

A
Project Report
On
“SWIMMING POOL MANAGER”

Submitted By

Yeotikar Prashant S.
Kale Shrikant S.
Waghamare Prajwal S.

B.Sc :- III Year

Guided By

Dr. Quazi F. A



YESHWANT MAHAVIDYALAYA, Nanded.

Affiliated to

Swami Ramanad Teerth Marathwada University, Nanded

2019 – 2020

CERTIFICATE

This is to certify that the seminar entitled **“SWIMMING POOL
MANAGER”** has been submitted by Yeotikar Prashant S., Shrikant Kale S.,
Waghamare Prajwal S., under my guidance in partial fulfilment of the
degree of Bachelor of Computer Science course under Swami Ramanand
Teerth Marathwada University, Nanded. During the academic year 2019-
2020.

Dr. Quazi F. A

Leaturer

(Project Guide)

Dr. Pathak P. B.

H.O.D (Comp. Sci.)

ACKNOWLEDGEMENT

We are greatly indebted to our Seminar guide Dr. **Quazi F. A.** for his valuable guidance & enlighten comments throughout the course of this project work. It has been an altogether different experience to work with Name of the guide. We would like to thank him for his helpful suggestions.

I gladly take this opportunity to thanks **Dr. Pathak P. B.** for providing all kind of facilities during progress of project on “**SWIMMING POOL MANAGER**”

Last but not the least, we are also thankful to all those who helps directly or indirectly to develop this project and complete it successfully. Then at the last we are thankful to the department staff, friends & all my well wishers for encouragement.

Name of the Student

Yeotikar Prashant S.

Kale Shrikant S.

Waghamare Prajwal S.

“DECLARATION”

To,
The Principal,
Yeshwant Mahavidyalaya
Nanded.

I hereby declare that Project entitled **“SWIMMING POOL MANAGER”** submitted by me, to the Swami Ramanad Teerth Marathwada University, Nanded is an original work. The contents of Project report, in full or in parts have not been submitted to any other institute or university for the fulfilment of any other degree.

Your faithfully,
Yeotikar Prashant S.
Kale Shrikant S.
Waghamare Prajwal S.

INDEX

Sr. No	Particulars	Page No
1	Introduction	
2	Methodology	
3	S/w and H/w Requirement Specifications	
4	Project Analysis	
5	Project Design	
6	Output Screens	
7	Conclusion	
8	Bibliography and References	

INTRODUCTION

In our project we have computerize the working of “**SWIMMING POOL MANAGER**”. So all the works related to Swimming Pool Manager done easily on a single mouse click with accuracy. Our project contains following modules.

Opening Screen : Under this module we authenticate username and password.

Admin : Under this module we can authenticate personal login.

Membership : Under this module we can maintain customer details.

Billing : Under this module we can maintain customer bills.

Visitor : Under this module we can see information, time table etc.

Exit : Under this module we can immediately back.

Methodology

Front End

Visual Basic (VB) is the third-generation event-driven programming language and integrate development environment (IDE) from Microsoft for its COM programming model. VB is also considered a relatively easy to learn and use programming language, because of its graphical development features and BASIC heritage.

Visual Basic was derived from BASIC and enables the rapid application development (RAD) of graphical user interface (GUI) application, access to databases using data Access Objects, Remote Data Objects, or Active X Data Objects, and creation of ActiveX control and objects. Scripting languages such as VBA and VBScript are syntactically similar to Visual Basic, but perform differently.

A programmer can put together an application using the components provided with Visual Basic itself. Programs written in Visual Basic can also use the Windows API, but doing so requires external function declarations.

The final release was version 6 in 1998. Microsoft's extended support ended in March 2008 and the designated successor was Visual Basic .NET (now known simply as Visual Basic).

Language features

Like the BASIC programming language, Visual Basic was designed to be easily learned and used by beginner programmers. The language not only allows programmers to create simple GUI applications. But can also develop complex applications. Programming in VB is a combination of visually arranging components or controls on a form, specifying attributes and actions of those components, and actions are defined for the components, a simple program can be created without the programmer having to write many lines of code. Performance problems were experienced by earlier versions. But with faster computers and native code compilation this has become less of an issue.

Although programs can be compiled into native code executable from version 5 onwards, They still require the presence of runtime libraries of approximately 1 MB in size. This runtime is included by default in windows 2000 and later, but for earlier versions of windows like 95/98/NT it must be distributed together with the executable.

Forms are created using drag-and-drop techniques. A tool is used to place controls (e.g., text boxes buttons, etc.) on the form (window). Controls have attributes and event handlers associated with them. Default values are provided when the control is created, but may be changed by the programmer. Many attribute values can be modified during run time based on user actions on changes in the environment, providing a dynamic applications. For example, code can be inserted into the form resize event handler to reposition a control so that it remains centered on the form, expands to fill up the form, etc. by inserting code into the event handler for a key press in a text box, the program can automatically translate the case

of the text being entered, or even prevent certain characters from being inserted.

Visual Basic can create executable (EXE files), Active x controls, or DLL files, but is primarily used 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 will automatically display its list and allow the user to select any element. And event handler is called when an item is selected, which can then execute additional code created by the programmer to perform some action based on which element was selected, such as populating a related list.

Alternatively, a Visual Basic component can have no user interface, and instead provide active X objects to other programs via component object model (COM). This allows for server-sides processing or an add-in module.

The language is garbage collected using reference counting, has a large library of utility objects, and has basic object oriented support. Since the more common components are included in the default project template, the programmer seldom need s to specify additional libraries. Unlike many other programming language, Visual Basic is generally not case sensitive, although it will transform keyword into a standard case configuration and force the case of variable names to conform to the case of the entry within the symbol table. String comparisons are case sensitive by default, but can be made case insensitive if so desired.

The Visual Basic compiler is shared with other visual studio languages (C, C++), but restrictions in the IDE do not allow the creation of some targets (Windows model DLLs) and threading models.

Characteristics

Visual Basic has the following traits which differ from C-derived languages:

- Multiple assignments available in C language are not possible. $A = B = C$ does not imply that the values of A, B and C are equal. The Boolean result of “is $B=C$?” is stored in A. result stored in A could therefore be false (0) or true (-1)
- Boolean constant true has numeric value -1. This is because the Boolean data type is stored as a 16-bit signed integer. In this construct -1 evaluates to 16 binary 1s (the Boolean value true), and 0 as 16 0s (the Boolean value false). This is apparent when performing a not operation on a 16 bit signed integer value 0 which will return the integer value - 1, in other words true = Not False. This inherent functionality becomes especially useful when performing logical operations on the individual bits of an integer such as And, Or, Xor and Not . This definition of True is also consistent with BASIC since the early 1970s Microsoft BASIC implementation and is also related to the characteristics of CPU instructions at the time.

- Logical and bitwise operators are unified. This is unlike some C-derived languages (such as C), which have separate logical and bitwise operators. This again is a traditional feature of BASIC.
- Variable array base. Arrays are declared by specifying the upper and lower bounds in a way similar to Pascal and Fortran. It is also possible to use the `OPTION BASE` statement to set the default lower bound. Use of the `OPTION BASE` statement can lead to confusion when reading Visual Basic code and is best avoided by always explicitly specifying the lower bound of the array. This lower bound is not limited to 0 or 1, because it can also be set by declaration. In this way, both the lower and upper bounds are programmable. In more subscript-limited languages, the lower bound of the array is not variable. This uncommon trait does exist in Visual Basic .NET but not in VBScript.

`OPTION BASE` was introduced by ANSI, with the standard for ANSI minimal BASIC in the late 1970s.

- Relatively strong integration with the Windows operating system and the component object model. The native types for strings and arrays are the dedicated COM types, `BSTR` and `SAFEARRAY`.
- Banker's rounding as the default behaviour when converting real numbers to integers with the `Round` function. `> Round(2.5, 0)` gives 2, `? Round(3.5, 0)` gives 4.
- Integers are automatically promoted to `Double` in expressions involving the normal division operator (`/`) so that division of an odd integer by an even integer by an even integer produces the intuitively correct

result. There is a specific integer divide operator (/) which does truncates.

- By default, if a variable has not been declared or if no type declaration character is specified, the variable is of type variant. However this can be changed with Deftype statements such as DefInt, DefBool, DefVar, DefObj and DefStr. There are 12 Deftype statements in total offered by Visual Basic 6.0. The default type may be overridden for a specific declaration by using a special suffix character on the variable name (# for Double ! for single, & for Long, %for Integer, \$ for String, and @ for Currency) or using the key phrase As (type). VB can also be set in a mode that only explicitly declared variables can be used with the command option Explicit.

VB 1.0 was introduced in 1991. The drag and drop design for creating the user interface is derived from a prototype form generator developed by Alan cooper and his company called Tripod. Microsoft contracted with cooper and his associates to develop Tripod into a programmable form system for windows 3.0, under the code name Ruby (no relation to the Ruby programming language).

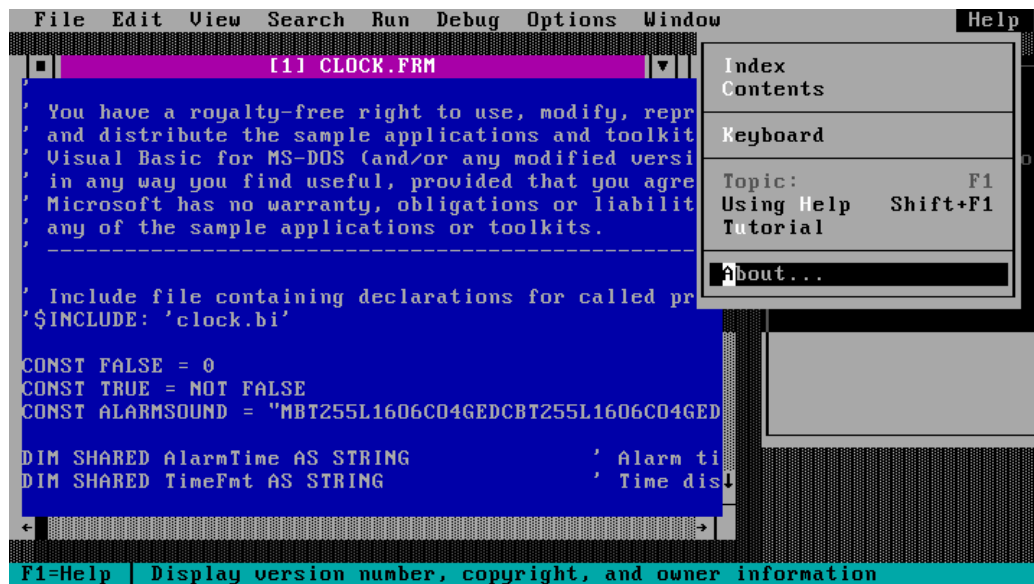
Tripod did not include a programming language at all. Microsoft decided to combine Ruby with the Basic language to create Visual Basic.

The Ruby interface generator provided the “Visual” part of Visual Basic and this was combined with the “EB” Embedded BASIC engine designed for

Microsoft's abandoned "Omega" database system. Ruby also provided the ability to load dynamic link libraries containing additional controls (the called "gizmos"), Which later became the VNX interface.

Timeline

- Project 'thunder' was initiated
- Visual Basic 1.0 (May 1991) was released for Windows at the Comdex/Windows World trade show in Atlanta, Georgia.
- Visual Basic 1.0 for DOS was released in September 1992. The language itself was not quite compatible with Visual Basic for Windows, as it was actually the next version of Microsoft's DOS-based BASIC compilers, QuickBasic and BASIC Professional Development system. The interface used a Text user interface, using extended ASCII characters to simulate the appearance of a CUI.



Visual Basic for MS-DOS

- Visual Basic 2.0 was released in November 1992. The programming environment was easier to use, and its speed was improved. Notably, forms became instatable objects, thus laying the foundational concepts of class modules as were later offered in VB4.
- Visual Basic 3.0 was released in the summer of 1993 and came in Standard and Professional versions. VB3 included version 1.1 of the Microsoft Jet Database Engine that could read and write Jet (or Access) 1.x databases.
- Visual Basic 4.0 (August 1995) was the first version that could create 32-bit as well as 16-bit Windows programs. It also introduced the ability to write non-GUI classes in Visual Basic. Incompatibilities

between different releases of VB4 caused installation and operation problems. While previous versions of Visual Basic had used VBX controls, Visual Basic now used OLE controls (with file names ending in .OCX) instead. These were later to be named Active X controls.

- With version 5.0 (February 1997), Microsoft released Visual Basic exclusively for 32-bit versions of Windows. Programmers who preferred to write 16-bit programs were able to import programs written in Visual Basic 4.0 to Visual Basic 5.0, and Visual Basic 5.0 programs can easily be converted with Visual Basic 4.0. Visual Basic 5.0 also introduced the ability to create custom user controls, as well as the ability to compile to native Windows executable code, speeding up calculation-intensive code execution. A free, downloadable Control creation Edition was also released for creation of ActiveX controls. It was also used as an introductory form of Visual Basic a regular .exe project could be created and run in the IDE, but not compiled.
- Visual Basic 6.0 (Mid 1998) improved in a number of areas including the ability to create web-based applications. VB6 has entered Microsoft's "non-supported phase" as of March 2008. Although the Visual Basic 6.0 development environment is no longer supported, the runtime is supported on Windows Vista, Windows Server 2008 and Windows 7
- Mainstream Support for Microsoft Visual Basic 6.0 ended on March 31, 2005. Extended support ended in March 2008. In response, the Visual Basic user community expressed its grave concern and lobbied

users to sign a petition to keep the product alive. Microsoft has so far refused to change their position on the matter. (but see) Ironically, around this time (2005), it was exposed that Microsoft's new anti-spyware offering, Microsoft Anti Spyware (part of the GIANT Company Software purchase), was coded in Visual Basic 6.0 Its replacement, Windows Defender, was rewritten as C++ code.

Derivative languages

Microsoft has developed derivatives of Visual Basic for use in scripting. Visual Basic itself is derived heavily from BASIC, and subsequently has been replaced with a .NET platform version.

Some of the derived languages are:

- Visual Basic for Applications (VBA) is included in many Microsoft applications (Microsoft Office) and also in many third-party products like Solid Works, AutoCAD. WordPerfect Office 2002. ArcGIS, Sage Accpac ERP, and Business Objects Desktop Intelligence. There are small inconsistencies in the way VBA is implemented in different applications, but it is largely the same language as VB6 and uses the same runtime library.
- VBScript is the default language for Active Server Pages. It can be used in Windows scripting and client-side web page scripting.

Although it resembles VB in syntax, it is a separate language and it is executed by vbscript.dll as opposed to the VB runtime, ASP and VBScript should not be confused with ASP.NET which uses the .NET Framework for compiled web pages.

- Visual Basic .NET is Microsoft's designated successor to Visual Basic 6.0, and is part of Microsoft's .NET platform. Visual Basic.Net compiles and runs using the NET Framework. It is not backwards compatible with VB6. An automated conversion tool exists, but fully automated conversion for most projects is impossible.
- Star Office Basic is a Visual Basic compatible interpreter included in Star Office suite, developed by Sun Microsystems.
- Gambas are a Visual Basic inspired free software programming language. It is not a clone of Visual Basic, but it does have the ability to convert Visual Basic programs to Gambas.

Performance and other issues

Earlier counterparts of Visual Basic (prior to version 5) compiled the code to P-Code or Pseudo code only. Visual Basic 5 and 6 are able to compile the code to either native or P-Code as the programmer chooses. The P-Code is interpreted by the language runtime, also known as virtual machine, implemented for benefits such as portability and small code. However, it usually slows down the execution by adding an additional layer of interpretation of code by the runtime although small amounts of code and algorithms can be constructed to run faster than the compiled native code. Visual Basic applications require Microsoft Visual Basic runtime MSVBVMxx.DLL, where xx is the relevant version number, either 50 or 60. MSVBVM60.dll comes as standard with Windows in all editions after Windows 98 while MSVBVM50.dll comes with all editions after Windows 95. A Windows 95 machine would however require inclusion with the installer of whichever dll was needed by the program.

Criticisms levelled at Visual Basic editions prior to VB.NET include:
Versioning problems associated with various runtime DLLs, known as DLL hell

- Poor support for object-oriented programming Inability to create multi-threaded applications, without resorting to Windows API calls
- Inability to create Windows services
- Variant types have a greater performance and storage overhead than strongly typed programming languages

- Dependency on complex and fragile COM Registr
- y entries The development environment is no longer supported by Microsoft.

Legacy development and support

All versions of the Visual Basic development environment from 1.0 to 6.0 have been retired and are now unsupported by Microsoft. The associated runtime environments are unsupported too, with the exception of the Visual Basic 6 core runtime environment, which will be officially supported by Microsoft for the lifetime of Windows 7. Third party components that shipped with Visual Studio 6.0 are not included in this support statement. Some legacy Visual Basic components may still work on newer platforms, despite being unsupported by Microsoft and other vendors.

Development and maintenance development for Visual Basic 6 is possible on legacy Windows XP, Windows Vista and Windows 2003 using Visual Studio 6.0 platforms, but is unsupported. Documentation for Visual Basic 6.0, its application programming interface and tools is best covered in the last MSDN release before Visual Studio.NET 2002. Later releases of MSDN focused on .NET development and had significant parts of the Visual Basic 6.0 programming documentation removed. The Visual Basic IDE can be installed and used on Windows Vista, where it exhibits some minor incompatibilities which do not hinder normal software development and maintenance. As of August 2008, both Visual Studio 6.0 and the MSDN documentation mentioned above are available for download by MSDN subscribers.

Example code

Here is an example of the language: Code snippet that displays a message box "Hello, World!" as the window *Form* loads:

```
Private Sub Form Load ()  
    'execute a simple message box that will say "Hello, World!"  
    MsgBox "Hello, World!"  
End Sub
```

Another good example for a Serial Key program.

If Textbox1.text = "Your serial key here" then

Msgbox"Correct Key"

Form2.show

else

Msgbox"InCorrect Key"

me.close

You can change the Msgbox Message if u want.

Back End

Microsoft Office Access, previously known as Microsoft Access, is a 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 Microsoft Office suite of applications and is included in the Professional and higher versions for Windows and also sold separately. There is no version for MacOS or for Microsoft Office Mobile.

Access stores data in its own format based on the Access Jet Database Engine. It can also import or link directly to data stored in other Access databases, Excel,

Share Point list, text, XML, Outlook, HTML, dBase, Paradox, Lotus 1-2-3, or any ODBC- compliant data container including Microsoft SQL Server, Oracle, MySQL and PostgreSQL. Software developers and data architects can use it to develop application software and non- programmer "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.

Microsoft Access is part of the Microsoft Office suite and is the most popular Windows desktop database application. It is targeted for the information worker market, and is the natural progression for managing

data when the need for a relational database arises or after reaching the limits of Microsoft Excel.

Microsoft Access is used by programmers and non-programmers to create their own database solutions. Access tables support a variety of standard field types, indices, and referential integrity. Access also includes a very intuitive query interface, forms to display and enter data, and reports for printing. The underlying Jet database which contains these objects is multiuser aware and handles record locking and referential integrity including cascading updates and deletes.

Microsoft Access is known for its ability to empower non-professional developers to create database applications on their own. Users can easily create tables, queries, forms and reports, and connect them together with macros. Advanced users can use VBA to write rich solutions with advanced data manipulation and user control.

The original concept of Access was for end users to be able to "access" data from any source. Access offers the ability to import and export data to many formats including Excel, Outlook, ASCII, dBase, Paradox, FoxPro, SQL Server, Oracle, ODBC, etc. It also has the ability to link to data in its existing location and use it for viewing, querying, editing, and reporting. This allows the existing data to change and the Access platform to always use the latest data. It can even perform heterogeneous joins between data sets stored across different platforms. Access is often used by people downloading data from enterprise level databases for manipulation, analysis, and reporting locally.

S/w and H/w Requirement Specification

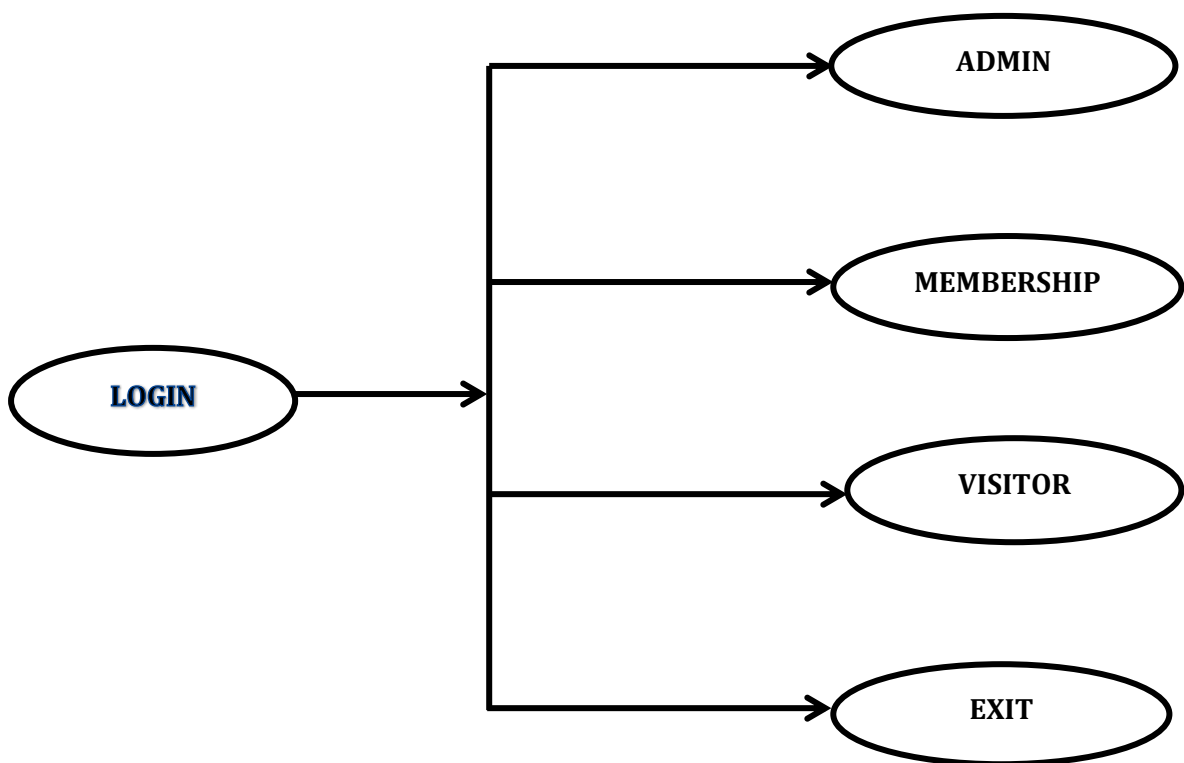
SOFTWARE REQUIREMENT:

- WINDOWS 10 Pro
- VISUAL BASIC 6.0
- MICTOSFT ACCESS

HARDWARE REQUIREMENT:

- INTEL® CORE™ i5, 3.20 GHZ
- 4 GB RAM
- LAN CARD
- 1 TB MIN HDD SPACE

Project Analysis
DATA FLOW DIAGRAM
MDI FORM



CODING

CODING

USERNAME AND PASSWORD FORM CODE:

```
Private Sub Command1_Click()
```

```
If Trim(Text1) = "TONY" And Trim(Text2) = "STARK" Then
```

```
Unload Me
```

```
Form8.Show
```

```
Else
```

```
MsgBox "WRONG USER NAME OR PASSWORD"
```

```
End If
```

```
End Sub
```

```
Private Sub Command2_Click()
```

```
Text1.Text = ""
```

```
Text2.Text = ""
```

```
End Sub
```

```
Private Sub Image1_Click()
```

```
End Sub
```

```
Private Sub VI_Click()
```

```
Form2.Show
```

Load Me

End Sub

VISITOR FORM CODE:

```
Private Sub Command1_Click()
```

```
Form3.Show
```

```
End Sub
```

```
Private Sub Command2_Click()
```

```
Form4.Show
```

```
End Sub
```

```
Private Sub Command3_Click()
```

```
Form5.Show
```

```
End Sub
```

```
Private Sub Command4_Click()
```

```
Form6.Show
```

```
End Sub
```

```
Private Sub Command5_Click()
```

```
Form7.Show
```

```
End Sub
```

```
Private Sub Command6_Click()
```

```
Form1.Show
```

```
Unload Me
```

```
End Sub
```

MANAGER WORK FORM CODE:

```
Private Sub Command1_Click()
```

```
Form9.Show
```

```
End Sub
```

```
Private Sub Command2_Click()
```

```
Form10.Show
```

```
End Sub
```

```
Private Sub Command3_Click()
```

```
PrintForm
```

```
End Sub
```

```
Private Sub EXIT_Click()
```

```
Form8.Show
```

```
Unload Me
```

```
End Sub
```

```
Private Sub Image1_Click()
```

End Sub

Private Sub NEW_Click()

Form9.Show

End Sub

Private Sub OPEN_Click()

Form10.Show

End Sub

Private Sub PRINTBILL_Click()

PrintForm

End Sub

Private Sub SAVEBILL_Click()

PrintForm

End Sub

CUSTOMER DETAILS FORM CODE:

```
Dim db As Database
```

```
Dim rs As Recordset
```

```
Private Sub Command1_Click()
```

```
Set db = OpenDatabase("D:\Visual Basic6.0\PROJECT\Swiming pool\SWIMMING POOL.mdb")
```

```
Set rs = db.OpenRecordset("SELECT * from CUSTOMER")
```

```
rs.AddNew
```

```
rs.Fields(0).Value = CInt(Text1.Text)
```

```
rs.Fields(1).Value = (Text2.Text)
```

```
rs.Fields(2).Value = CDbl(Text3.Text)
```

```
rs.Fields(3).Value = (Text4.Text)
```

```
rs.Fields(4).Value = (Text5.Text)
```

```
rs.Fields(5).Value = (Text6.Text)
```

```
If Option1.Value = True Then
```

```
rs.Fields(6).Value = "Male"
```

```
End If
```

```
If Option2.Value = True Then
```

```
rs.Fields(6).Value = "Female"
```

```
End If
```

```
rs.Fields(7).Value = (Text7.Text)
```

```
rs.Fields(8).Value = (Text8.Text)
```

```
rs.Fields(9).Value = CDbl(Text9.Text)
```

```
MsgBox ("record saved")
```

```
rs.Update
```

```
End Sub
```

```
Private Sub Command2_Click()
```

```
Unload Me
```

```
Form8.Show
```

```
End Sub
```

```
Private Sub Form_Load()
```

```
Text1.Text = " "
```

```
Text2.Text = " "
```

```
Text3.Text = " "
```

```
Text4.Text = " "
```

```
Text5.Text = " "
```

```
Text6.Text = " "
```

```
Text7.Text = " "
```

```
Text8.Text = " "
```

```
Text9.Text = " "
```

```
End Sub
```

BILLING FORM CODE:

Private Sub Command1_Click()

Form9.Show

End Sub

Private Sub Command2_Click()

Form10.Show

End Sub

Private Sub Command3_Click()

PrintForm

End Sub

Private Sub EXIT_Click()

Form8.Show

Unload Me

End Sub

Private Sub Image1_Click()

End Sub

Sub SHOWBILL()

Text1.Visible = False

Text2.Visible = False

Text3.Visible = False

Text4.Visible = False

Labe1.Visible = False

Labe2.Visible = False

Labe3.Visible = False

Labe4.Visible = False

SALES.BackColor = vbWhite

SALES.ForeColor = vbBlack

Cls

SALES.FontSize = 10

Print "BATCH NO.", Text1

Print "NAME", Text2

Print "EXPIRY DATE", Text3

Dim T

T = 0

Print "-----"
-----"

Print "MEMBERS", vbTab, vbTab, vbTab, "MONTHS", vbTab, vbTab, "AMOUNT"

Print "-----"
-----"

For A = 0 To C - 1

```

Print MONTHS(A, 0), vbTab, MONTHS(A, 1), vbTab, MONTHS(A, 2), vbTab, MONTHS(A, 3), vbTab

T = T + MONTHS(A, 3)

Next A

Print "-----"
-----"

Print "TOTAL BILL ", vbTab, vbTab, vbTab, vbTab, vbTab, vbTab, T

Print "-----"
-----"


End Sub

Private Sub NEW_Click()

Form9.Show


End Sub


Private Sub OPEN_Click()

Form10.Show


End Sub


Private Sub PRINTBILL_Click()

PrintForm


End Sub


Private Sub SAVEBILL_Click()

PrintForm

```

End Sub

Private Sub Command1_Click()

PrintForm

End Sub

Private Sub Command2_Click()

Unload Me

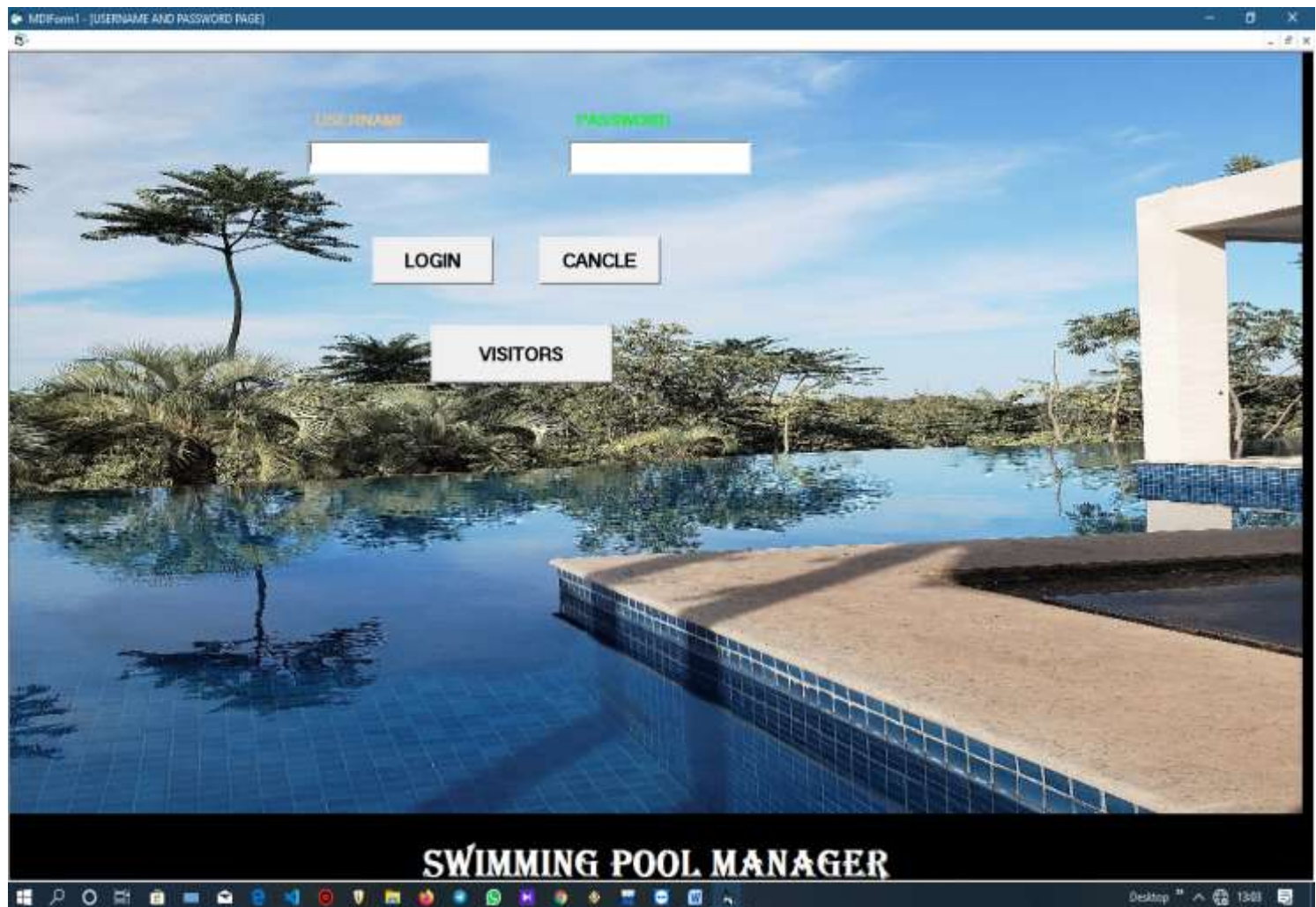
Form8.Show

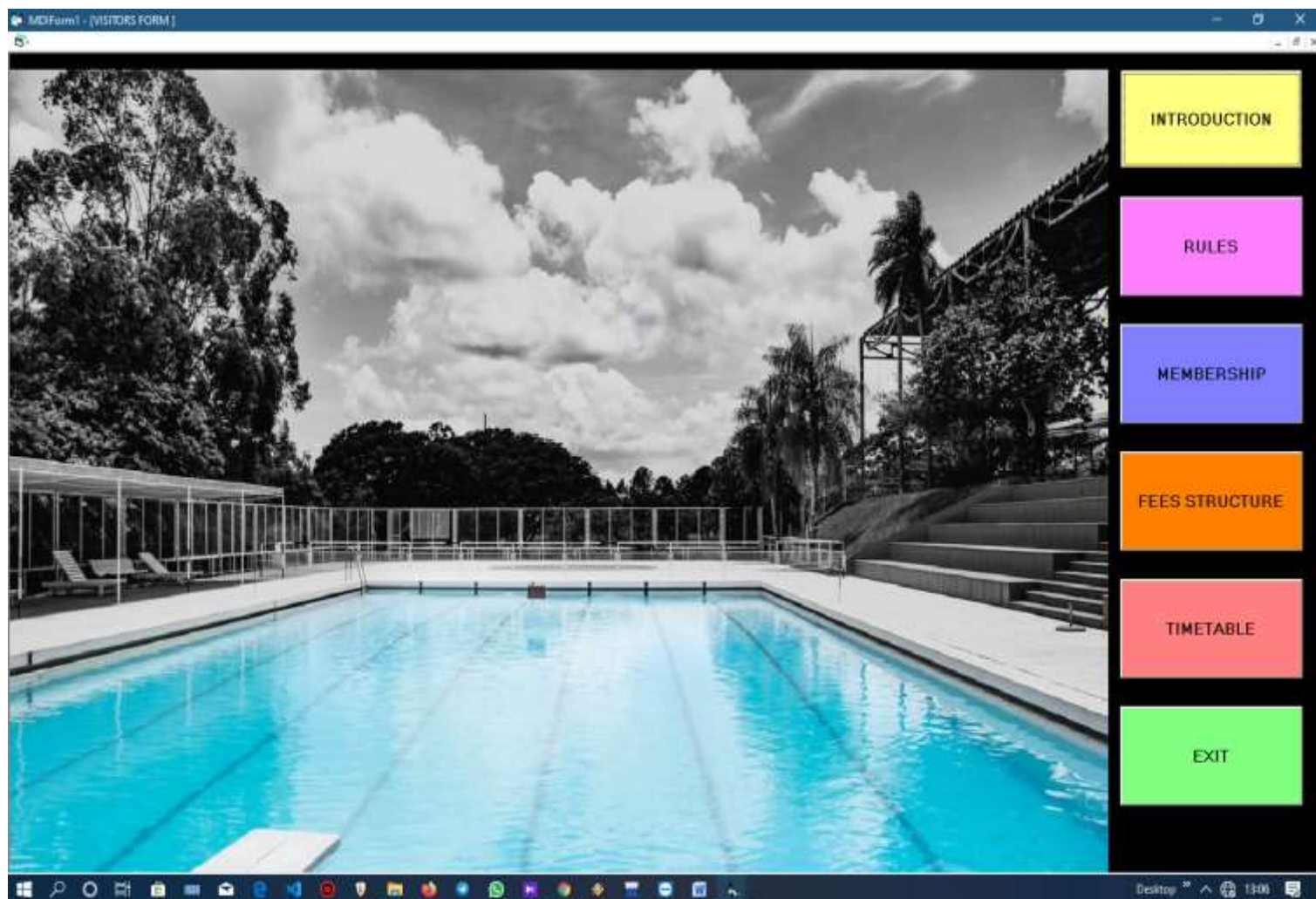
End Sub

Private Sub Form_Load()

End Sub

OUTPUT







MDForm1 - [MEMBERSHIP]

# MEMBERSHIP #	
Adults (Above 18 Years)	Life Time Membership Pass - 20,000/-
	Per Year Membership Pass - 3,000/-
	Six Month Membership Pass - 1,500/-
Kids (Below 18 Years)	Life Time Membership Pass - 15,000/-
	Per Year Membership Pass - 2,500/-
	Six Month Membership Pass - 1,200/-

Desktop 11:56

MDFormF - [FEES STRUCTURE]

FEES STRUCTURE

FOR ADULTS (Above 18 Years)

- > First Month - 600/-
 - Registration - 300/-
 - Coach - 300/-
- > Next Month - 300/-

FOR KIDS (Above 18 Years)

- > First Month - 400/-
 - Registration - 200/-
 - Coach - 200/-
- > Next Month - 250/-

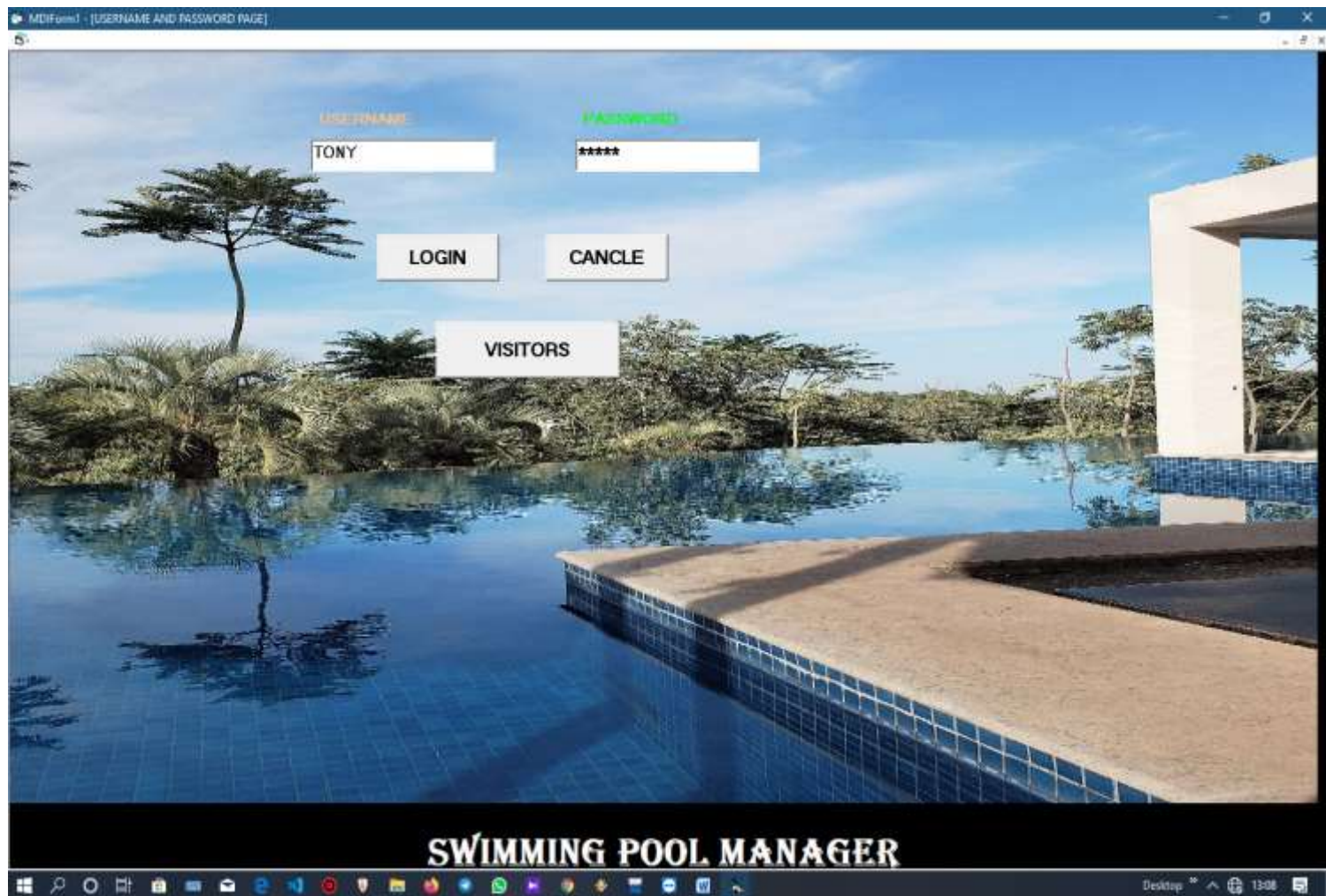


Desktop 11:56

SWIMMING POOL TIME TABLE

Batch No.	Time To Enter	Batch Timeout
***** <u>MORNING SESSION</u> *****		
1	6:00 To 6:15 (Then close the gate)	7:00
2	7:00 To 7:15 (Then close the gate)	8:00
3	8:00 To 8:15 (Then close the gate)	9:00
4	9:00 To 9:15 (Then close the gate)	10:00
Swimming Pools Will Be Closed For 10 To 4 Swims		
***** <u>EVENING SESSION</u> *****		
5	4:00 To 4:15 (Then close the gate)	5:00
6	5:00 To 5:15 (Then close the gate)	6:00

Batch No.	Time To Enter	Batch Timeout
***** <u>EVENING SESSION</u> *****		
7	6:00 To 6:15 (Then close the gate)	7:00



MDForm1 - [MANAGER WORK FORM]

MEMBERSHIP BILLING EXIT

BATCH NUMBER

NAME

EXPIRY DATE

FEES

MDForm1 - [Form0]

CUSTOMER DETAILS

BATCH NUMBER

FULL NAME

PHONE NUMBER

ADDRESS

EMAIL ID

AGE GENDER: ☐ MALE ☐ FEMALE

JOINING DATE EXP DATE

FEES

SAVE CANCEL

MDForm1 - [LIST OF DATABASE]

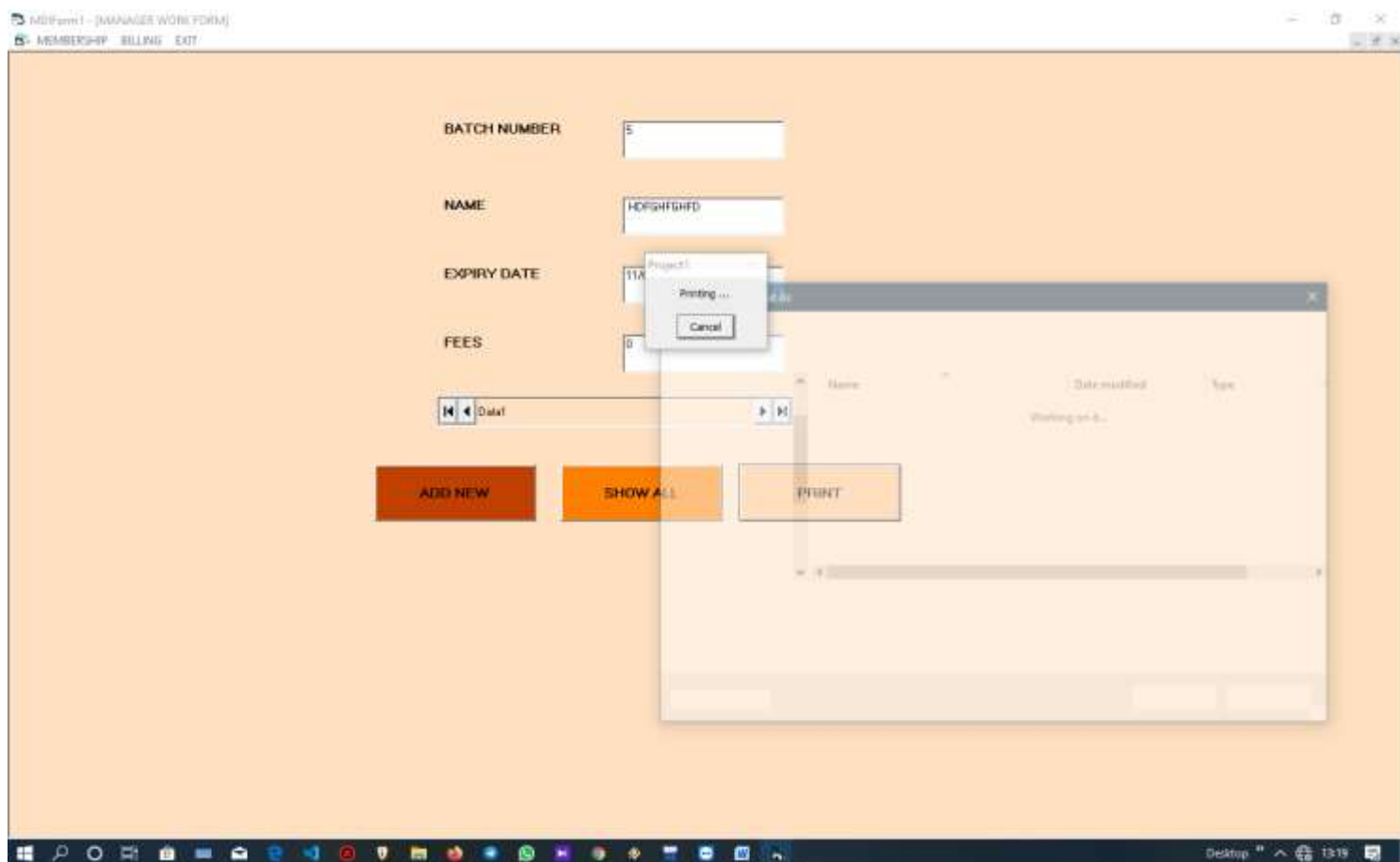
LIST OF DATABASE

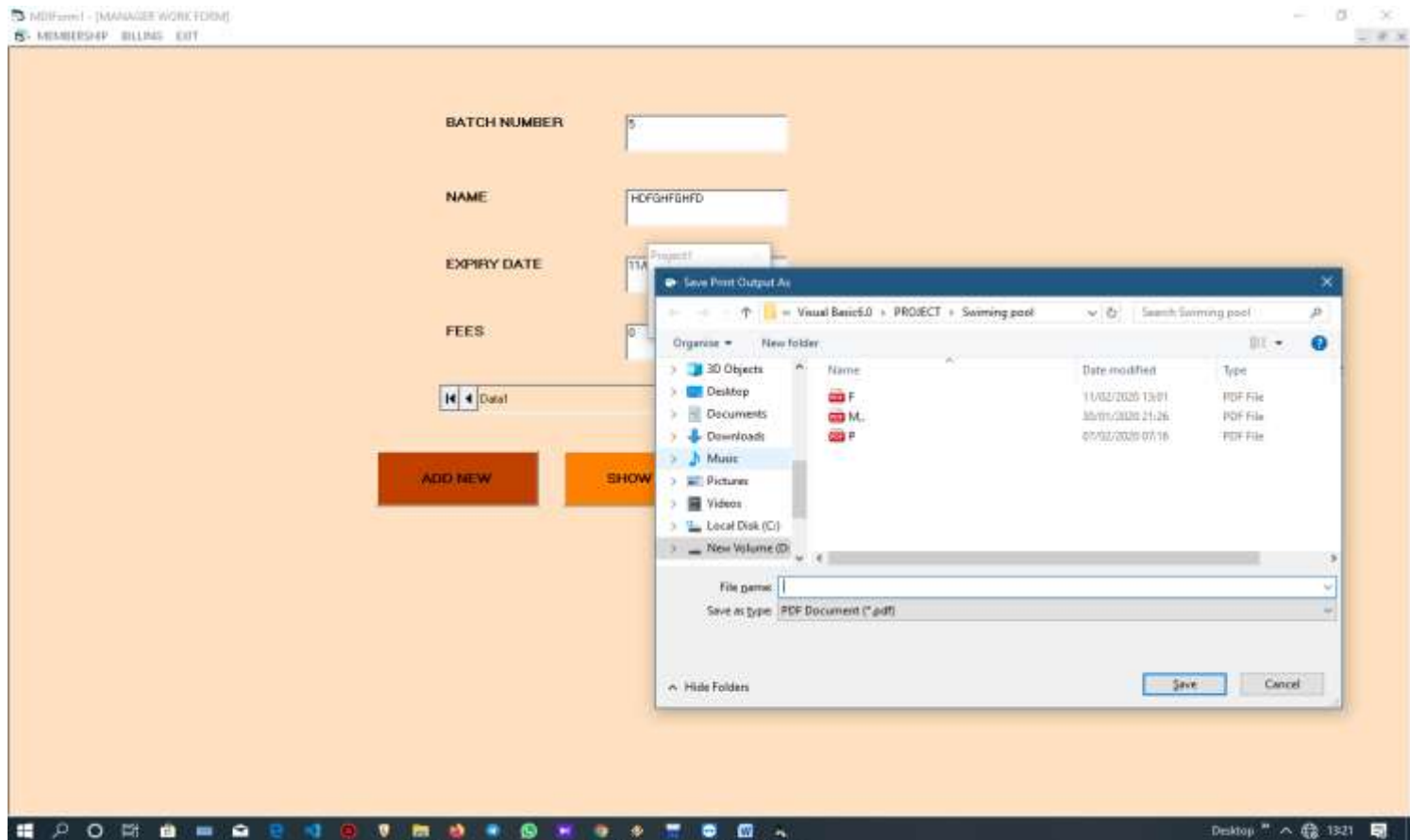
BATCHNUMBER	FULLNAME	PHONENUMBER	ADDRESS	EMAILID	AGE	GENDER	JOINNGDATE	EXPDATE	FEES
1	ADFGHFGHFD	76354634563	KUFJHFGHFGH	FGHFGHFGHFGH	45	Male	12/02/2020	11/02/2020	0
2	ASHUTOSH PATIL	955454566512	BABA NAGAR NANDE	ashutoshpatil@gmail.co	21	Male	12/02/2020	13/03/2020	5222
1	PRASHANT	9552012892	MANTRI NAGAR NANDE	pr00000099@gmail.co	21		12/01/2020	12/02/2020	20000
2	SHIVRAJ	9552012892	BABA NAGAR NANDE	PY000000@gmail.com	47	Male	08/02/2020	08/03/2020	600

1/4

PRINT

EXIT





CONCLUSION

- With the help of this software we have reduced the paper work which will reduce the efforts of storing data of swimming pool Customers.
- This is also helpful for peoples, specially the "visitors" section, which is capable of resolving various FAQs of so many visitors within one interface, such as instructions regarding the pool and fees structure and all other necessary answers are present here in one click.
- Also the visitors section will aware customers about exiting offers and deals for purchasing vacation courses and long term membership of the pool.
- This software is capable of storing customer records for too many days for their convenience.
- The authentication section could only been accessed by the chairperson or the manager of the swimming pool, Which will be much safer to take care of exact period of validity of customers.

So many goals that are covered in this software and yet there are so many functionalities we are trying to put, in our next increments.

Book Referred (Bibliography)

Book Name : - Programming with Visual Basic

Author : - Julia case Bradley, Anita C Millspough

Publication : - Tata Mc-Graw Hill Edition