Button:**

This button is used to search an entry in the database by entering the value of any of its attribute like title, author, year, etc. Suppose we are searching the book from it Title name. So we have to enter the name of its title and click on Search Entry button. Its output will be:



1. **Add Entry Button:**

For adding an entry we have to fill the details of the attributes of a book in the textboxes of the application and later click on the Add Entry button to add that entry in database.

1. **Update Selected Button:**

For updating the details of any previously stored entry in the database of this application. We have to first find that entry whose details we have to update and then select that entry. And update the details on textboxes and then click on update selected button to update the details of that entry.

1. **Delete Selected Button:**

For deleting any record or entry from the database of this application we have to first search that entry which we want to delete and then select it in list box and the click on Delete Selected button to delete that entry.

1. **Close Button:**

After all work is done we can close this application either by clicking on the cross button at top right corner of the window of application or we can click on the Close button to close the window of this application.

So these are the functionalities that this application can perform. It is a simple GUI application with limited functionalities. But its simple design makes it easy to use by even a naïve user.

**4.3 SYSTEM REQUIREMENTS**

This is a simple application so it has very few system requirements. It is also Cross Platform which means it can run equally on different platforms like Windows, Linux, UNIX, Mac OS, etc. It has some system requirements which are listed below:

* Python 3.5 or higher version of Python environment must be insstalled in computer.
* Libraries like Tkinter and Sqlite3 must be installed in computer.
* There are no as such space and RAM requirements.