ANALYSIS

1. ANALYSIS:

Currently most of the libraries are manual and some are using computer, but the software they are using is not user friendly and everything you have to perform manually it is so time consuming. The Library Management System is divided into four main sections which can be used to perform different types of operation in library. In the first section we collect information from the students that which book they require. If book is not in the library then we will have to purchase or buy that book to make available for the students. So that this section is used to fulfill the demands of student. In the next section of our project the book which we have purchased is verified and the information of that book is entered in our database. While entering information of that book in our database we consider fields like title of the book, author of the book, pages of the book, and price of the book etc. After that the book will placed at the proper location in the library.

Our project Library Management System is for monitoring and controlling transactions in the library. In the project we mainly focused on basic operations in the library. Creating new membership, adding new books in library, issuing books, returning books, etc. The purpose of the library management system is to maintain library records in simple and sophisticated manner. Our project Library Management System is very easy to use and it will be used by the Librarian.

But the existing system is is not well suited for all the operations performing in the library. So that we are developing the new system for maintaining records and performing operations easy and sophisticated way. After that our next section is use for circulating the books of library with the member or students of the college. We will also create membership of the library, Issuing and returning books, etc. in this section. While creating membership we will gather and store all the information of student in database like student class, course, contact details, and other important information. In our project we can also manage which book is missing and which book is lost from the library. If a book is lost by the student then we will have to calculate fine for the student. In the last section of our project we will keep an eye on every book of library. It means that, how many books are present in library and how many books are issued to the student from the library. In library we will have to keep record of each book number of copies of that book available in library.

DISADVANTAGES

2. DISADVANTAGES:

- The Accession Number of the book in the existing system have to enter manually which take more time.

- Every book has an ISBN number and there is no provision for entering the ISBN number in the existing.

- If any student loses the book of library then we have to calculate fine manually for book which is lost or damaged.

- The existing systems are very slow and time consuming. We have to enter all the data manually.

PROPOSED SYSTEM

3. PROPOSED SYSTEM

Library Management System is a desktop based application which will be used by the Librarian. Everyone needs to go to library once in a day. So librarian has to do lot of work manually. Currently most of the libraries are manual and some are using computer, but the software they are using is not user friendly and everything you have to perform manually it is so time consuming. This project "Library Management System" is developed to automate the task of entering the records and retrieving the details of books available in library. It manages all the data of books in the library. It is a very simple application to handle and anyone can handle it easily in good manner. Library is a place where all kinds of books are available. In the Library Management System there are four menus are present by using that menus we can perform different kinds of operation in the Library. We have four different menus in our project which are used for different purpose. We can perform different types of operation in the library like issuing BT card, Issuing book on BT, Returning book, searching for a book by using its accession number.

When we will have to buy or purchase new books in the library then we will add those books in acquisition. This menu is very useful in the library. For adding the book in acquisition we have a special menu in which using first submenu we will add book into acquisition list. The name of the sub-menu is Add Book. After clicking on Add Book one form will open. On that form we will have to enter the name of the book and the author of that book. We will also mention that how many copies of each book we required as per our need. In this Acquisition menu another menu is used for displaying the list of books which we have added in the first submenu Add Book. The name of this submenu is Books in Acquisition.

When we purchased book from market then we have to add all the details about that books in our database. For this we have another menu named as Cataloging. In this menu there are total four submenus from which we will use first submenu named as Book Entry for entering details in the database. In order to allow multiple copies of the same book each book will have a unique ID number known as Accession Number. There is no need to enter accession number manually because in the database the accession number

will be provided automatically to the book. While entering the data of that book in the database we will considered the fields like, ISBN Number, Name of Book, Author of that book, Price, Pages of the book, Publication, Publishing Year, etc. when we will add the information of the book in our database it will automatically delete the information in the acquisition. It is the cataloging menu of our application. If we have to search details of any book then we can use another submenu of menu Cataloging that is Search submenu. For searching the book in the database we will have to enter the accession number of that book. Then it will show all the information regarding with the book like Name, Author, Pages, Price, Publishing Year, etc. The operations like issuing a book or returning a book will be performed in the Circulation menu but the list of the issued and returned book only will display in this menu. We have a submenu for displaying list of books which was issued by the students from library. When student will issue book then the Accession Number of that book and BT number of that student will be added in the database. When student will return book then that book will be stored in another table of our database. The date of issuing book and returning book will be provided automatically. There is no need to enter date of issuing book or returning book manually. We can see that list of books in the other submenu.

The library must keep track of the status of each media item: its location, status, descriptive attributes, and cost for losses and late returns. Books will be identified by their ISBN. Student cannot issue book from the library without BT. BT is the abbreviation for the Browser's Ticket. So we have to generate the BT card of the student by that student will able to issue book from the library. We can use this sub menu for creating membership, making transaction and doing some other operations in our library. Creating a membership is nothing but the issuing Browser's Ticket to the students. BT Card is valid for current year only. Validity of the BT card is calculated from the current issuing year of BT. There is no need to enter validity of BT manually. While issuing BT we have to fill all information of student so we can assign him a BT. After filling all the information with regarding with student BT number will be generated automatically from the information of student. The dialog box will be shown on the window after clicking on submit button of form. The BT number is consists of course, year and Roll no. of the

student (Eg. BCA313). By using this BT he can issue any book from library. The student can issue only one book on a BT and to each student two BT cards will be provided. Student can issue the book on his BT. When we are issuing book to the student we will have to enter the BT number and Accession Number of that book in our Issue Book form. This system takes issuing date of the book automatically there is no need to fill it manually. Then the book will added in one of the table of our database. When any student wants to return his issued book then we will have to enter his BT number and the Accession Number of that issued book. While returning the book we don't have to enter date of returning it will take automatically and saved into database. This record will be deleted from the issue book table and added to the return book table. This will help the librarian to save his time by deleting data automatically form the issue book table.

Librarian will check the book at the time of returning is there any damage to the book or if the book is lost by any student then it will be processed into our submenu Book Status. Now we will consider the book is lost then we will enter the BT number of the student and select the status lost. Then from that entered BT number it will search which book was issued to the student and fetch the accession number of book. After getting the accession number of the book we will access the table Actual book entry to get the price of that book and the dialog box will be shown on which calculated fine will display. And if student is returni+ng the book and if it is damage then we will enter the BT number and select the status Damage then it will be added into table damage and the dialog box will be shown on which calculated fine will display. If the book is lost then the student will have to pay complete price of that book and if the book is damaged then student will have to pay one fifth price of that book.

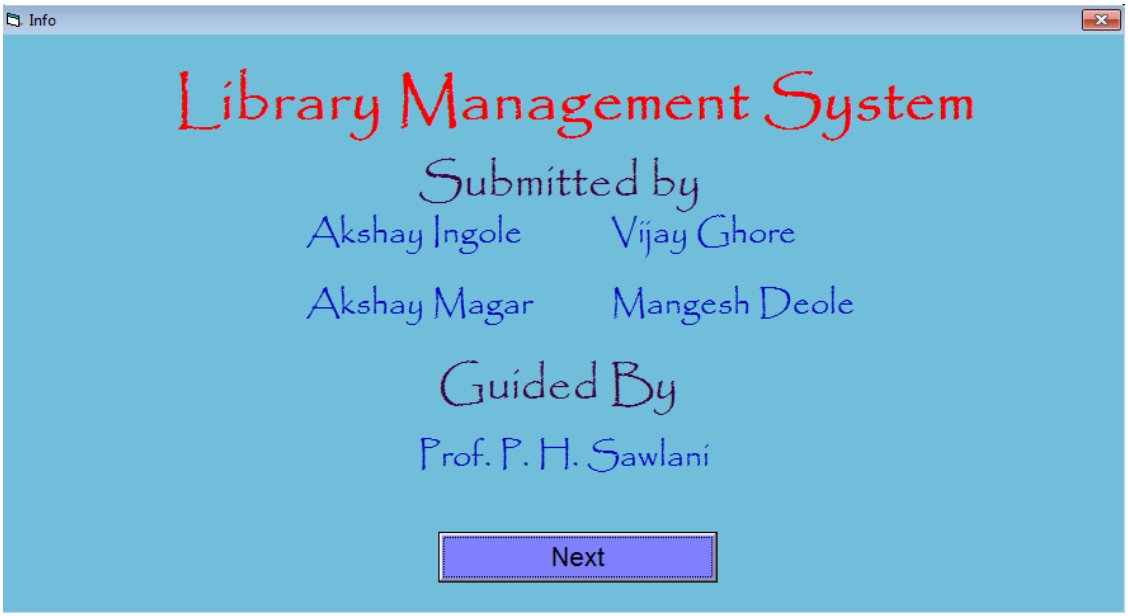
In the last menu we will keep an eye on every book. This menu we have three submenus first is to see how many books are available in the library. By clicking this menu it will show a dialog box which will show the message that how many books are available in the library. In the second submenu we can see how many books are issued to students it will count all the records of the issue book table. This is the need of the Library Management System by this we can minimize the efforts and manual work.

INPUT/OUTPUT FORMS OF PROPOSED SYSTEM

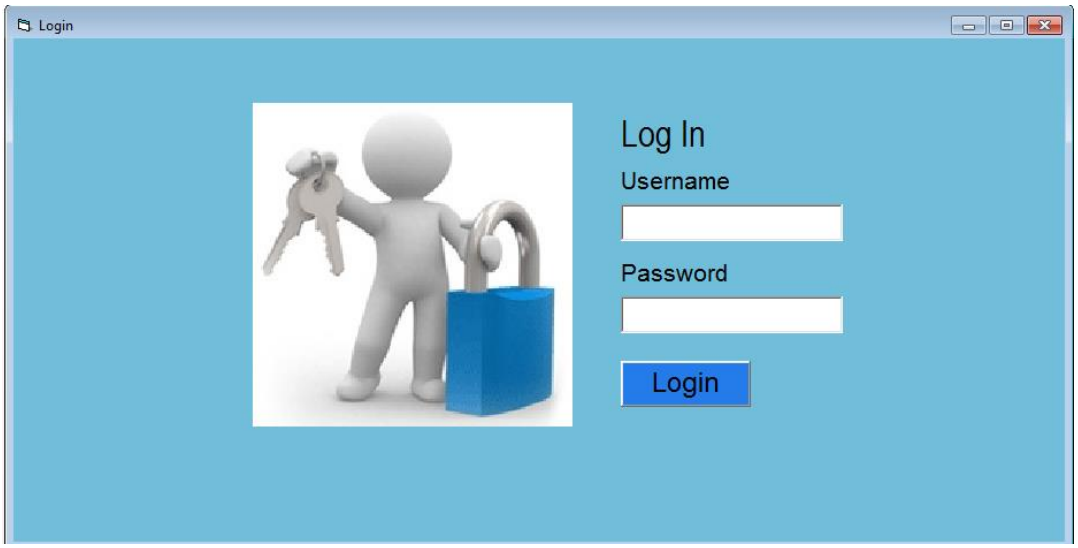
4. INPUT/OUTPUT FORMS OF PROPOSED SYSTEM:

Welcome Form:

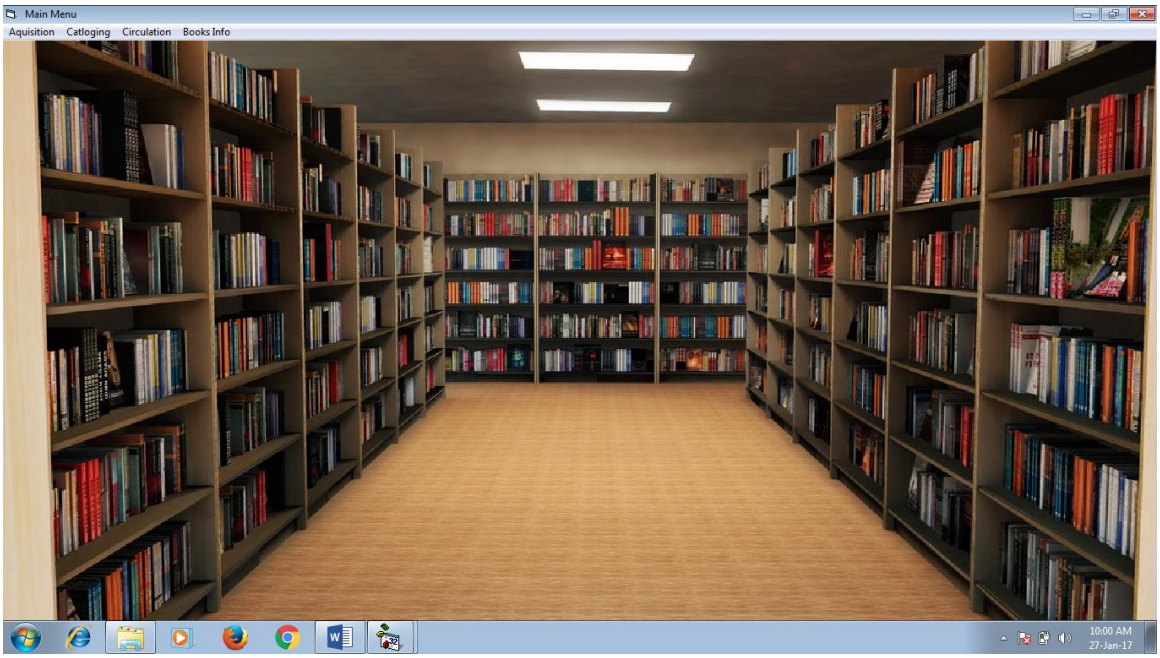
Information Form:



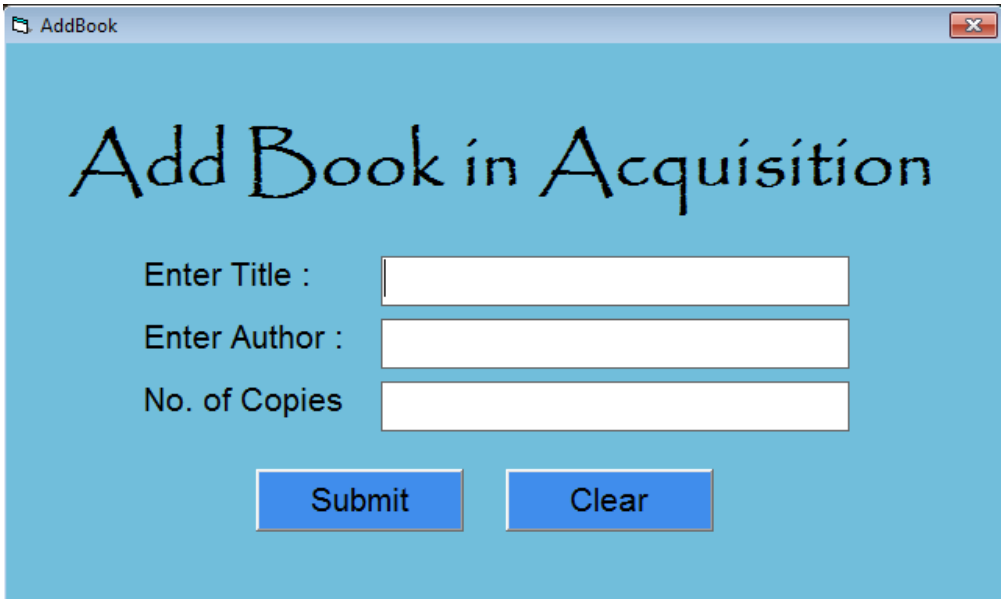
Login Form:



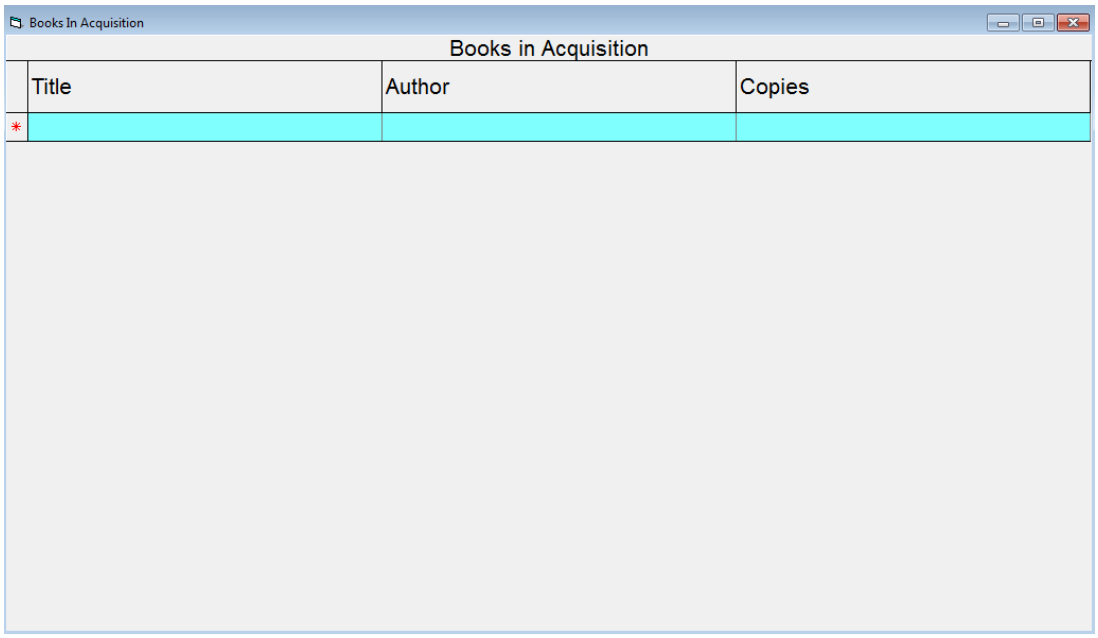
Main Menu Form:



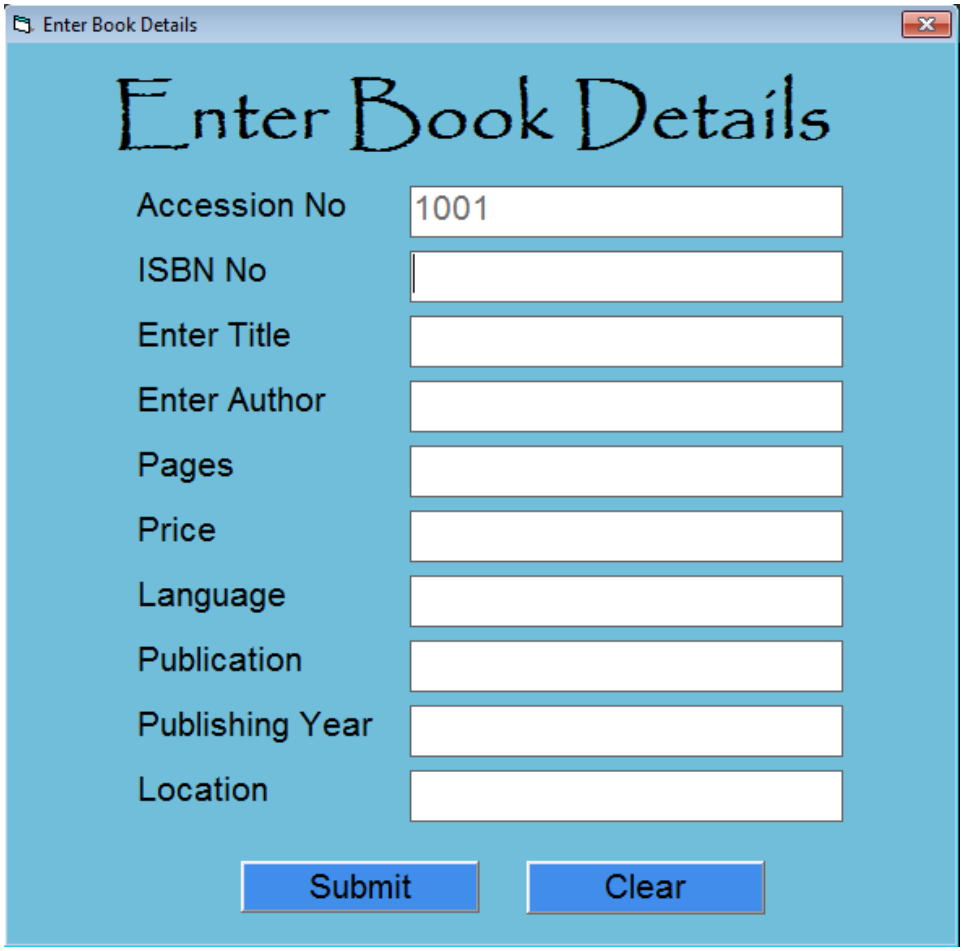
Add Book in Acquisition Form:



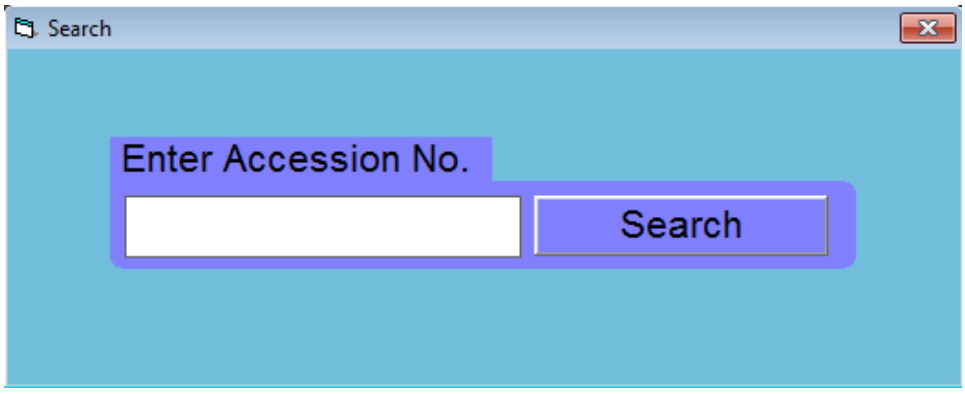
List of Books in Acquisition Form:



Book Entry Form:

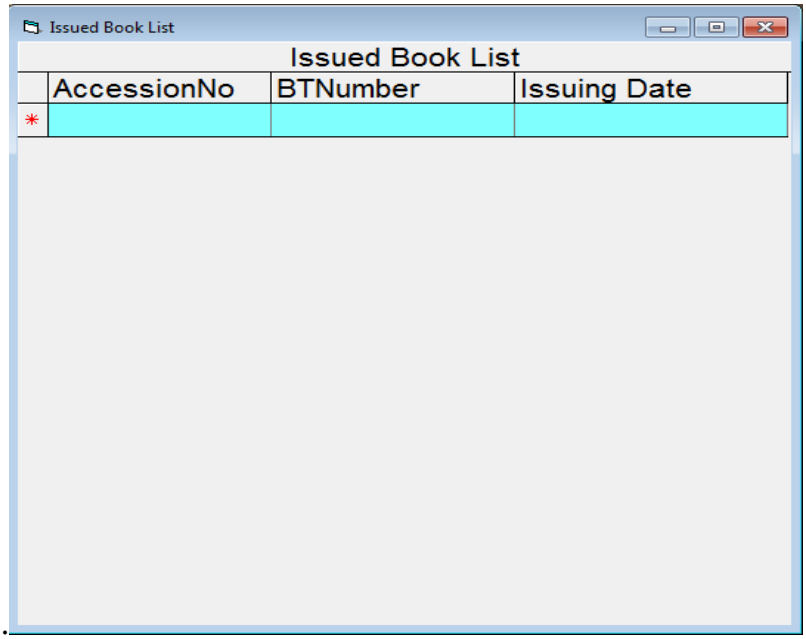


Search Book Form:

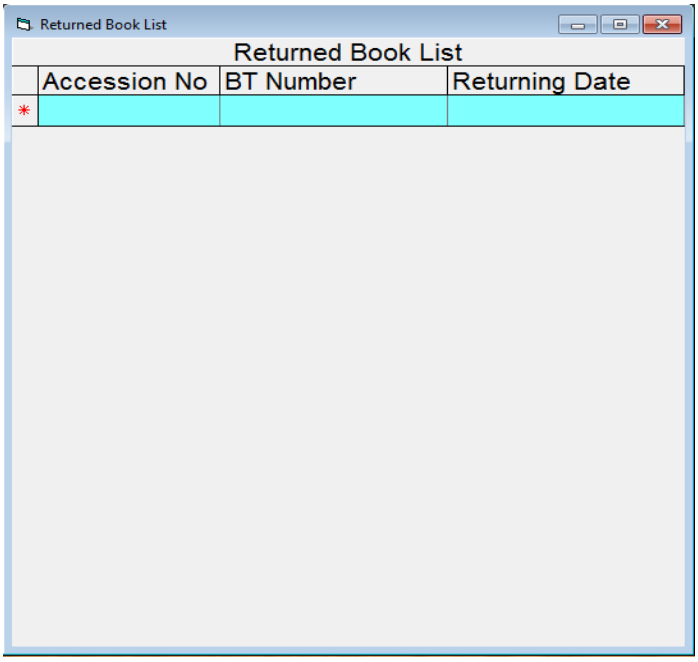


Search Book Detail Form:

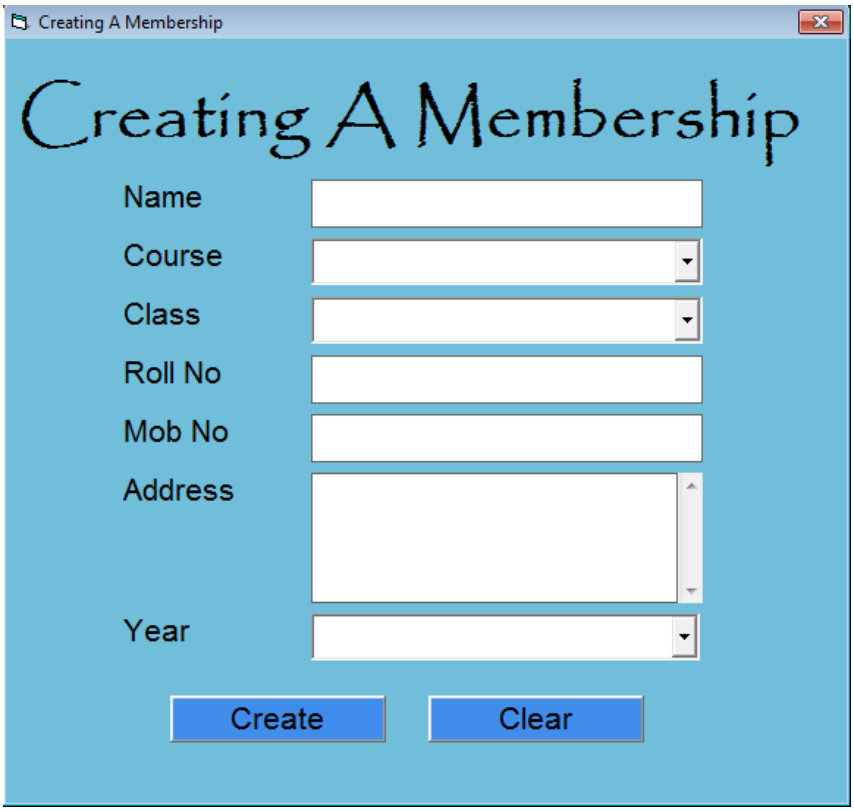
Issue Book List:



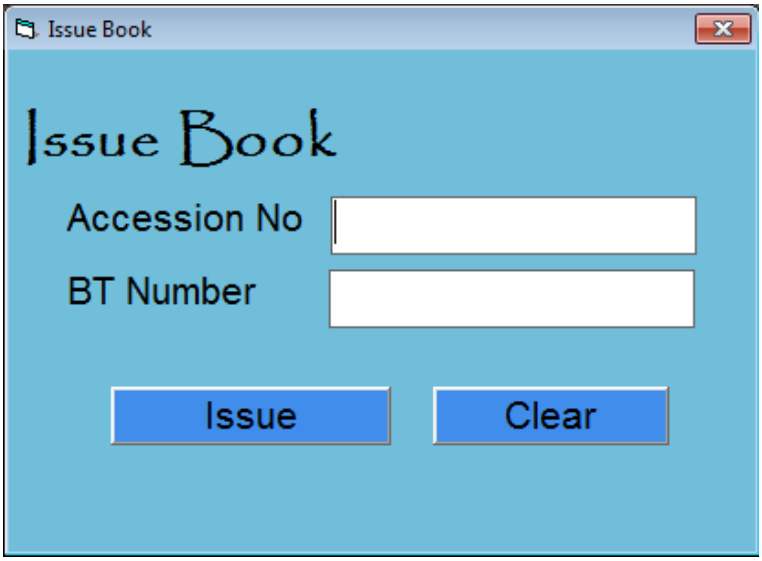
Returned Book List Form:



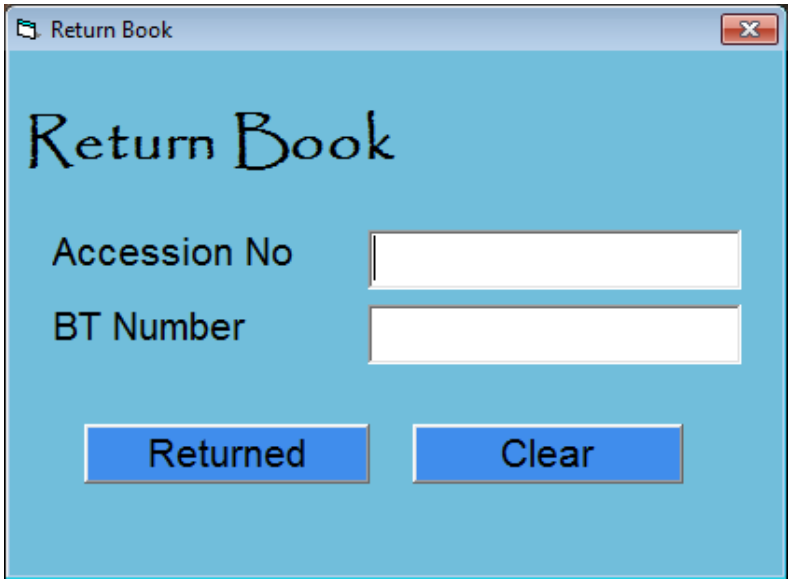
Creating Membership Form:



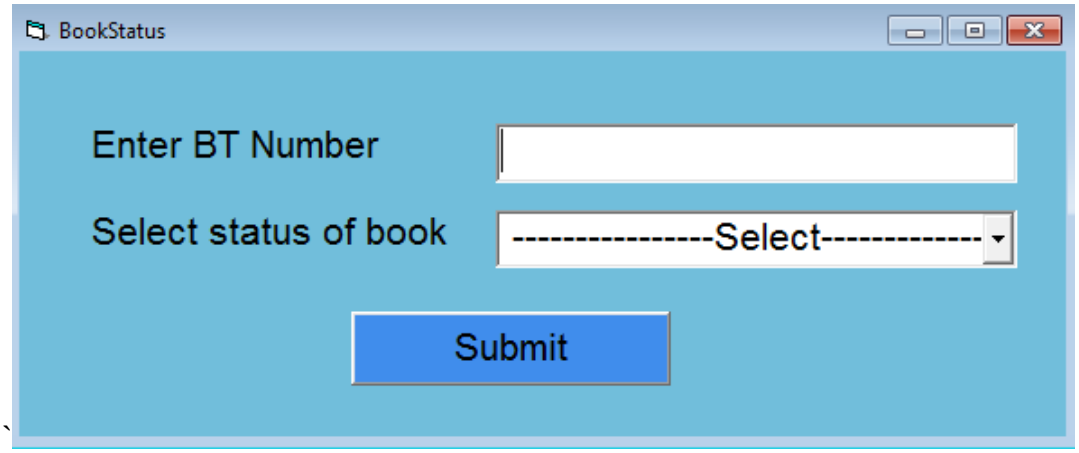
Issue Book Form:



Return Book Form:



Book Status Form:



DATABASE SYSTEM

5. DATABASE SYSTEM:

Table ActualBookEntry:

This table is used to store the complete information about the book. The book we have purchased form market we have to enter information of that book in the table below. The design view of the table is given below.

Field Name	Datatype	Primary Key	Description
AccessionNo	Number	Yes	Accession Number of the book
ISBNNo	Long Text	No	ISBNNo of the book
Title	Long Text	No	Title of the book
Author	Long Text	No	Author Name of the book
Pages	Number	No	Pages of the book
Price	Number	No	Price of the book
Language	Long Text	No	Language of the book
Publication	Long Text	No	Publication of the book
PublishingYear	Number	No	Year of Publishing book
Location	Long Text	No	Place of book in library

Table AddBook:

This table is used when we have to make list of book which we have to purchase for the library. We have to enter title, author and copies of book. Which is stored in this table. The design view of the table is shown below.

Field Name	Datatype	Primary Key	Description
Title	Long Text	No	Title of the book
Author	Long Text	No	Author of that book
Copies	Number	No	Copies required of that book

Table BookEntry:

This is a dummy table from which transaction of library will occurs. This is an important table in the database which will use in performing transaction. This table is exactly similar to the table ActualBookEntry. The design view of the table is shown below.

Field Name	Datatype	Primary Key	Description
AccessionNo	Number	Yes	Accession Number of the book
ISBNNo	Long Text	No	ISBNNo of the book
Title	Long Text	No	Title of the book
Author	Long Text	No	Author Name of the book
Pages	Number	No	Pages of the book
Price	Number	No	Price of the book
Language	Long Text	No	Language of the book
Publication	Long Text	No	Publication of the book
PublishingYear	Number	No	Year of Publishing book
Location	Long Text	No	Place of book in library

Table CreatingMembership:

In this table we will fill the information regarding with the student. While creating membership of the student we have to consider fields which are shown in the following table.

Field Name	Datatype	Primary Key	Description
FullName	Long Text	No	Name of the Student
Course	Long Text	No	Course of the student
Class	Long Text	No	Class of the student
RollNo	Number	Yes	Roll number of the student
MobNo	Long Text	No	Mobile number of student
Address	Long Text	No	Contact detail of student
ForYear	Short Text	No	Academic Year of student
BTNumber	Long Text	Yes	BT Number of the student
Validity	Long Text	No	Validity of the student's BT

Table Damage:

When student issue any book and while returning if it is damaged then entry of that book have to made in this book in this table. We will consider only two fields while entering data. The design view of table is shown below.

Field Name	Datatype	Primary Key	Description
BTNumber	Long Text	No	BT number of student who issue that book
AccessionNo	Number	No	Accession Number of book which is issued

Table IssueBook:

When any student issue book from the library then we have to maintain records of student. In this table we will store the information of students who issue the book. While entering the records we will consider three fields form which we have to enter manually two fields. The design view of the table is shown below.

Field Name	Datatype	Primary Key	Description
AccessionNo	Number	No	Accession Number of book which have to issue
BTNumber	Long Text	No	BTNumber of student to whom book will issue
IssuingDate	Long Text	No	On which date book is issued

Table Lost:

In this table we will fill information of the students who have lost their books which was issued form the library. The design view of the table is shown in following table.

Field Name	Datatype	Primary Key	Description
BTNumber	Long Text	No	BT number of student who issue that book
AccessionNo	Number	No	Accession Number of book which is issued

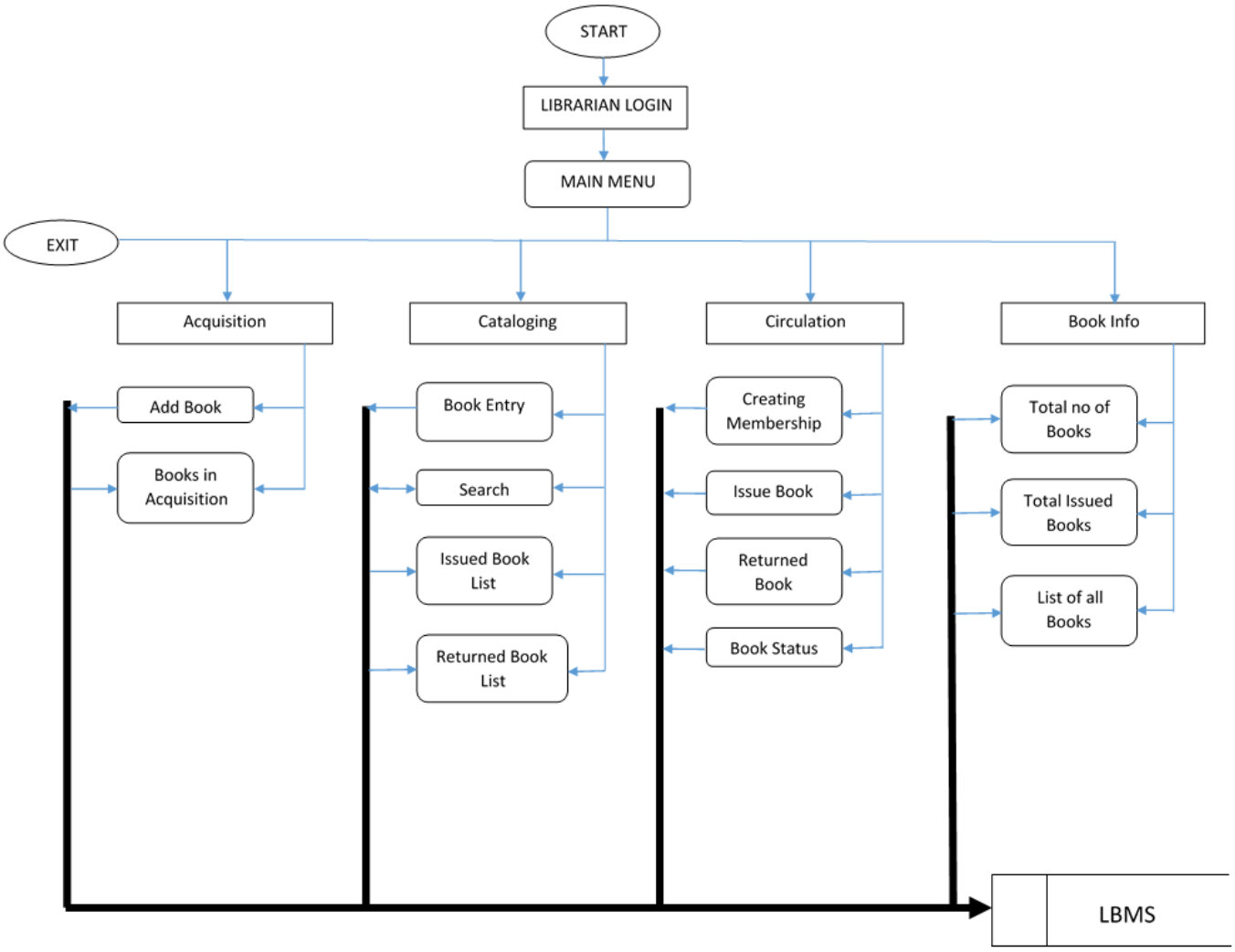
Table Returning Book:

When any student return book which was issued to him that information will be filled in this table. The design view of the ReturnBook table is shown below.

Field Name	Datatype	Primary Key	Description
AccessionNo	Number	No	Accession Number of book which have to return
BTNumber	Long Text	No	BTNumber of student who is returning book
IssuingDate	Long Text	No	On which date book is Returned

DATA FLOW DIAGRAM (DFD)

6. DATA FLOW DIAGRAM (DFD):



TECHNOLOGY USED FOR PROPOSED SYSTEM

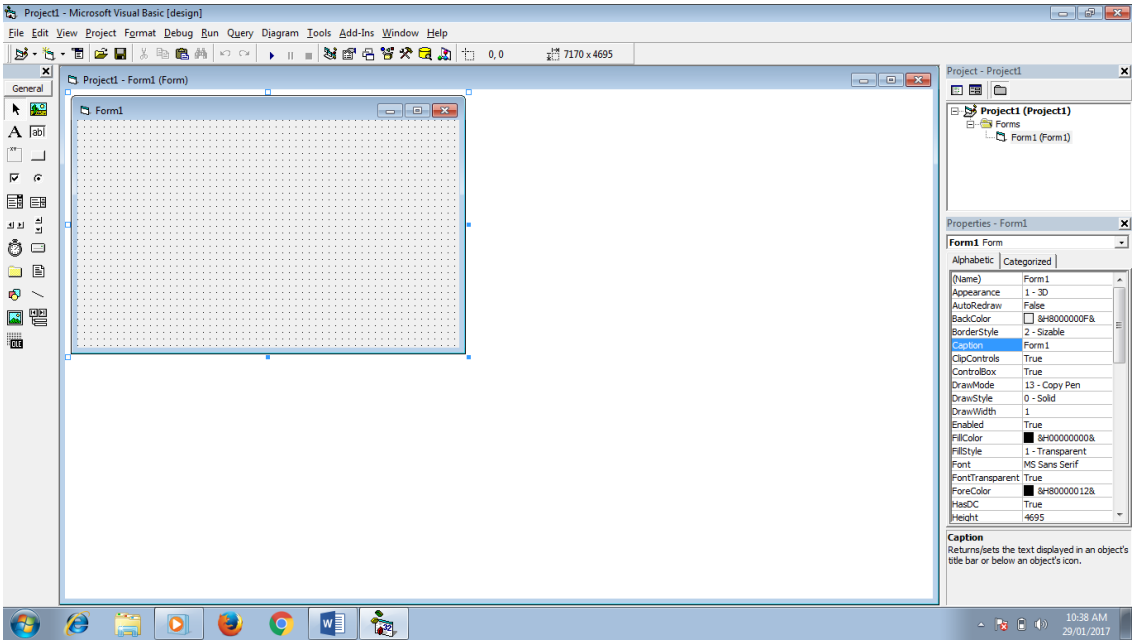
7. TECHNOLOGY USED FOR PROPOSED SYSTEM

7.1 Front End: VISUAL BASIC 6.0

Visual Basic is a third-generation event-driven programming language and integrated development environment (IDE) from Microsoft for its Component Object Model (COM) programming model first released in 1991 and declared legacy in 2008. Microsoft intended Visual Basic to be relatively easy to learn and use. Visual Basic was derived from BASIC, a user-friendly programming language designed for beginners, and it enables the rapid application development (RAD) of graphical user interface (GUI) applications, access to databases using Data Access Objects, Remote Data Objects, or ActiveX Data Objects, and creation of ActiveX controls and objects. A programmer can create an application using the components provided by the Visual Basic program itself. Over time the community of programmers developed third party components. Programs written in Visual Basic can also use the Windows API, which requires external function declarations.

Visual Basic can create executable (EXE files), ActiveX controls, or DLL files, but is primarily used to develop Windows applications and to interface database systems. Dialog boxes with less functionality can be used to provide pop-up capabilities. Controls provide the basic functionality of the application, while programmers can insert additional logic within the appropriate event handlers. For example, a drop-down combination box automatically displays a list. When the user selects an element, an event handler is called that executes code that the programmer created to perform the action for that list item. Alternatively, a Visual Basic component can have no user interface, and instead provide ActiveX objects to other programs via Component Object Model (COM). This allows for server-side processing or an add-in module.

The final release was version 6 in 1998 (now known simply as Visual Basic). On April 8, 2008 Microsoft stopped supporting Visual Basic 6.0 IDE. The Microsoft Visual Basic team still maintains compatibility for Visual Basic 6.0 applications on Windows Vista, Windows Server 2008 including R2, Windows 7, Windows 8, Windows 8.1, Windows Server 2012 and Windows 10 through its "It Just Works" program. In 2014, some software developers still preferred Visual Basic 6.0 over its successor, Visual Basic .NET. In 2014 some developers lobbied for a new version of Visual Basic 6.0. In 2016, Visual Basic 6.0 won the technical impact award at The 19th Annual D.I.C.E. Awards. A dialect of Visual Basic, Visual Basic for Applications (VBA), is used as a macro or scripting language within several Microsoft applications, including Microsoft Office. IDE of visual basic is shown in following figure.


Project1 - Microsoft Visual Basic [design]
File Edit View Project Format Debug Run Query Diagram Tools Add-Ins Window Help
Project1 - Form1 (Form)
Form1
Project - Project1
Project1 (Project1)
Forms
Form1 (Form1)
Properties - Form1
Form1 Form
Alphabetic
Categorized
(Name) Form1
Appearance 1 - 3D
AutoRedraw False
BackColor &H8000000F&
BorderStyle 2 - Sizable
Caption Form1
ClipControls True
ControlBox True
DrawMode 13 - Copy Pen
DrawStyle 0 - Solid
DrawWidth 1
Enabled True
FillColor &H00000000&
FillStyle 1 - Transparent
Font MS Sans Serif
FontTransparent True
ForeColor &H80000012&
HasDC True
Height 4695
Caption
Returns/sets the text displayed in an object's title bar or below an object's icon.
10:38 AM
29/01/2017

Features:

- Support for numerous data types including arrays, Boolean, integer, floating point, string and variant for complex data types.
- Support for referencing OLE object and calling DLLs. Support for named arguments.
- Visual basic includes entire all the application feature set and extends it further. Both tools share the same language engine, but visual basic provides a superset of the Microsoft visual basic for application development environment.
- Microsoft visual basic provides so many tools in its IDE including debugger and object browser, and can also integrate with additional third-party tools such as version control and test utilities. Developers using Microsoft visual basic for application can integrate OLE object into a custom solution.
- Microsoft visual basic developers can integrate, create, and distribute OLE object in a network environment. Finally, Microsoft visual basic for application must be included or "hosted" within an application such as Microsoft Excel, while Microsoft visual basic a standalone development tools.
- Developers can write code that can be reused across all the application that include Microsoft visual basic for application. Developers who want to write portable Microsoft visual basic for application implements discussed previously.
- Developers should also fully declare all object references in their Microsoft visual basic for application code. Fully declaring an object reference means there is no ambiguity about which object the Microsoft visual basic for application code refers to. Developers can fully declare an object reference by using both the library (application) name and the object name of the object in the form.

  <Application name>.<object name>

Benefits

- Visual basic application are event driven, it means the user is in control of the application. The user generates a stream of event each time he clicks with mouse or press the key on the keyboard. The visual basic application respond to those events though the code written and attach to those events.

- Visual basic support the principle of object oriented designs, it means that one can separate into isolated categories of its application as objects independently of the rest application.

- The greatest advantage of using objects is encapsulation, which are the capabilities to wrap all aspects of functionality and user interface into a single entry.

- A visual basic object is a programmable entity of one sort or another.

- Visual basic has been designed to be a complete windows applications development system. Visual basic is infinitely extensible through the use of ActiveX control dynamically linked.

7.2 Back End: Microsoft Access 2013:

Microsoft Access is the default database of Microsoft Visual Basic. Microsoft Access 2003 provides many new features that make working with data and designing a database even easier. Microsoft Access database is a collection of data and objects related to particular topic or purpose. Microsoft Access Database may contain tables, queries, forms, report, and macros module and shortcuts top data access pages. Microsoft Access is a Relational Database Management System. Using Access we can organize our data according to subject and can store information about how different subject are related. In general MS-Access database can have several small tables.

Microsoft office Access, previously known as Microsoft Access, is Relational Database Management System. From Microsoft that combines the Relational Microsoft Jet Database Engine with a graphical user interface and software development tools. It is a member of the 2007 Microsoft Office system. Access can use data stored in Access/Jet, Microsoft SQL Server, Oracle, or any ODBC-compliant data container. Skilled software developers and data architects use it to develop application software. Relatively unskilled programmers and non-programmers "power users" can use it to build simple applications. It supports some object-oriented techniques but falls short of being a fully object-oriented development tool.

Access was also the name of a communications programs from Microsoft, meant to complete with ProComm and other programs. This proved a failure and was dropped. Years later Microsoft reused the name for its database software. Access is used by small business, within departments of large co-operations, and by hobby programmers to create ADHOC customize desktop systems for handling the creation and manipulation of data Access can be used as a database for basic web based applications hosted on Microsoft Internet Information Services and utilizing Microsoft active Server Pages ASP. Most typical web applications should use tools like ASP/Microsoft SQL Server or the LAMP stack.

Features:

- Access is a window application and therefore has an interface similar to Windows.NET. You can cut, copy, and paste data from any windows application.
- Access maintain a single disk file for a database and all its associated objects. The maximum size of this file can be 1 GB. Access allows a maximum of 32 GB, 768 tables and any other DBMS like FoxPro. Therefore the maximum size of table and object is limited only the amount of storage space that user have. Access a powerful DBMS capable of handling large volumes of data, spread across several database and users.
- Access lets you import from or export to FoxPro, Excel, Oracle and other data formats. Importing creates an Access table, exporting an Access table created a file in the format that you are exporting it linking means that you can use external data without creating an Access table. You can link to FoxPro, Excel and ASCII data.
- For those user who do not want to write programs. Access provides macros. Macros let you perform common task without user interventions. Let you manipulate data, open forms and automate many tasks that is respective or complex.
- Reports present on your data in a printed format. You can create different types of reports with a DBMS. You can even generate a report that lists selective records. Your reports can combine multiple tables to present complex relationship among different sets of data. When you can design your database, keep in mind the information that you want printed. Doing so ensure that all the information you required in your reports is easily available.

Benefits:

- It is most commonly used database creating tool.
- It is a very friendly and an interactive tool.
- Help is given in such way that user can learn the whole tool itself without any special training.
- Selection of database pages is given that is blank database, database access page, and other many forms with different categories.
- All the formatting facilities are available.
- Also you can edit the database very easily.
- You can also put validation without writing any code.
- You can create the database by wizard. Any special training.
- Page and other many forms with different categories.

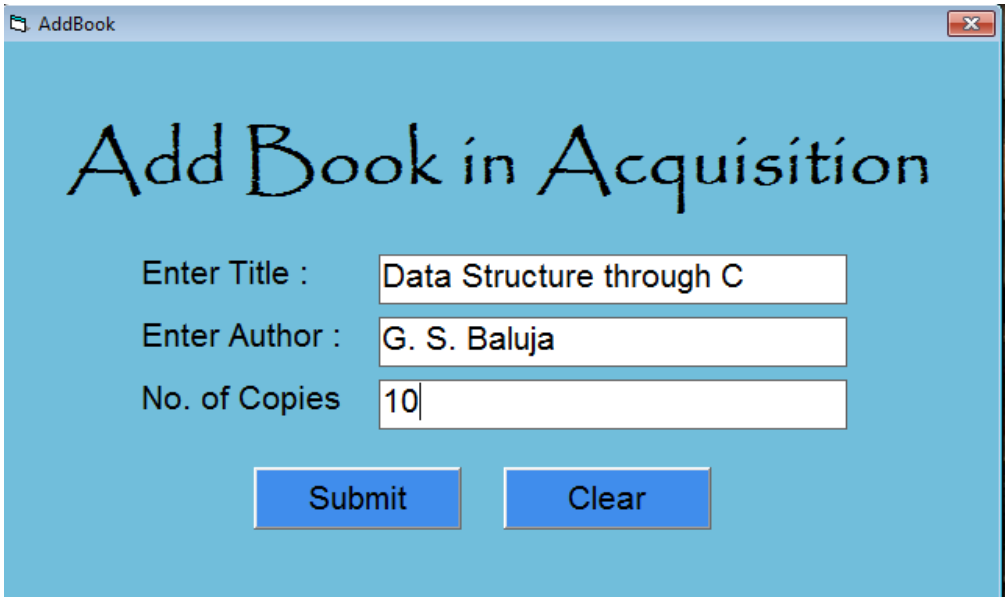
CODING

8. CODING:

Login Form:


```
Private Sub BtnLogin_Click()

   If Me.TxtUserName.Text = "" Or Me.TxtPassword.Text = "" Then

      MsgBox "Please enter your Username & Password", vbOKOnly + vbCritical, “Login Failed"

   Else

      If TxtUserName.Text = "123" And TxtPassword.Text = "123" Then

         MainMenu.Show

         Unload Me

      Else

         MsgBox "Please Enter Correct Username & Password", vbOKOnly + vbCritical, "Login Failed"

      End If

   End If

End Sub
```

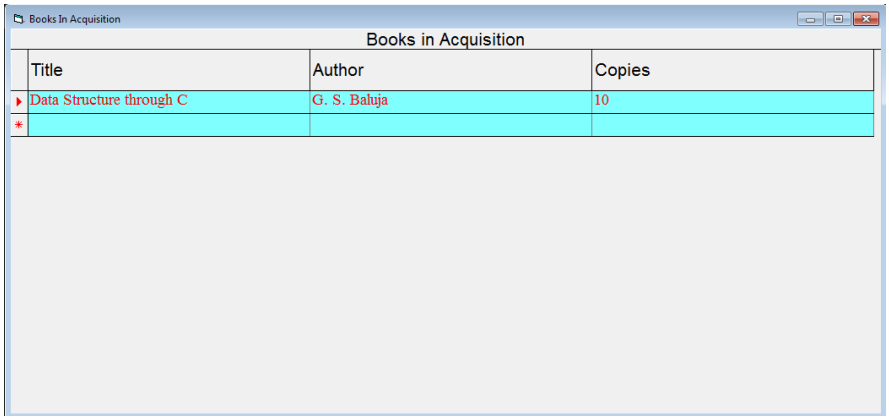
AddBook Form:




```
Private Sub BtnClear_Click()
    TxtTitle.Text = ""
    TxtAuthor.Text = ""
    TxtCopies.Text = ""
End Sub

Private Sub BtnSubmit_Click()
    Dim con As New ADODB.Connection
    Dim cmd As New ADODB.Command

    Dim Title, Author As String
    Dim Copies As Integer

    Title = TxtTitle.Text
    Author = TxtAuthor.Text
```

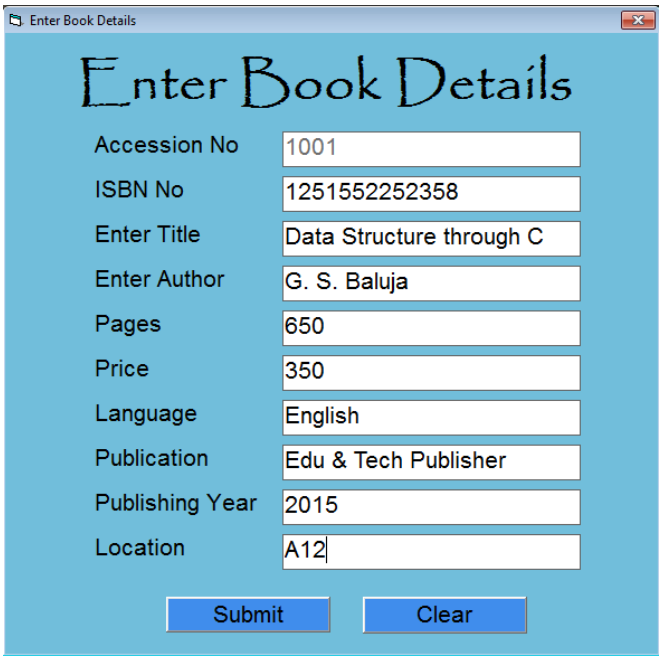
```
    Copies = CInt(Me.TxtCopies.Text)
    Dim sql As String
    sql = "insert into AddBook values('" & Title & "' , '" & Author & "'," & Copies & ")"
    Dim cs As String
    cs = "provider=Microsoft.jet.oledb.4.0;data source=" & App.Path & "\LBMS.mdb"
    con.ConnectionString = cs
    con.Open
    cmd.ActiveConnection = con
    cmd.CommandType = adCmdText
    cmd.CommandText = sql
    Dim n As Integer
    cmd.Execute n
    con.Close
    If n = 1 Then
      mb = MsgBox("Book Added...", vbOKOnly + vbInformation, "Message")
    Else
       mb = MsgBox("Cannot add the book..", vbOKOnly + vbCritical, "Message")
    End If
End Sub
```

Book in Acquisition:




```
Private Sub Form_Load()

    Adodc1.ConnectionString = "provider=Microsoft.jet.oledb.4.0;data source=" &
    App.Path & "\LBMS.mdb;Persist Security Info=False"

    Adodc1.RecordSource = "select * from AddBook"

    Set DataGrid1.DataSource = Adodc1

End Sub
```

Book Entry Form:




```
Private Sub BtnClear_Click()
    Me.TxtAccessionNo.Text = ""
    Me.TxtISBNNo.Text = ""
    Me.TxtTitle.Text = ""
    Me.TxtAuthor.Text = ""
    Me.TxtPages.Text = ""
    Me.TxtPrice.Text = ""
    Me.TxtLanguage.Text = ""
    Me.TxtPublication.Text = ""
    Me.TxtPublishingYear.Text = ""
    Me.TxtLocation.Text = ""
```

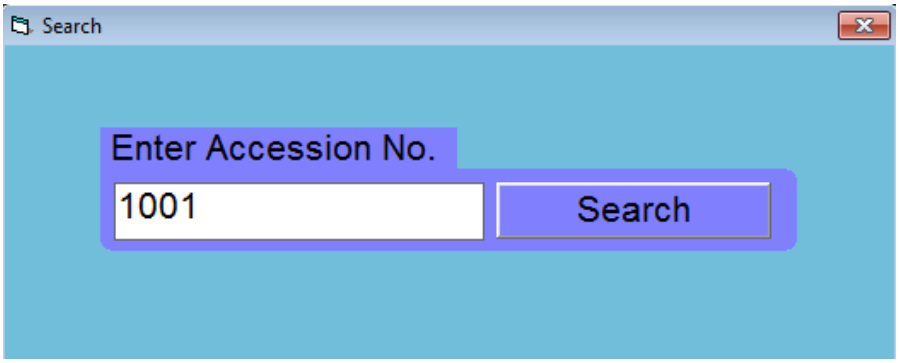
```
    Dim rs As New ADODB.Recordset
    Dim con As New ADODB.Connection

    con.Open "provider=Microsoft.jet.oledb.4.0;data source=" & App.Path &
"\LBMS.mdb;persist security info=false"
    Set rs = con.Execute("select max(AccessionNo) from ActualBookEntry")
    rs.Requery
    TxtAccessionNo.Text = rs.Fields(0) + 1
End Sub

Private Sub BtnSubmit_Click()
    Dim con As New ADODB.Connection
    Dim cmd As New ADODB.Command
    Dim ISBNNo, Title, Author, Language, Publication, Location As String
    Dim AccessionNo, Pages, Price, PublishingYear As Integer

    AccessionNo = BookEntry.TxtAccessionNo.Text
    ISBNNo = BookEntry.TxtISBNNo.Text
    Title = BookEntry.TxtTitle.Text
    Author = BookEntry.TxtAuthor.Text
    Pages = CInt(BookEntry.TxtPages.Text)
    Price = CInt(BookEntry.TxtPrice.Text)
    Language = BookEntry.TxtLanguage.Text
    Publication = BookEntry.TxtPublication.Text
    PublishingYear = CInt(BookEntry.TxtPublishingYear.Text)
```

```
    Location = BookEntry.TxtLocation.Text


    Dim sql As String

    sql = "insert into BookEntry values(" & AccessionNo & ", '" & ISBNNo & "', '" &
Title & "', '" & Author & "', " & Pages & ", " & Price & ", '" & Language & "', '" &
Publication & "', " & PublishingYear & ", '" & Location & "')"


    Dim cs As String

    cs = "provider=Microsoft.jet.oledb.4.0;data source=" & App.Path & "\LBMS.mdb"


    con.ConnectionString = cs

    con.Open

    cmd.ActiveConnection = con

    cmd.CommandType = adCmdText

    cmd.CommandText = sql


    Dim n As Integer

    cmd.Execute n

    con.Close

'-----------------------------------------------------------------------------------------------------

    Dim sqldel As String

    sqldel = "delete from AddBook where Title = '" & Title & "' AND Author= '" &
Author & "'"


    con.ConnectionString = cs

    con.Open
```

```
    cmd.ActiveConnection = con
    cmd.CommandType = adCmdText
    cmd.CommandText = sqldel

    Dim n1 As Integer
    cmd.Execute n1
    con.Close
'-----------------------------------------------------------------------------------------------
    Dim sqladd As String
    sqladd = "insert into ActualBookEntry values(" & AccessionNo & ", '" & ISBNNo & "', '" & Title & "', '" & Author & "', " & Pages & ", " & Price & ", '" & Language & "', '" & Publication & "', " & PublishingYear & ", '" & Location & "')"

    con.ConnectionString = cs
    con.Open
    cmd.ActiveConnection = con
    cmd.CommandType = adCmdText
    cmd.CommandText = sqladd

    Dim n2 As Integer
    cmd.Execute n2
    con.Close
    If n = 1 Then
       MsgBox "Book Added in the Database", vbOKOnly + vbInformation, "Add Book"
    Else
       MsgBox "Unable to Add in the Database", vbOKCancel, "Add Book"
```

```
    End If
End Sub


Private Sub Form_Load()
    Dim rs As New ADODB.Recordset
    Dim con As New ADODB.Connection

    con.Open "provider=Microsoft.jet.oledb.4.0;data source=" & App.Path &
"\LBMS.mdb;persist security info=false"
    Set rs = con.Execute("select max(AccessionNo) from ActualBookEntry")
    rs.Requery
    TxtAccessionNo.Text = rs.Fields(0) + 1
End Sub
```

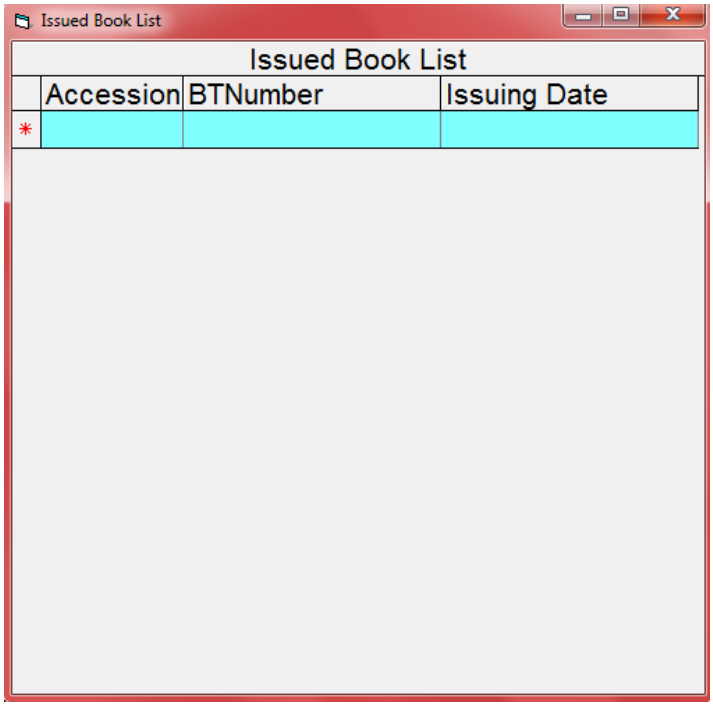
Search Form:




```
Private Sub BtnSearch_Click()

    Dim n, MaxBook As Integer

    n = Me.TxtEnterAccessionNo.Text


    Dim rs1 As New ADODB.Recordset

    Dim con1 As New ADODB.Connection

    con1.Open "provider=Microsoft.jet.oledb.4.0;data source=" & App.Path &
"\LBMS.mdb;persist security info=false"

    Set rs1 = con1.Execute("select max(AccessionNo) from ActualBookEntry")

    rs1.Requery

    MaxBook = rs1.Fields(0) + 1


    If n = "" Then

       MsgBox "Please enter Accession No", vbOKOnly + vbInformation, "Search"

    Else

       If n > 1000 And n < MaxBook Then

          Dim rs As New ADODB.Recordset

          Dim con As New ADODB.Connection
```

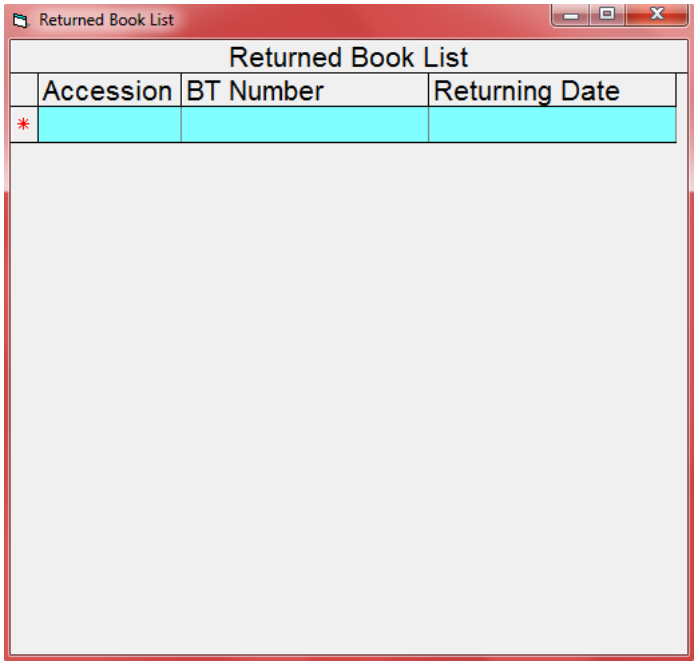
```
        con.Open "provider=Microsoft.jet.oledb.4.0;data source=" & App.Path &
"\LBMS.mdb;persist security info=false"

        Set rs = con.Execute("select * from ActualBookEntry where AccessionNo=" & n)

        rs.Requery

        BookDetails.LblShowAccessionNo.Caption = rs.Fields(0)

        BookDetails.LblShowISBNNo.Caption = rs.Fields(1)

        BookDetails.LblShowTitle.Caption = rs.Fields(2)

        BookDetails.LblShowAuthor.Caption = rs.Fields(3)

        BookDetails.LblShowPages.Caption = rs.Fields(4)

        BookDetails.LblShowPrice.Caption = rs.Fields(5)

        BookDetails.LblShowLanguage.Caption = rs.Fields(6)

        BookDetails.LblShowPublication.Caption = rs.Fields(7)

        BookDetails.LblShowPublishingYear.Caption = rs.Fields(8)

        BookDetails.LblShowLocation.Caption = rs.Fields(9)

        Load BookDetails

        BookDetails.Show

      Else

        mb = MsgBox("Enter Valid Accession Number", vbOKOnly + vbExclamation,
"Search")

        If mb = vbOK Then

          Me.TxtEnterAccessionNo.Text = ""

        End If

      End If

   End If

End Sub
```

IssueBook List Form:



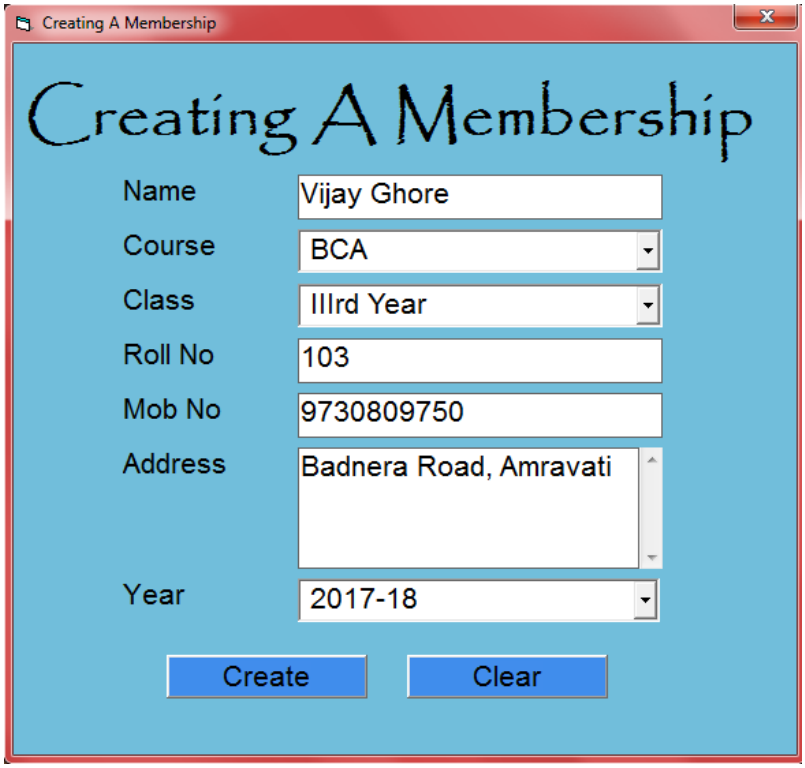

```
Private Sub Form_Load()

    Adodc1.ConnectionString = "provider=Microsoft.jet.oledb.4.0;data source=" &
App.Path & "\LBMS.mdb;Persist Security Info=False"

    Adodc1.RecordSource = "select * from IssueBook"

    Set DataGrid1.DataSource = Adodc1

End Sub
```

ReturnBook List Form:




```
Private Sub Form_Load()

    Adodc1.ConnectionString = "provider=Microsoft.jet.oledb.4.0;data source=" &
App.Path & "\LBMS.mdb;Persist Security Info=False"

    Adodc1.RecordSource = "select * from ReturnBook"

    Set DataGrid1.DataSource = Adodc1

End Sub
```

Creating A Membership Form:




```
Private Sub BtnCreate_Click()

    Dim con As New ADODB.Connection

    Dim cmd As New ADODB.Command

    Dim Name, Class, BClass, Course, MobNo, Address, IssueDate, BTNumber, Validity,
BRollNo, Year, RollNo As String

    Name = Me.TxtName.Text

    Class = Me.CBClass.Text

    Course = Me.CBCourse.Text

    RollNo = Me.TxtRollNo.Text

    MobNo = Me.TxtMobNo.Text

    Address = Me.TxtAddress.Text

    Year = Me.CBYear.Text

    IssueDate = Date
```

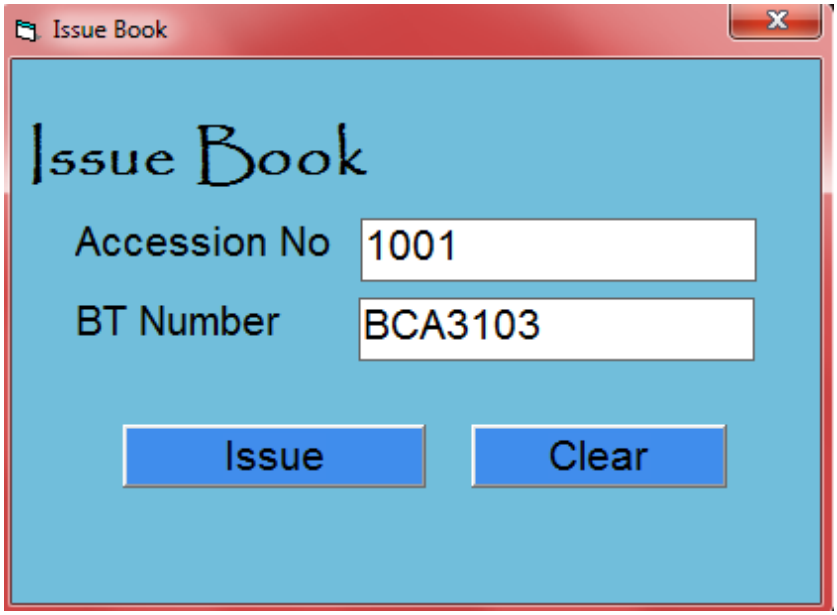
```
If Me.CBYear.Text = "2017-18" Then
   Validity = "30/04/18"
Else
   If Me.CBYear.Text = "2018-19" Then
      Validity = "30/04/19"
   Else
      If Me.CBYear.Text = "2019-20" Then
         Validity = "30/04/20"
      Else
         If Me.CBYear.Text = "2020-21" Then
            Validity = "30/04/21"
         Else
            If Me.CBYear.Text = "2021-22" Then
               Validity = "30/04/22"
            Else
               If Me.CBYear.Text = "2022-23" Then
                  Validity = "30/04/23"
               Else
                  If Me.CBYear.Text = "2023-24" Then
                     Validity = "30/04/24"
                  Else
                     If Me.CBYear.Text = "2024-25" Then
                        Validity = "30/04/25"
                     End If
```

```
                End If
            End If
        End If
    End If
End If

If IsNumeric(Me.TxtName) Then
    MsgBox "Enter valid Name", vbOKOnly + vbCritical, "Error"
Else
    If IsNumeric(Me.TxtRollNo.Text) Then
        If Len(Me.TxtMobNo.Text) < 10 Then
            MsgBox "Enter valid mobile no.", vbOKOnly + vbCritical, "Error"
        Else
            If Class = "Ist Year" Then
                BClass = "1"
            Else
                If Class = "IInd Year" Then
                    BClass = "2"
                Else
                    If Class = "IIIrd Year" Then
                        BClass = "3"
                    End If
                End If
```

```
        End If

        BRollNo = CStr(RollNo)

        BTNumber = Course & BClass & BRollNo


        Dim sql As String

        sql = "insert into CreatingMembership values('" & Name & "','" & Course &
"','" & Class & "', " & RollNo & ", '" & MobNo & "','" & Address & "','" & Year & "','" &
IssueDate & "','" & BTNumber & "','" & Validity & "')"

        Dim cs As String

        cs = "provider=Microsoft.jet.oledb.4.0;data source=" & App.Path &
                                                                "\LBMS.mdb

        con.ConnectionString = cs

        con.Open

        cmd.ActiveConnection = con

        cmd.CommandType = adCmdText

        cmd.CommandText = sql


        Dim n As Integer

        cmd.Execute n

        con.Close

        If n = 1 Then

           mb = MsgBox("Your BT Number is " & BTNumber, vbOKOnly +
vbInformation, "Membership Created")

           If mb = vbOK Then

              Me.TxtAddress.Text = ""

              Me.TxtMobNo.Text = ""
```

```
                Me.TxtName.Text = ""
                Me.TxtRollNo.Text = ""
                Me.CBCourse.Text = ""
                Me.CBClass.Text = ""
                Me.CBYear.Text = ""
            End If
        Else
            MsgBox "Unable to Create MemberShip", vbOKCancel, "Membership"
        End If
    End If
Else
    MsgBox "Enter Valid Roll Number", vbOKOnly + vbCritical, "Error"
End If
End If
End Sub
```

Issue Book Form:

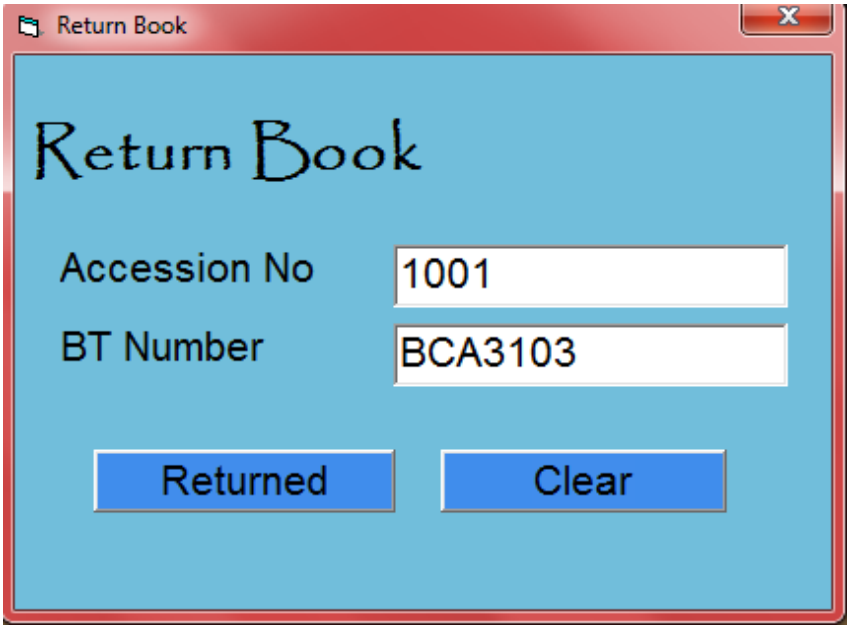



```
Private Sub BtnClear_Click()
    Me.TxtAccessionNo = ""
    Me.TxtBTNumber = ""
End Sub

Private Sub BtnIssue_Click()
    Dim con As New Connection
    Dim cmd As New Command
    Dim AccessionNo As Integer, BTNumber, IDate As String

    AccessionNo = CInt(Me.TxtAccessionNo.Text)
    BTNumber = Me.TxtBTNumber.Text
    IDate = Date

    Dim sql As String
```

```
    sql = "insert into IssueBook values(" & AccessionNo & ", '" & BTNumber & "' , '" & IDate & "')"


    Dim cs As String

    cs = "provider=Microsoft.jet.oledb.4.0;data source=" & App.Path & "\LBMS.mdb"


    con.ConnectionString = cs

    con.Open

    cmd.ActiveConnection = con

    cmd.CommandType = adCmdText

    cmd.CommandText = sql


    Dim n As Integer

    cmd.Execute n

    con.Close

'-----------------------------------------------------------------------------------------------

    Dim sqldel As String

    sqldel = "delete from BookEntry where AccessionNo = " & CInt(Me.TxtAccessionNo)


    con.ConnectionString = cs

    con.Open

    cmd.ActiveConnection = con

    cmd.CommandType = adCmdText

    cmd.CommandText = sqldel
```

```
    Dim n1 As Integer
    cmd.Execute n1
    con.Close
    If n = 1 Then
        mb = MsgBox("Book Issued...", vbOKOnly, "Message")
    Else
        mb = MsgBox("Book can't be Issue....", vbOKOnly, "Message")
    End If

    Me.TxtAccessionNo = ""
    Me.TxtBTNumber = ""
End Sub
```

Returned Book Form:



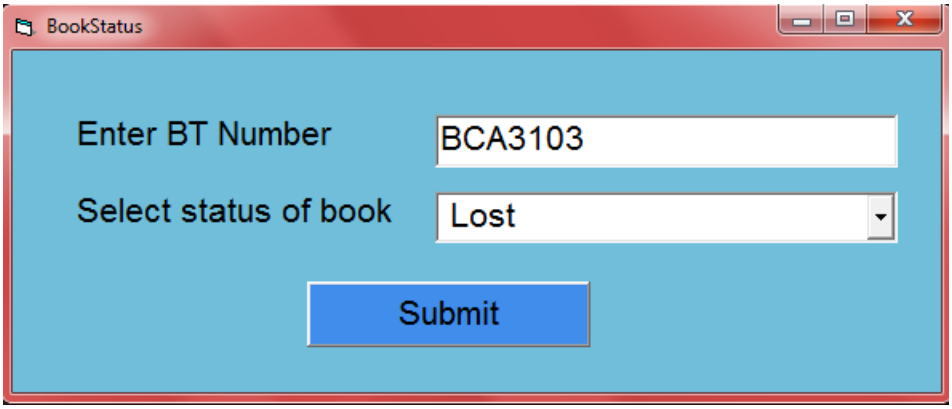

```
Private Sub BtnClear_Click()
    Me.TxtAccessionNo = ""
    Me.TxtBTNumber = ""
End Sub

Private Sub BtnReturned_Click()
    Dim con As New ADODB.Connection
    Dim cmd As New ADODB.Command
    Dim AccessionNo, BTNumber, RDate As String

    AccessionNo = Me.TxtAccessionNo.Text
    BTNumber = Me.TxtBTNumber.Text
    RDate = Date
```

```
    Dim sql As String
    sql = "insert into ReturnBook values('" & AccessionNo & "', '" & BTNumber & "' , '" & RDate & "')"
    Dim cs As String
    cs = "provider=Microsoft.jet.oledb.4.0;data source=" & App.Path & "\LBMS.mdb"

    con.ConnectionString = cs
    con.Open
    cmd.ActiveConnection = con
    cmd.CommandType = adCmdText
    cmd.CommandText = sql

    Dim n As Integer
    cmd.Execute n
    con.Close
    If n = 1 Then
       MsgBox "Book Returned..", vbOKOnly + vbInformation, "Message"
    Else
      MsgBox "Book can't be Return", vbOKOnly + vbCritical, "Message"
    End If
'-------------------------------------------------------------------------------------------------------
    Dim sqldel As String
    sqldel = "delete from IssueBook where BTNumber = '" & BTNumber & "'"

    con.ConnectionString = cs
```

```
    con.Open
    cmd.ActiveConnection = con
    cmd.CommandType = adCmdText
    cmd.CommandText = sqldel

    cmd.Execute n
    con.Close
'-----------------------------------------------------------------------------------------------------
    Dim rs As New ADODB.Recordset
    Dim ISBNNo, Title, Author, Language, Publication, Location As String
    Dim Pages, Price, PublishingYear As Integer

    con.Open "provider=Microsoft.jet.oledb.4.0;data source=" & App.Path &
"\LBMS.mdb;persist security info=false"
    Set rs = con.Execute("select * from ActualBookEntry where AccessionNo=" &
AccessionNo)
    rs.Requery

    AccessionNo = rs.Fields(0)
    ISBNNo = rs.Fields(1)
    Title = rs.Fields(2)
    Author = rs.Fields(3)
    Pages = rs.Fields(4)
    Price = rs.Fields(5)
    Language = rs.Fields(6)
    Publication = rs.Fields(7)
```

```
    PublishingYear = rs.Fields(8)

    Location = rs.Fields(9)


    con.Close

    sqlnew = "insert into BookEntry values('" & AccessionNo & "', '" & ISBNNo & "', '" & Title & "', '" & Author & "', " & Pages & ", " & Price & ", '" & Language & "', '" & Publication & "', " & PublishingYear & ", '" & Location & "')"

    con.ConnectionString = cs

    con.Open

    cmd.ActiveConnection = con

    cmd.CommandType = adCmdText

    cmd.CommandText = sqlnew


    cmd.Execute n

    con.Close

    If n = 1 Then

       MsgBox "Book Returned", vbOKOnly + vbInformation, "Returned Book"

    Else

       MsgBox "Book cannot be Returned", vbOKCancel + vbCritical, "Returned Book"

    End If

End Sub
```

BookStatus Form:




```
Private Sub BtnSubmit_Click()

    Dim con As New ADODB.Connection

    Dim rs As New ADODB.Recordset

    Dim rs1 As New ADODB.Recordset


    Dim Status As String, BTNumber As String


    Status = Me.DdlSelStatus.Text

    BTNumber = Me.TxtBTNumber


    con.Open "provider=Microsoft.jet.oledb.4.0;data source=" & App.Path &
"\LBMS.mdb;persist security info=false"

    Set rs = con.Execute("select * from IssueBook where BTNumber = '" & BTNumber &
"'")

    rs.Requery


    Dim AccNo As String

    AccNo = rs.Fields(0)
```

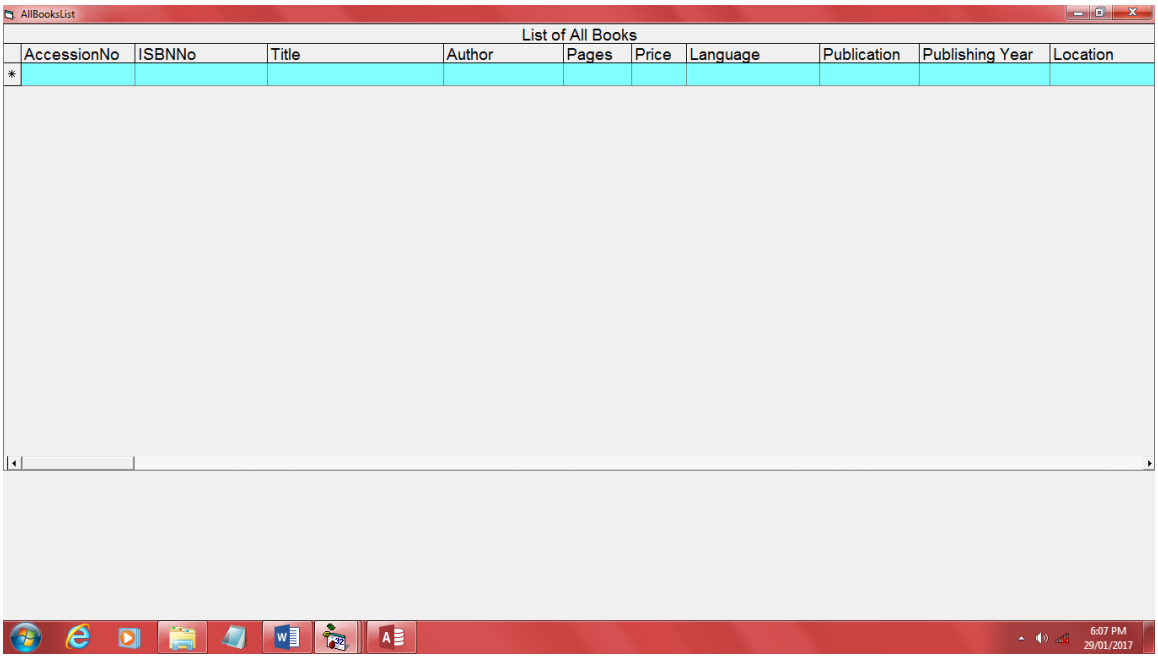
```
    Set rs1 = con.Execute("select * from ActualBookEntry where AccessionNo = " &
AccNo)

    rs1.Requery


    Dim Price As String

    Price = rs1.Fields(4)

    con.Close

'-------------------------------------------------------------------------------------------------------

    Dim con1 As New ADODB.Connection

    Dim cmd As New ADODB.Command


    Dim sql As String

    sql = "insert into " & Status & " values('" & BTNumber & "', " & AccNo & ")"


    Dim cs As String

    cs = "provider=Microsoft.jet.oledb.4.0;data source=" & App.Path & "\LBMS.mdb"


    con1.ConnectionString = cs

    con1.Open

    cmd.ActiveConnection = con1

    cmd.CommandType = adCmdText

    cmd.CommandText = sql


    Dim n As Integer
```

```
    cmd.Execute n
    con1.Close
    If n = 1 Then
        mb = MsgBox("Book " & Me.DdlSelStatus.Text, vbOKOnly, "Message")
    Else
        mb = MsgBox("Book can't be Add", vbOKOnly, "Message")
    End If
'-------------------------------------------------------------------------------------------------------
    If Me.DdlSelStatus.Text = "Lost" Then
        Dim con2 As New ADODB.Connection
        Dim cmd1 As New ADODB.Command

        Dim sqldel As String
        sqldel = "delete from ActualBookEntry where AccessionNo = " & AccNo
        Dim cs1 As String
        cs1 = "provider=Microsoft.jet.oledb.4.0;data source=" & App.Path & "\LBMS.mdb"

        con2.ConnectionString = cs1
        con2.Open
        cmd1.ActiveConnection = con2
        cmd1.CommandType = adCmdText
        cmd1.CommandText = sqldel

        Dim n1 As Integer
        cmd1.Execute n1
```

```
        con2.Close

        MsgBox "Calculated Fine = " & Price, vbOKOnly + vbExclamation, "Fine"

    Else: Me.DdlSelStatus.Text = "Damage"
        Dim df As Integer
        df = (CInt(Price)) / 5

        MsgBox "Calculated Fine = " & df, vbOKOnly + vbExclamation, "Fine"
    End If
'-------------------------------------------------------------------------------------------------
    Dim con3 As New ADODB.Connection
    Dim cmd2 As New ADODB.Command
    Dim sqldel1 As String
    RDate = Date
    sqldel1 = "insert into ReturnBook values(" & AccNo & ", '" & BTNumber & "' ,'" &
RDate & "' )"

    Dim cs2 As String
    cs2 = "provider=Microsoft.jet.oledb.4.0;data source=" & App.Path & "\LBMS.mdb"

    con3.ConnectionString = cs2
    con3.Open
    cmd2.ActiveConnection = con3
    cmd2.CommandType = adCmdText
```

```
    cmd2.CommandText = sqldel1

    Dim n2 As Integer
    cmd2.Execute n2
    con3.Close
'-------------------------------------------------------------------------------------------------------
    Dim con4 As New ADODB.Connection
    Dim cmd3 As New ADODB.Command

    Dim sqldel2 As String
    sqldel2 = "delete from IssueBook where BTNumber = '" & BTNumber & "'"

    Dim cs3 As String
    cs3 = "provider=Microsoft.jet.oledb.4.0;data source=" & App.Path & "\LBMS.mdb"

    con4.ConnectionString = cs3
    con4.Open
    cmd3.ActiveConnection = con4
    cmd3.CommandType = adCmdText
    cmd3.CommandText = sqldel2

    Dim n3 As Integer
    cmd3.Execute n3
    con4.Close
End Sub
```

All Book List Form:



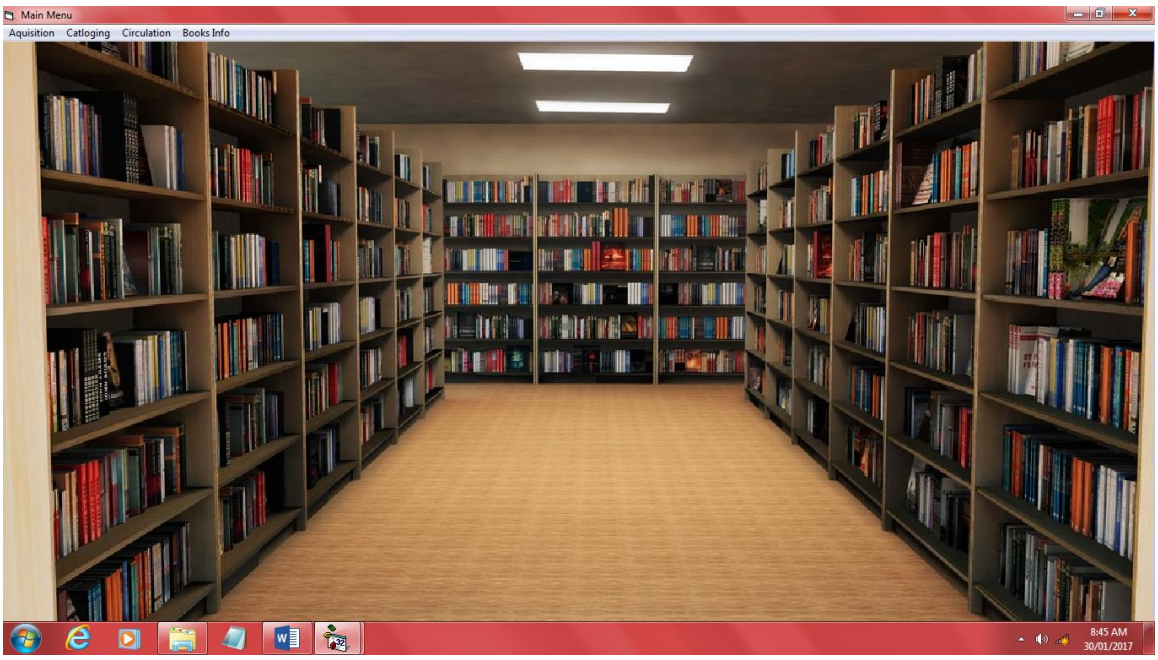

```
Private Sub Form_Load()

    Adodc1.ConnectionString = "provider=Microsoft.jet.oledb.4.0;data source=" &
App.Path & "\LBMS.mdb;Persist Security Info=False"

    Adodc1.RecordSource = "select * from ActualBookEntry"

    Set DataGrid1.DataSource = Adodc1

End Sub
```

SAMPLE INPUT AND OUTPUT

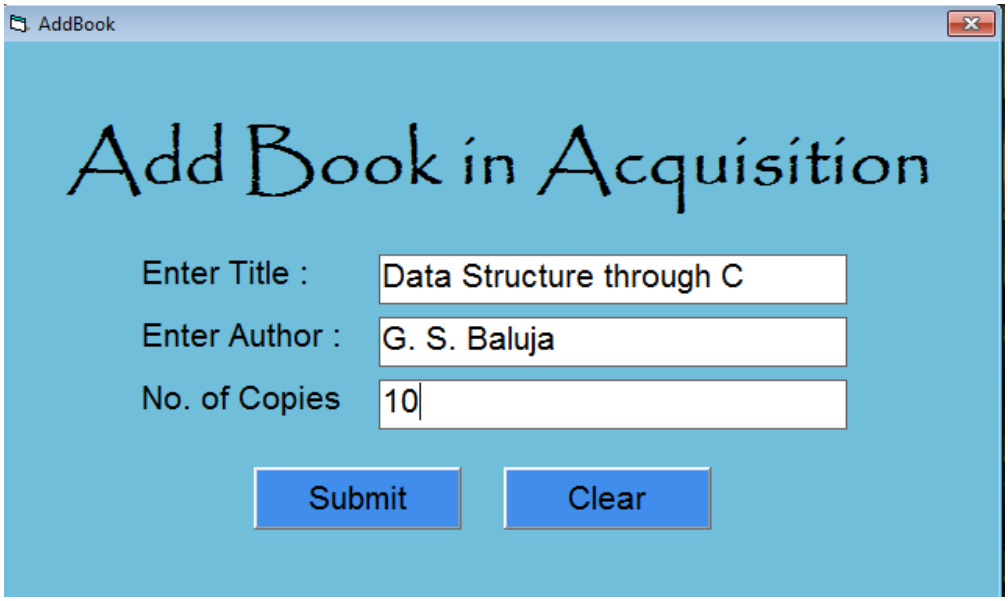
9. SAMPLE INPUT AND OUTPUT:

Login Form Input:

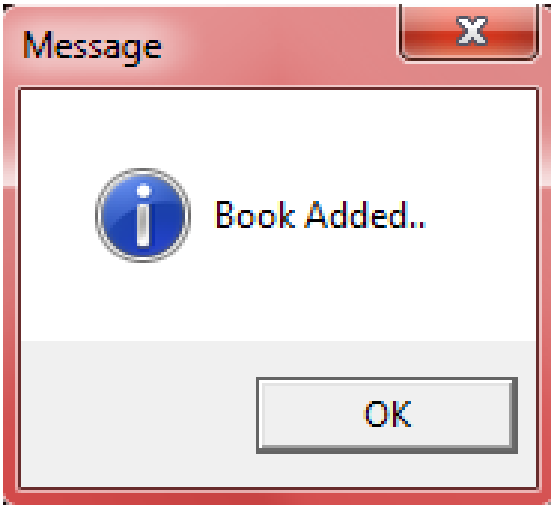
Login Form Output:



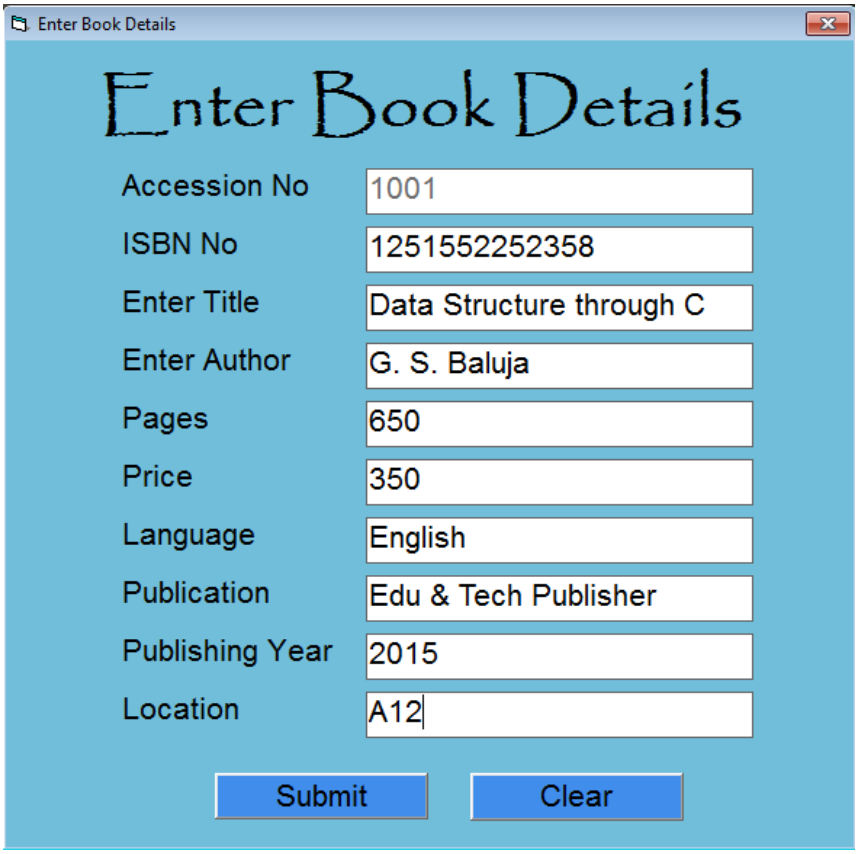
Add Book Form Input:



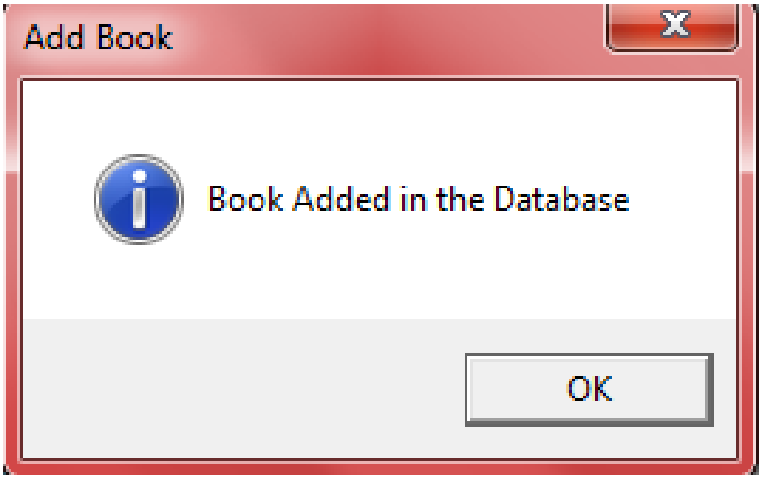
Add Book Form Output:



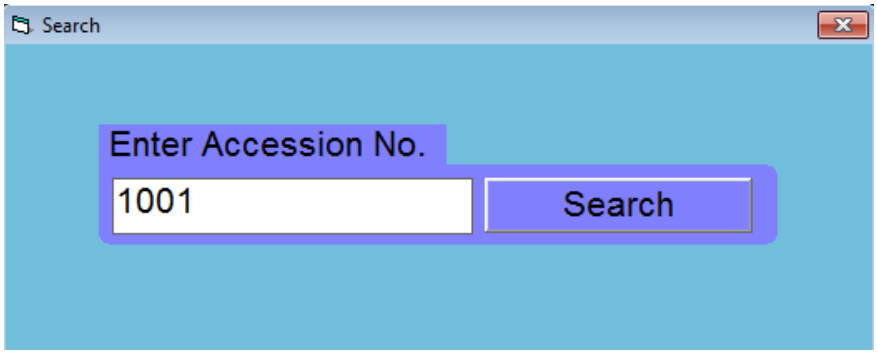
Enter Books Detail Form Input:



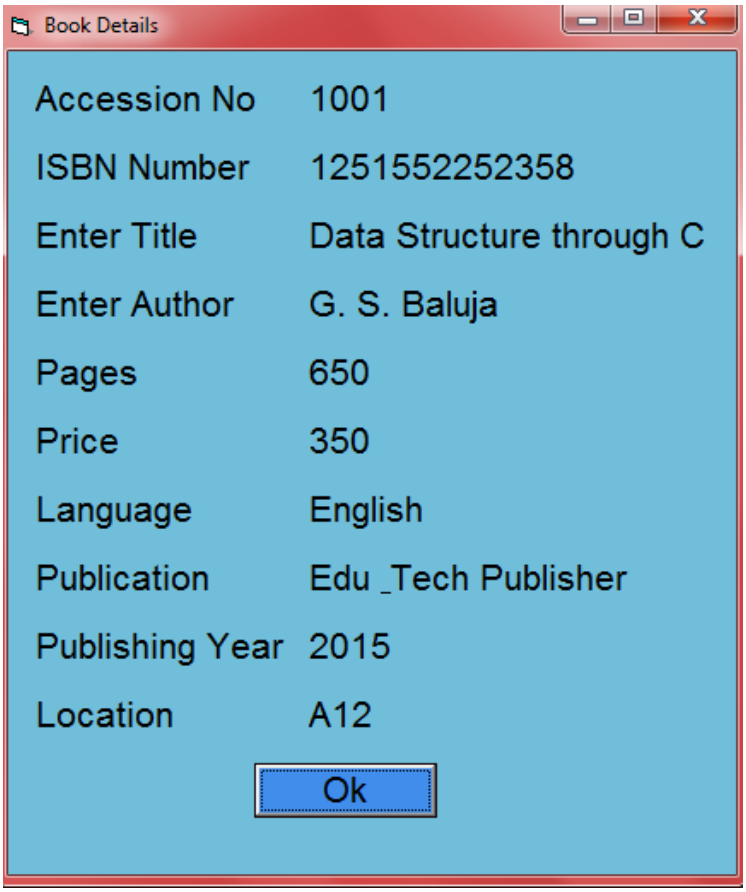
Enter Books Detail Form Output:



Search Form Input:

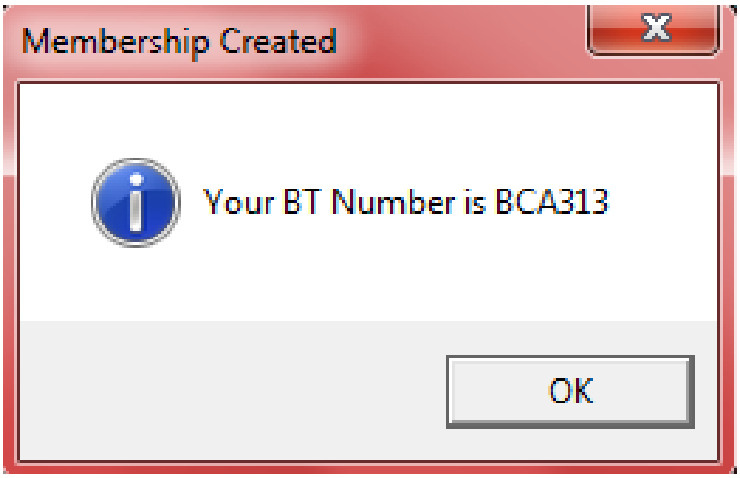


Search Form Output:

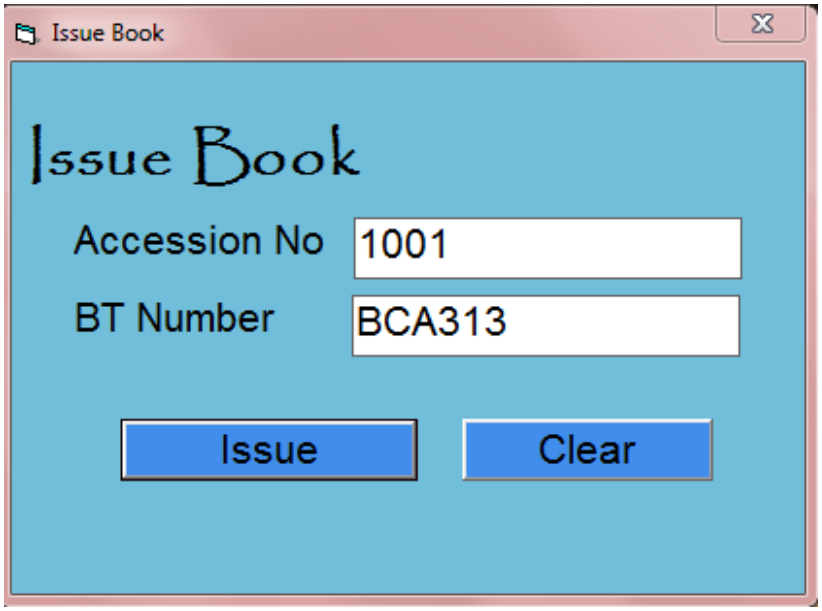


Creating Membership Form Input:

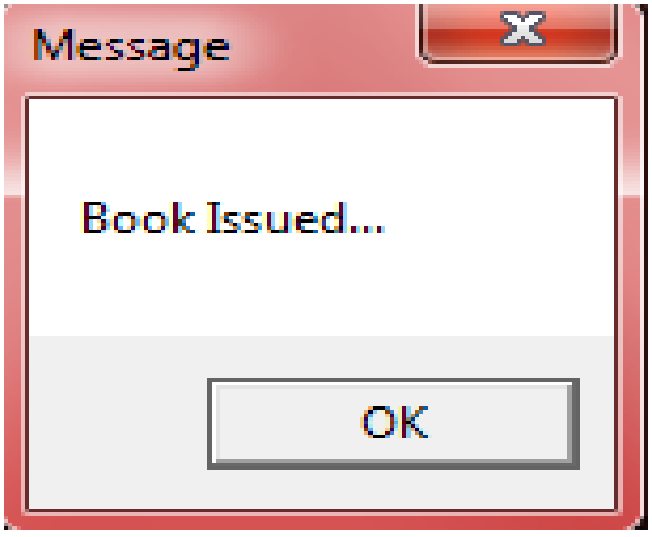
Creating Membership Form Output:



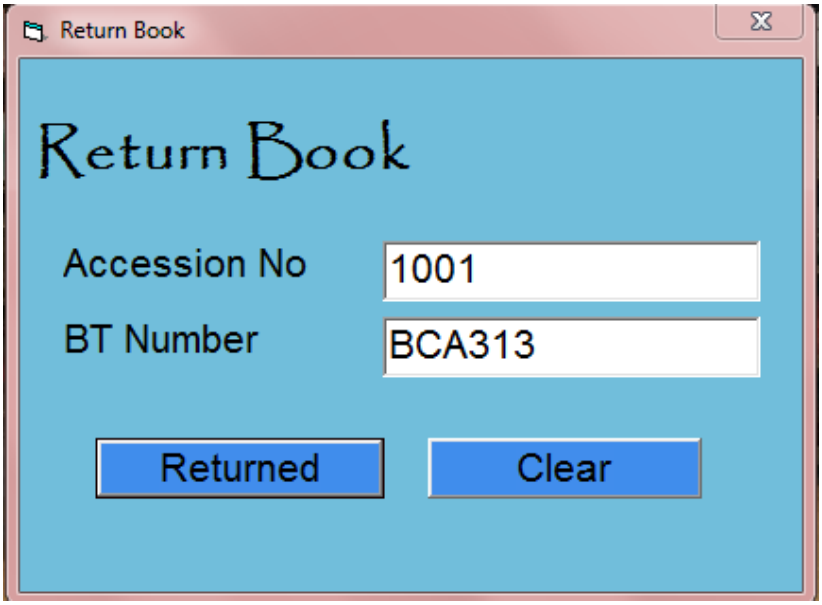
Issue Book Form Input:



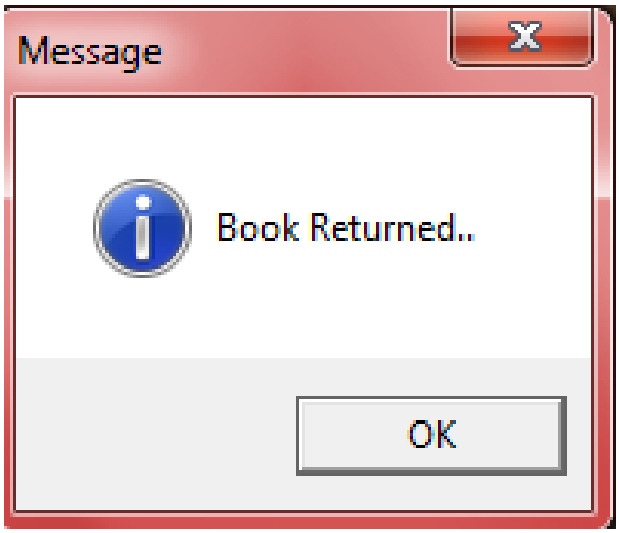
Issue Book Form Output:



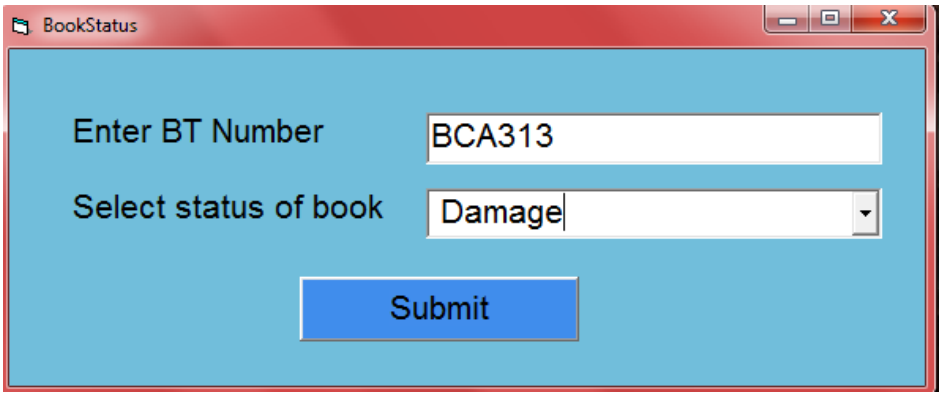
Return Book Form Input:



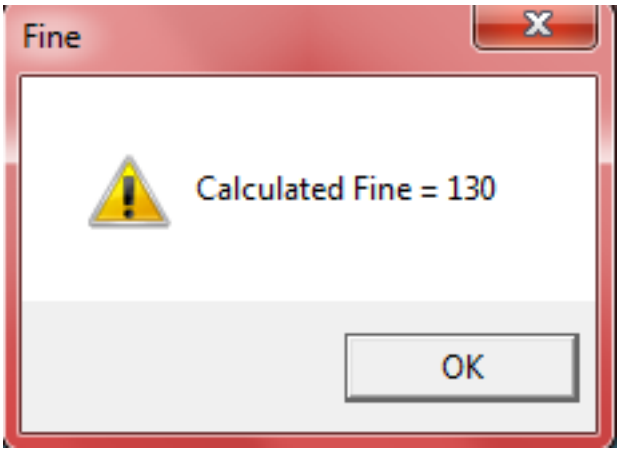
Return Book Form Output:



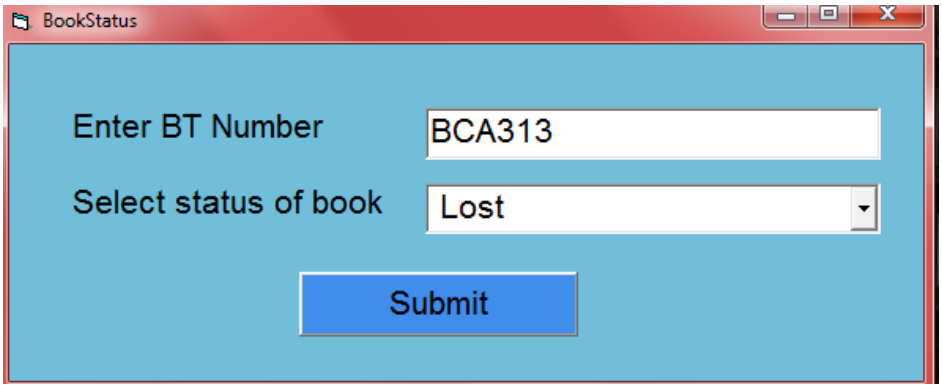
Book Status Form Input 1:



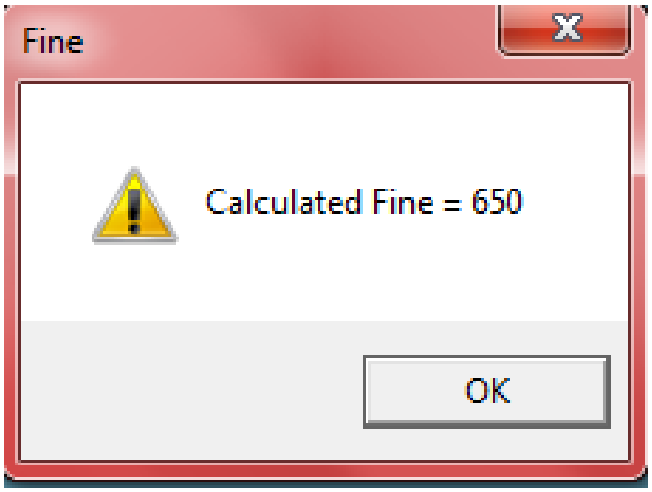
Book Status Form Output 1:



Book Status Form Input 2:



Book Status Form Output 2:



LIMITATIONS

10.LIMITATIOINS:

- We cannot add fields in the database if we require it later. The fields must be specified at the time of designing database.

- If we want to share our Library information with another department of our college then it is not possible because the proposed system is not online.

- The proposed system can only be handled by single user. Because it is desktop base application.

- This system cannot be used at distributed platform it is not a serverside application and cannot use on different client system.

- If student wants to issue two books at a time then it is not possible to issue two book to a single student because only one BT is issued to a student.

CONCLUSION

11. CONCLUSION:

Library Management System is a desktop based application which is used by the Librarian. Any Educational institute can make use of it for providing information about author, contents of the available books in the Library. Everyone needs to go to library once in a day. Our project Library Management System is for monitoring and controlling transactions in the library. We have provided the unique id number known as Accession number by which we can search details of any book easily. The purpose of the library management system is to maintain library records in simple and sophisticated manner. The library management system is very user friendly and time saving. So the library management system makes librarian's work easy.

FUTURE SCOPE

12.FUTURE SCOPE:

- The proposed system can be made distributed to make our transaction easy and sophisticated.
- The list of students who created membership in library can be added in the proposed system.
- This proposed system can be modified to issue two books on a single BT to a student.
- The proposed system can be made online so that we can share our library data with the whole organization.

REFERENCES

13.REFERENCES:

Books:

- Visual Basic 6.0 Programming (Black Book)

  -Steven Holzner

- Mastering Visual Basic 6

  -Evangelos Petroutsos

Websites:

- https://msdn.microsoft.com/en/us/library.y4wf33f0.aspx
- www.vbtutor.net.vbtutor
- https://www.tutorialspoint.com/listtutorials/visual-basic/2