# BANK MANAGEMENT SYSTEM
## A PROJECT REPORT

**Submitted by:-**
**Tanya Goel**
University Roll no – 1900290149101

**Submitted in partial fulfilment of the**

**Requirements for the Degree of**

MASTER OF COMPUTER APPLICATIONS

**Under the Supervision of**
**TATA CONSULTANCY SERVICES**

**Submitted to**

**DEPARTMENT OF COMPUTER APPLICATIONS**
**KIET Group of Institutions,**
**Ghaziabad**
**Uttar Pradesh-201206**
**(JUNE 2021)**

# DECLARATION

I hereby declare that the work presented in this report entitled “Bank Management System", was carried out by me. I have not submitted the matter embodied in this report for the award of any other degree or diploma of any other University or Institute.

I have given due credit to the original authors/sources for all the words, ideas, diagrams, graphics, computer programs, experiments, results, that are not my original contribution. I have used quotation marks to identify verbatim sentences and given credit to the original authors/sources.

I affirm that no portion of my work is plagiarized, and the experiments and results reported in the report are not manipulated. In the event of a complaint of plagiarism and the manipulation of the experiments and results, I shall be fully responsible and answerable.

Name : Tanya Goel

Roll No. 1900290149101

## CERTIFICATE

Certified that **Tanya Goel (**Roll no 1900290149101) has carried out the research work presented in this report entitled **"Bank Management System"** for the award of **Masters of Computer Application** from Dr. APJ Abdul Kalam Technical University, Lucknow under my supervision. This report embodies results of original work, and studies are carried out by the student herself and the contents of the report do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution

**Date:**

**Tanya Goel**
**University Roll No.**
**1900290149101**

**Date:**

**Department of Computer Applications**
**KIET GROUP OF INSTITUTIONS,**
**GHAZIABAD**

**Signature of Internal Examiner** **Signature of External Examiner**

# ABSTRACT

The Bank Account Management System is an application for maintaining a person's account in a bank. In this project I tried to show the working of a banking account system and cover the basic functionality of a Bank Account Management System. To develop a project for solving financial applications of a customer in banking environment in order to nurture the needs of an end banking user by providing various ways to perform banking tasks. Also to enable the user's work space to have additional functionalities which are not provided under a conventional banking project. The Bank Account Management System undertaken as a project is based on relevant technologies. The main aim of this project is to develop software for Bank Account Management System. This project has been developed to carry out the processes easily and quickly, which is not possible with the manuals systems, which are overcome by this software. This project is developed using PHP, HTML language and MYSQL use for database connection. Creating and managing requirements is a challenge of IT, systems and product development projects or indeed for any activity where you have to manage a contractual relationship. Organization need to effectively define and manage requirements to ensure they are meeting needs of the customer, while proving compliance and staying on the schedule and within budget. The impact of a poorly expressed requirement can bring a business out of compliance or even cause injury or death. Requirements definition and management is an activity that can deliver a high, fast return on investment. The project analyzes the system requirements and then comes up with the requirements specifications. It studies other related systems and then come up with system specifications. The system is then designed in accordance with specifications to satisfy the requirements. The system design is then implemented with MYSQL, PHP and HTML. The system is designed as an interactive and content management system. The content management system deals with data entry, validation confirm and updating whiles the interactive system deals with system interaction with the administration and users. Thus, above features of this project will save transaction time and therefore increase the efficiency of the system.

# ACKNOWLEDGEMENTS

Success in life is never attained single handedly. My deepest gratitude goes to **TATA CONSULTANCY SERVICES** for guidance, help and encouragement throughout my research work. Their enlightening ideas, comments, and suggestions.

Words are not enough to express my gratitude to **Dr. Ajay Kumar Shrivastava, Professor and Head, Department of Computer Applications**, for his insightful comments and administrative help at various occasions.

Fortunately, I have many understanding friends, who have helped me a lot on many critical conditions.

Finally, my sincere thanks go to my family members and all those who have directly and indirectly provided me moral support and other kind of help. Without their support, completion of this work would not have been possible in time. They keep my life filled with enjoyment and happiness.

**TANYA GOEL**

TABLE OF CONTENTS PAGE NO

# INTRODUCTION

During the past several decades personnel function has been transformed from a relatively obscure record keeping staff to central and top level management function.There are many factors that have influenced this transformation like technological advances, professionalism, and general recognition of human beings as most important resources.

A computer based management system is designed to handle all the primary information required to calculate monthly statements of customer account which include monthly statement of any month. Separate database is maintained to handle all the details required for the correct statement calculation and generation.

This project intends to introduce more user friendliness in the various activities such as record updation, maintenance, and searching. The searching of record has been made quite simple as all the details of the customer can be obtained by simply keying in the identification or account number of that customer. Similarly, record maintenance and updation can also be accomplished by using the account number with all the details being automatically generated. These details are also being promptly automatically updated in the master file thus keeping the record absolutely up-to-date.

The entire information has maintained in the database or Files and whoever wants to retrieve can’t retrieve, only authorization user can retrieve the necessary information which can be easily be accessible from the file.

# OBJECTIVE OF THE PROJECT

A computer based management system is designed to handle all the primary information required to calculate monthly statements of customer account which include monthly statement of any month. Separate database is maintained to handle all the details required for the correct statement calculation and generation.

This project intends to introduce more user friendliness in the various activities such as record updation, maintenance, and searching. The searching of record has been made quite simple as all the details of the customer can be obtained by simply keying in the identification or account number of that customer. Similarly, record maintenance and updation can also be accomplished by using the account number with all the details being automatically generated. These details are also being promptly automatically updated in the master file thus keeping the record absolutely up-to-date.

The main objective of our project is providing the different typed of customers facility, the main objective of this system is to find out the actual customer service. Etc.

- It should fulfill almost all the process requirements of any Bank.
- It should increase the productivity of bank by utilizing the working hours more and more, with minimum manpower.

This project includes the entire upgraded feature required for the computerization banking system. This system is very easy to use, so that any user can use without getting pre-knowledge about this. Its very much user friendly and meet almost all daily working process requirements. This system is completely GUI based and can be use by mouse and as well as keyboard. This system is melded in such a way that has got all features to upgrade without making much change in existing components.

# FEASIBILITY ANALYSIS:-

Depending on the results of the initial investigation, the survey is expanded to a more detailed feasibility study. A feasibility study is a test of a system proposal. According to its workability, impact on the organization, ability to meet user's needs and effective use of the resources its main task done during the feasibility study are:-

1. Evaluation of existing system and procedures. Our group went to various Banking Professionals to gather information about the software system. They are using and evaluating those system and the procedures invoked in it during the period of feasibility study.

2. Analysis of alternative candidate systems after studying the various systems we derived various alternatives through which we develop our project and evaluated the alternative. The most appropriate is selected.

**FEASIBILITY STUDY**

The only tangible benefit provided by the proposed system is that the paper work is reduced to the minimum and hence the reduction in cost incurred on Stationary and its storage. The system provides many benefits that can't be measured in terms of Money for e.g. user's friendliness, more user response being more efficient.

- ✓ TECHNICAL FEASIBILITY:-

  The proposed system is technically feasible as it can be developed easily with the help of available technology. The proposed system requires MS – VISUAL Studio 2005 using VB.Net as a Interface for Programming & back-end as MS-SQL Server 2000 for storing/maintaining database. The database can be easily interconnected using MS-SQL Server 2000.

- ✓ OPERATIONAL FEASIBILITY:-

Automation makes our life easy. The proposed system is highly user friendly and is much easily able to interact with the system. Therefore the users will readily accept the system as data entry and making queries can be easily done.

# SYSYTEM REQUIREMENTS

## Hardware specifications

Hardware is a set of physical components, which performs the functions of applying appropriate, predefined instructions. In other words, one can say that electronic and mechanical parts of computer constitute hardware.

This package is designed on a powerful programming language Visual Basic. It is a powerful Graphical User Interface. The backend is ACCESS, which is used to maintain database. It can run on almost all the popular microcomputers. The following are the minimum hardware specifications to run this package: -

Personal Computer: -

It minimum contains P-III

Processor with 128 MB RAM

## Software Requirements**:**

The software is a set of procedures of coded information or a program which when fed into the computer hardware, enables the computer to perform the various tasks. Software is like a current inside the wire, which cannot be seen but its effect can be felt.

1. Operating System:- Windows NT / 2000 / XP
2. Application Software:- Application software uses front end visual basic and database access etc.

Editor:- Visual basic.

# SOFTWARE DEVELOPMENT LIFE CYCLE

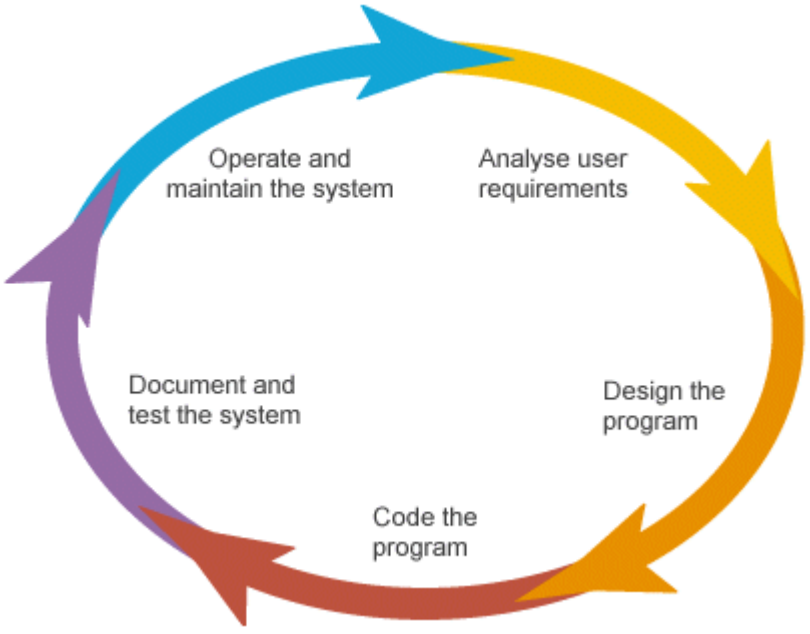
A system development life cycle is a logical process by which system analysts, software engineers, programmers, and end users build information systems and computer applications to solve business problems and needs.

The major phases involved in the MIS development process are referred to as system development life cycle. Each phase of the development process must have well defined objectives ,and at the end of each phase ,progress towards meeting the objectives must be evaluated.

The development process should not continue until the objectives of all prior phases have been met.

System development life cycle is a phased approach to analysis and design to ensure that systems are best developed.

The system development life cycle can be divided into seven phases as shown in fig



# INTRODUCTION TO FRONT END TOOL

- Visual programming aims at providing the user with an interface that is intuitive and easy to use. In developing such an interface, the programmer employs user-friendly features such as windows, menus, buttons and list boxes.

- Its Environment provides all features that are required to develop a graphical user interface as ready -to- use components. The programmer does not have to write code to create and display commonly required user-friendly features each time around.

- When the programmer needs a specific user interface feature such as button, he selects the appropriate ready-to-use component provided by the visual programming environment. These components can be moved, resized and renamed as required.

- **For Example:-**

  If the programmer needs to have a button then the visual programming environment provides him with one. All that, the programmer does this selec t the button and place it on screen at the required position.

- Typically the mouse is used to select and place the necessary components. Thus, the visual programming environment is also called a point and click environment.

- A visual programming environment automates the process of creating a user interface. The interface provided by the visual programming environment to the programmer designs the user interface visually instead of writing code.

- In addition it also provides a means of associating code with each component. In each case of calculator, for each button, we can specific that the code is to execute when we click on it.

## NEED FOR VISUAL PROGRAMMING:-

- There are several programming tools that allow us to build such visually appealing and intuitive interface. These tools allow us to design interface that employ user friendly features such as menus, buttons, windows etc.

- However, the disadvantage of such tools is that the interface is designed using code. The programmer has to code the user interface features specifying the size, position etc. this makes designing the user interface a major task in itself.

## ADVANTAGES OF VISUAL PROGRAMMING:-

- Visual development of graphical user interface which are easy to use and easy to learn.
- A programmer need not write code to display the required component.

- **For Example:-**

The visual programming environment displays a list of available components. The programmer picks up the required component from this list to display it.

- The component can be moved, resized and even deleted, if so required.
- There is no restriction on the number of controls that can be placed on a form.
- The interface components provided by the visual programming environment have some code built into them.

**For example:-**

A button' knows' when it has been clicked upon. In the case of conventional programming tools, the programmer has to write code to determine the component that has been clicked and then execute the appropriate code.

- Visual Basic is one of the most popular programming tools available today. And it's also secret that there have been massive changes in it as it became Visual Basic.Net.

- The reason of that change is Visual Basic itself, which has now become Visual Basic.Net. The difference between Visual Basic.Net and the previous version. Visual Basic 6.0 is revolutionary and far reaching. Visual Basic.Net has been more than four years in the marking and it represents entirely new directions for Visual Basic. Besides the biggest change integrated support for web development the very syntax, of techniques that you've probably learned carefully are now completely different such as data handling and many controls; project types and other aspects of Visual Basic 6.0 are no longer available at all.

- Visual Basic has a long and so far glorious history. When it first appeared, it created a revolution in windows programming. Visual Basic introduced unheard of ease to windows programming just builds the program you want right before your eyes, and then run it. In so doing it changed programming form a chore to something very like fun.

# INTRODUCTION TO BACK END TOOL

**Introduction to SQL: -**

SQL is a standard computer language for accessing and manipulating databases.

- SQL stands for **S**tructured **Q**uery **L**anguage.
- SQL allows you to access a database.
- SQL is an ANSI standard computer language.
- SQL can execute queries against a database.
- SQL can retrieve data from a database.
- SQL can insert new records in a database.
- SQL can delete records from a database.
- SQL can update records in a database.
- SQL is easy to learn.

- SQL is an ANSI (American National Standards Institute) standard computer language for accessing and manipulating database systems. SQL statements are used to retrieve and update data in a database. SQL works with database programs like MS Access, DB2, Informix, MS SQL Server, Oracle, Sybase, etc.

- Unfortunately, there are many different versions of the SQL language, but to be in compliance with the ANSI standard; they must support the same major keywords in a similar manner (such as SELECT, UPDATE, DELETE, INSERT, WHERE, and others).

- **SQL Database Tables: -**

A database most often contains one or more tables. Each table is identified by a name (e.g. "Customers" or "Orders"). Tables contain records (rows) with data.

**Below is an example of a table called "Persons": -**

| Last Name | First Name | Address | City |
|---|---|---|---|
| Hansen | Ola | Timoteivn 10 | Sandnes |
| Svendson | Tove | Borgvn 23 | Sandnes |
| Pettersen | Kari | Storgt 20 | Stavanger |

- The table above contains three records (one for each person) and four columns (Last Name, First Name, Address, and City).

- **SQL Queries: -**

With SQL, we can query a database and have a result set returned.

  - **A query like this: -**

```
SELECT Last Name FROM Persons
```

  - **Gives a result set like this: -**

| Last Name |
|---|
| Hansen |
| Svendson |
| Petersen |

**SQL Data Manipulation Language (DML)**

- SQL (Structured Query Language) is syntax for executing queries. But the SQL language also includes syntax to update, insert, and delete records.
- These query and update commands together form the Data Manipulation Language (DML) part of SQL: -

  - **SELECT** - extracts data from a database table
  - **UPDATE** - updates data in a database table
  - **DELETE** - deletes data from a database table
  - **INSERT INTO** - inserts new data into a database table

## SQL Data Definition Language (DDL)

- The Data Definition Language (DDL) part of SQL permits database tables to be created or deleted. We can also define indexes (keys), specify links between tables, and impose constraints between database tables.

- **The most important DDL statements in SQL are: -**

  - **CREATE TABLE** - creates a new database table
  - **ALTER TABLE** - alters (changes) a database table
  - **DROP TABLE** - deletes a database table
  - **CREATE INDEX** - creates an index (search key)
  - **DROP INDEX** - deletes an index MS SQL SERVER 2000

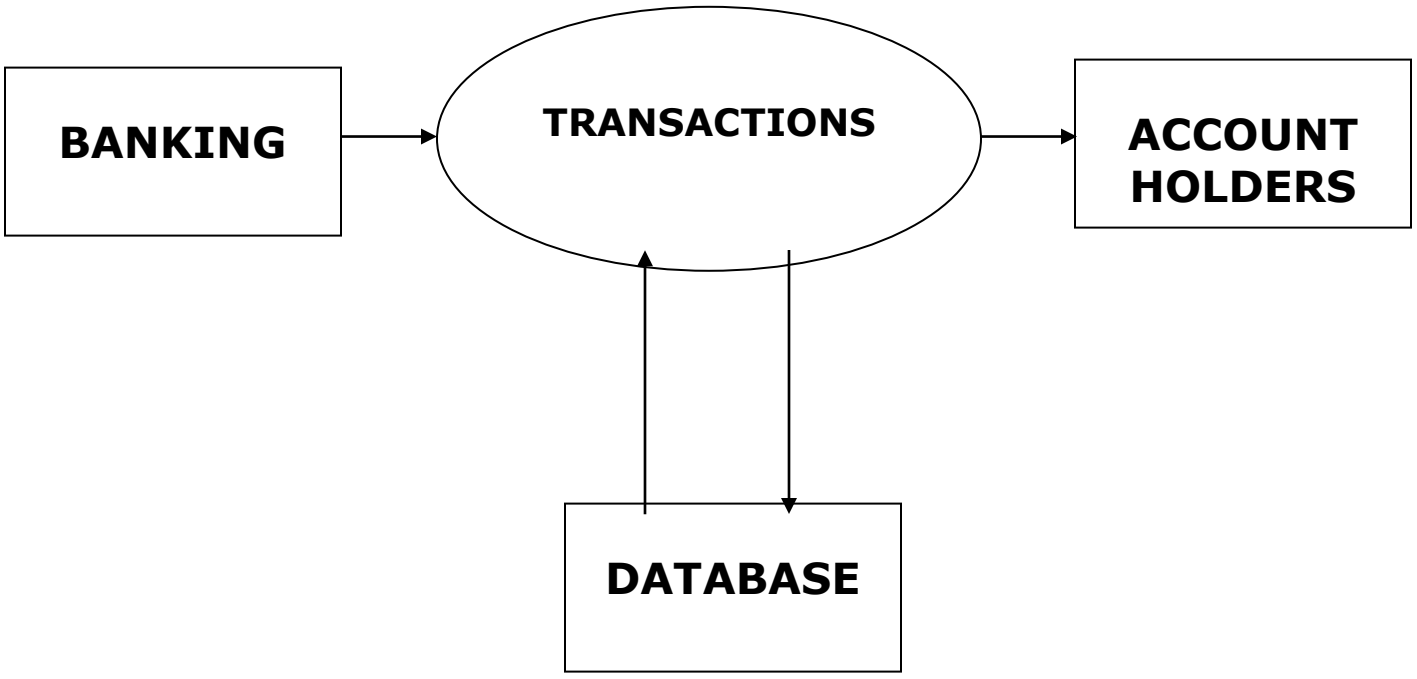
# DATA FLOW DIAGRAM

- **DATA FLOW DIAGRAM: -**The data flow diagram is also known as "bubble chart" has the purpose of clarifying system requirements and identifying major transformations that will become programs in system design so it is the starting point of specification down to the lowest level of detail. A DFDs consists of a series if bubbles joined by lines. The bubbles represent data transformation and the lines represent the data flow in the system.

- **DFD SYMBOLS:**

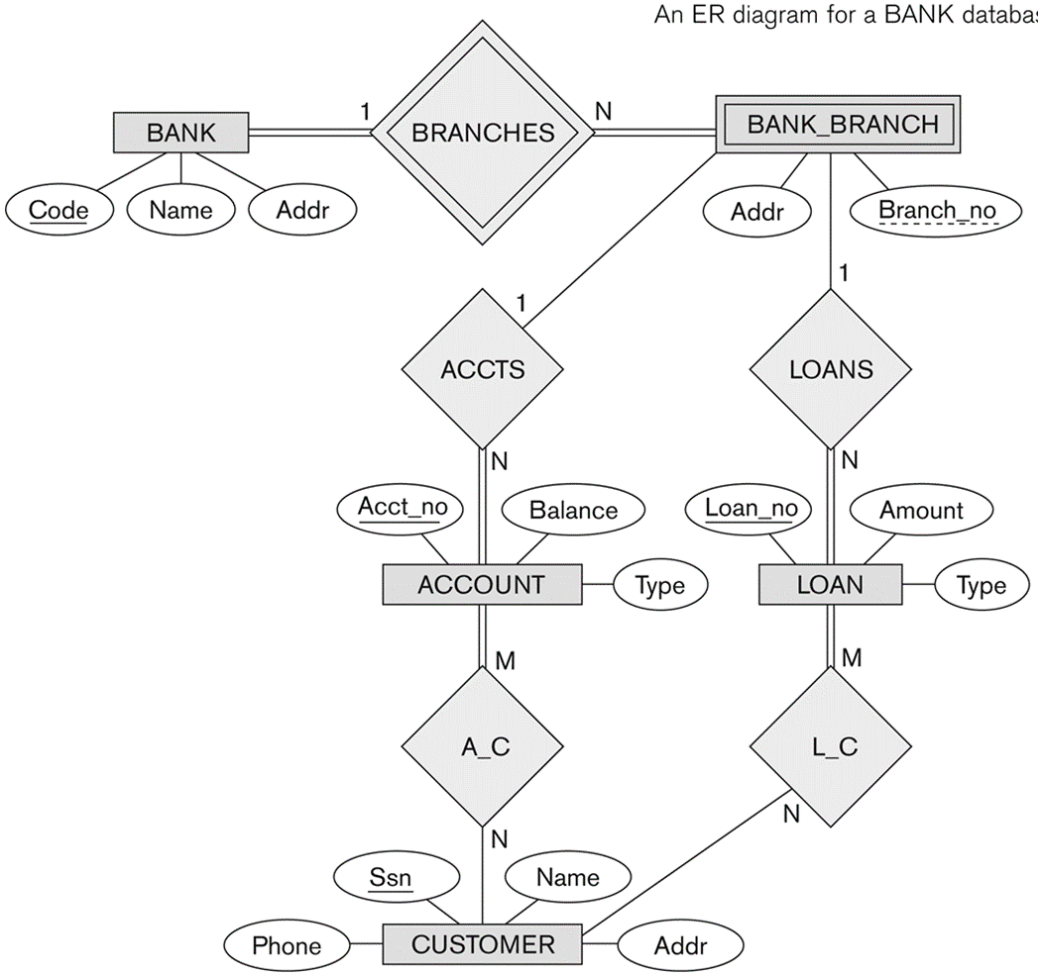
  - ✓ A system defined a source or destination of data.
  - ✓ An arrow identifies data flow, data in motion.
  - ✓ A circle represents the process that transforms incoming data flow to outgoing data flow.
  - ✓ An open rectangular is data store-data at rest or a temporary repository of data.

**SYSTEM DATA FLOW DIAGRAM**



# E-R DIAGRAM

An ER diagram for a BANK database schema.



ER-modeling is a data modeling technique used in software engineering to produce a conceptual data model of a information system. Diagrams created using this ER-modeling technique are called Entity-Relationship Diagrams, or ER diagrams or ERDs. So you can say that Entity Relationship Diagrams illustrate the logical structure of databases.

Dr. Peter Chen is the originator of the Entity-Relationship Model. His original paper about ER-modeling is one of the most cited papers in the computer software field. Currently the ER model serves as the foundation of many system analysis and design methodologies, computer-aided software engineering (CASE) tools, and repository systems.

The original notation for ER-Diagrams uses rectangles to represent entities, and diamonds to represent relationships.

There are three basic elements in ER-Diagrams:

- Entities are the "things" for which we want to store information. An entity is a person, place, thing or event.
- Attributes are the data we want to collect for an entity.
- Relationships describe the relations between the entities.

ERDs show entities in a database and relationships between tables within that database. It is essential to have ER-Diagrams if you want to create a good database design. The diagrams help focus on how the database actually works.

Entity (Instance)
An instance of a physical object in the real world.
Entity Class
: Group of objects of the same type.
–

E.g. Entity Class “Student”, Entities “John”, “Trish” etc

Attributes

Properties of Entities that describe their characteristics.

Types:

Simple

: Attribute that is not divisible, e.g. age.

Composite

: Attribute composed of several simple attributes,

e.g. address (house number, street, district)

Multiple

: Attribute with a set of possible values for the same

entity, e.g. Phone (home, mobile etc.) or email

Key

: Uniquely Ids the Entity e.g. PPSN, Chassis No.

Each simple attribute associated with a VS that may be assigned to that attribute for each individual entity,

e.g. age = integer

# DATA STRUCTURES AND DATABASE SPECIFICATIONS

**"ACCOUNT_INFO" Table: -**

| Field Name | Type | Constraints |
|---|---|---|
| Account_No | Int | Primary Key |
| Branch_No | Varchar(7) | References Branch_Info(Branch_No) |
| Branch_Name | VARCHAR(50) | Not Null |
| Account_H_Type | Varchar(15) | Not Null |
| No_Account_H | Varchar(5) | Not Null |
| Saluation_F | Varchar(5) | Not Null |
| Name_P_O_F | Varchar(50) | Not Null |
| Fa_Name_F | Varchar(50) | Not Null |
| Gender_F | Varchar(6) | Not Null |
| DOB_F | DateTime | Not Null |
| Age_F | Varchar(3) | Check(Age_F>=0 and Age_F<100) |
| Occupation_F | Varchar(15) | Not Null |
| Photo_F | Image | Not Null |
| Sign_F | Image | Not Null |
| Address_F | Varchar(100) | Not Null |
| Ph_No_F | Varchar(11) | Not Null |
| Mob_No_F | Varchar(14) | Not Null |

| | | |
|---|---|---|
| Saluation_S | Varchar(5) | |
| Name_P_O_S | Varchar(50) | |
| Fa_Name_S | Varchar(50) | |
| Gender_S | Varchar(6) | |
| DOB_S | DateTime | |
| Age_S | Varchar(3) | Check(Age_S>=0 and Age_S<100) |
| Occupation_S | Varchar(15) | |
| Photo_S | Image | |
| Sign_S | Image | |
| Address_S | Varchar(100) | |
| Ph_No_S | Varchar(11) | |
| Mob_No_S | Varchar(14) | |
| Saluation_T | Varchar(5) | |
| Name_P_O_T | Varchar(50) | |
| Fa_Name_T | Varchar(50) | |
| Gender_T | Varchar(6) | |
| DOB_T | DateTime | |
| Age_T | Varchar(3) | Check(Age_T>=0 and Age_T<100) |
| Occupation_T | Varchar(15) | |
| Photo_T | Image | |
| Sign_T | Image | |
| Address_T | Varchar(100) | |
| Ph_No_T | Varchar(11) | |
| Mob_No_T | Varchar(14) | |

| | | |
|---|---|---|
| Account_Type | Varchar(25) | Not Null |
| Witness_Name | Varchar(50) | Not Null |
| Witness_Sign | Image | Not Null |
| Nominee_Rel | Varchar(10) | Not Null |
| Nominee_Name | Varchar(50) | Not Null |
| Nominee_Sign | Image | Not Null |
| Opening_Bal | Varchar(10) | |
| T_Date | DateTime | Not Null |

**"Branch_Info" Table: -**

| Field Name | Type | Constraints |
|---|---|---|
| Branch_No | Varchar(7) | Primary Key |
| Branch_Name | Varchar(50) | |

**"Deposit_Info" Table: -**

| Field Name | Type | Constraints |
|---|---|---|
| Account_No | Int | References Account_Info(Account_No) |
| Branch_No | Varchar(7) | |
| Depositor_Name | Varchar(150) | |
| Account_H_Type | Varchar(15) | |
| Deposit_Amt | Varchar(10) | |

| | | |
|---|---|---|
| Deposit_Date | DateTime | |

**"Fixed_Info" Table: -**

| Field Name | Data type | Constraints |
|---|---|---|
| Account_No | Int | References Account_Info(Account_No) |
| Branch_No | Varchar(7) | |
| Depositor_Name | Varchar(50) | |
| Account_H_Type | Varchar(15) | |
| Time_Span | Varchar(5) | References Rate_Of_Interest_Info(Time_Span) |
| ROI | Varchar(5) | |
| Start_Date | DateTime | |
| Mature_Date | Varchar(15) | |
| Deposit_Amt | Varchar(10) | |
| Mature_Amt | Varchar(10) | |

**"Login_Info" Table: -**

| Field Name | Data type | Description |
|---|---|---|
| UserName | Varchar(20) | Primary Key |
| UserPassWord | Varchar(15) | |

| Field Name | Data type | Description |
|---|---|---|

| | | |
|---|---|---|
| Time_Span | Varchar(5) | Primary Key |
| ROI | Varchar(5) | |

**"Withdrawl_Info" Table: -**

| Field Name | Type | Constraints |
|---|---|---|
| Account_No | Int | References Account_Info(Account_No) |
| Branch_No | Varchar(7) | |
| Withdrawee_Name | Varchar(150) | |
| Account_H_Type | Varchar(15) | |
| Withdrawl_Amt | Varchar(10) | |
| Withdrawl_Date | DateTime | |

**"Loan_Info" Table: -**

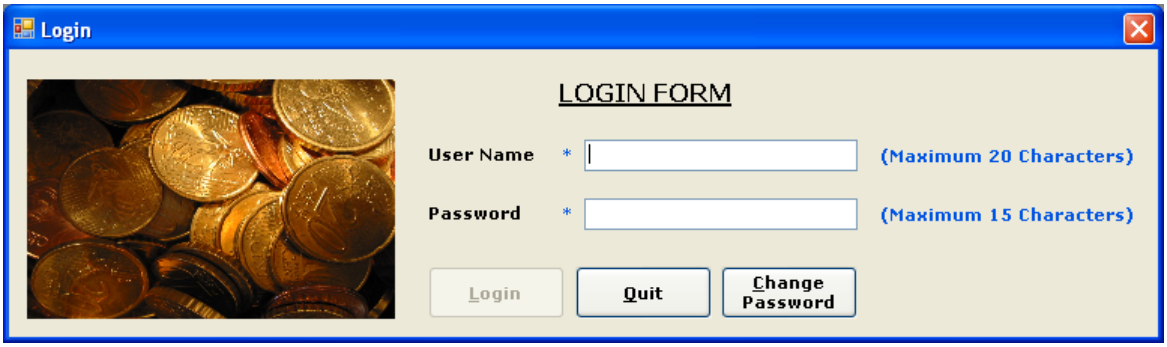
| Field Name | Type | Constraints |
|---|---|---|
| Account_No | Int | References Account_Info(Account_No) |
| Branch_No | Varchar(7) | |
| Acc_Holder_Name | Varchar(50) | |
| Account_Type | Varchar(15) | |
| Account_Sub_Type | Varchar(15) | |
| Time_Span | Varchar(5) | References Rate_Of_Interest_Info(Time_Span) |
| ROI | Varchar(5) | |

| | | |
|---|---|---|
| Issue_Date | DateTime | |
| Due_Date | Varchar(15) | |
| Loan_Sanctioned | Varchar(10) | |
| No_Installments | Varchar(5) | |
| EMI | Varchar(10) | |
| Total_Loan_Ret | Varchar(10) | |

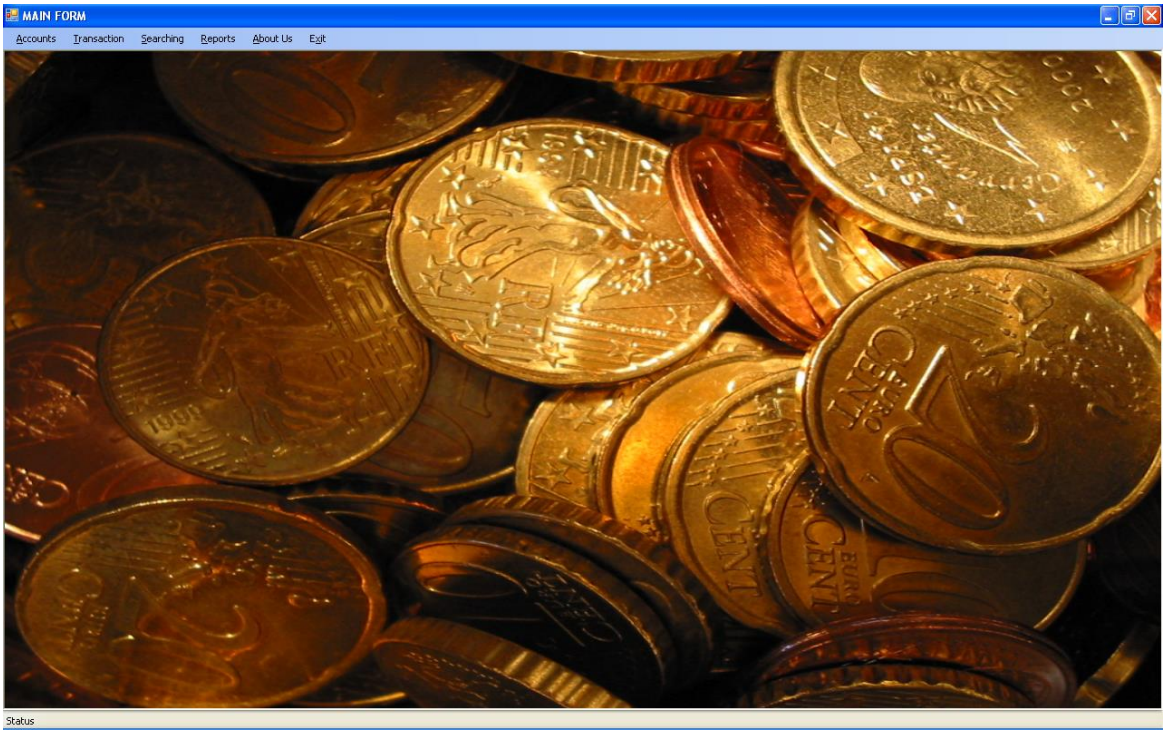
# DESIGN

- SPLASH FORM

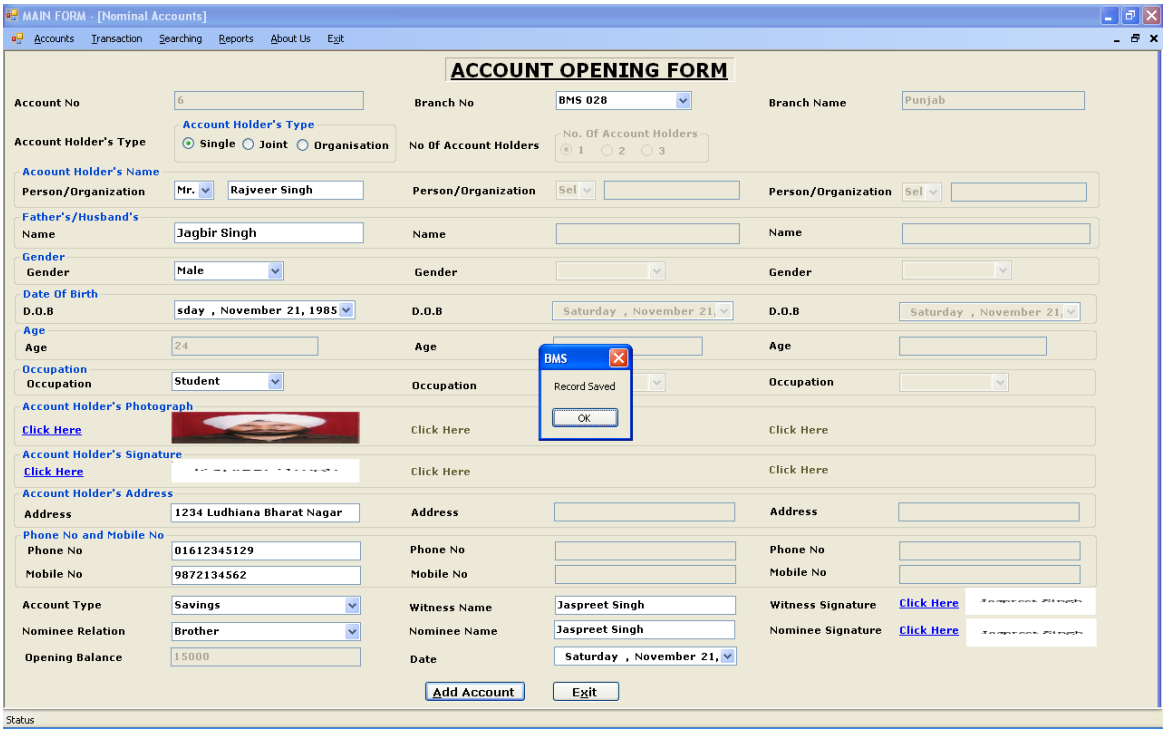

- LOGIN FORM



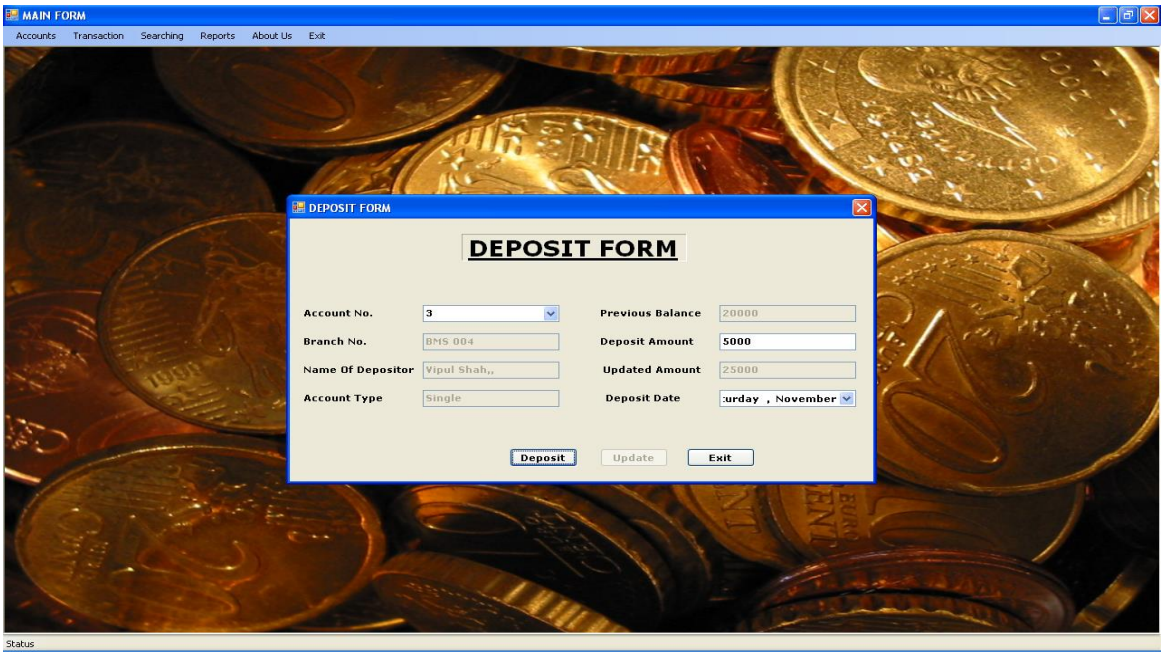
- MDI FORM


MAIN FORM
Accounts
Transaction
Searching
Reports
About Us
Exit
Status

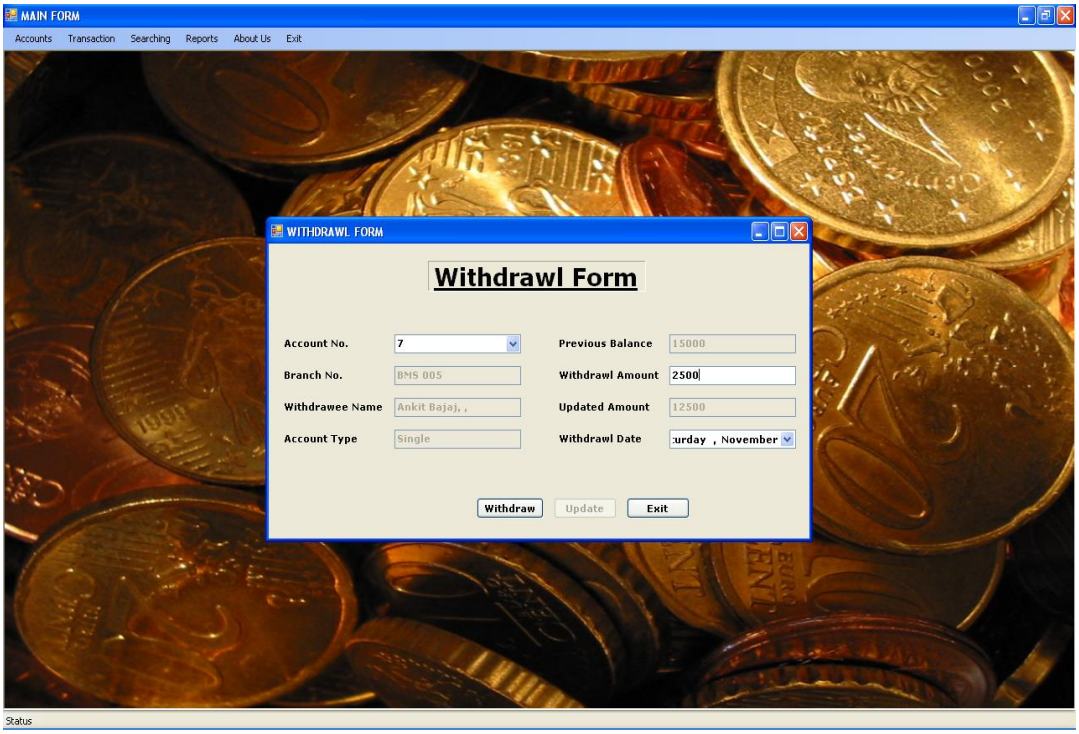
- ACCOUNT OPENING FORM



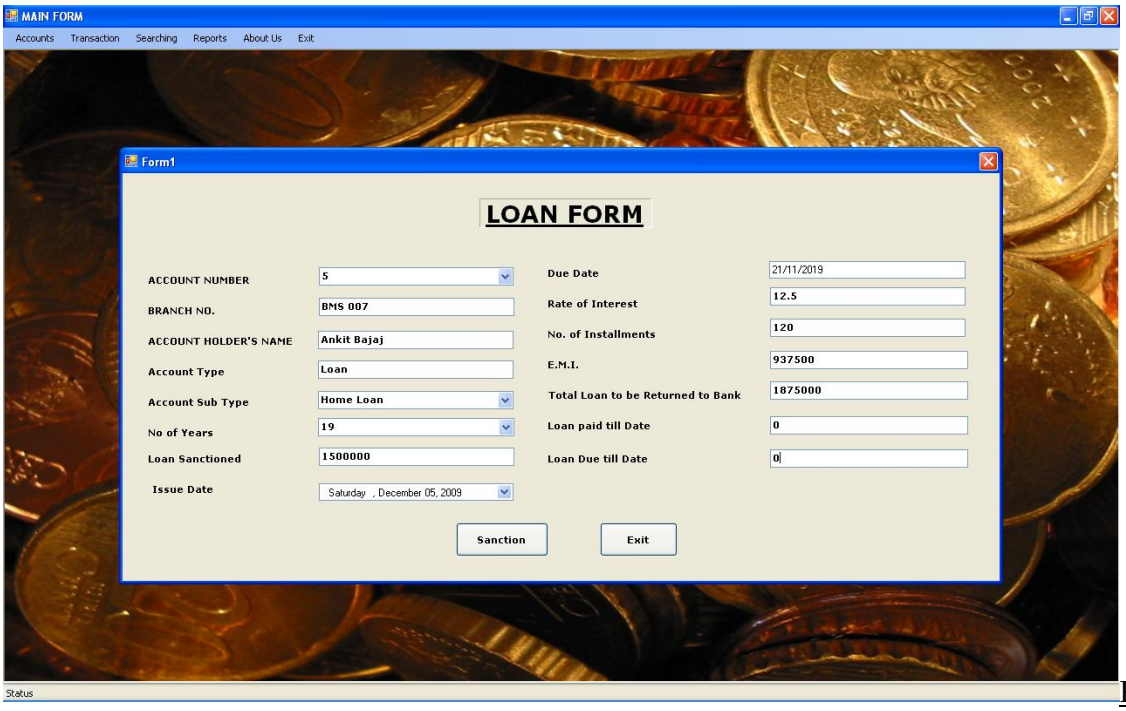

- DEPOSIT FORM




WITHDRAWL FORM


MAIN FORM
Accounts
Transaction
Searching
Reports
About Us
Exit
WITHDRAWL FORM
Withdrawl Form
Account No.
7
Previous Balance
15000
Branch No.
BMS 005
Withdrawl Amount
2500
Withdrawee Name
Ankit Bajaj, ,
Updated Amount
12500
Account Type
Single
Withdrawl Date
urday , November
Withdraw
Update
Exit
Status

- LOAN FORM


MAIN FORM
Accounts
Transaction
Searching
Reports
About Us
Exit
Form1
LOAN FORM
ACCOUNT NUMBER
5
BRANCH NO.
BMS 007
ACCOUNT HOLDER'S NAME
Ankit Bajaj
Account Type
Loan
Account Sub Type
Home Loan
No of Years
19
Loan Sanctioned
1500000
Issue Date
Saturday , December 05, 2009
Due Date
21/11/2019
Rate of Interest
12.5
No. of Installments
120
E.M.I.
937500
Total Loan to be Returned to Bank
1875000
Loan paid till Date
0
Loan Due till Date
0
Sanction
Exit
Status

R

E

# CODING

- **SPLASH FORM**

```
Public Class SPLASH
Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Timer1.Tick
        If ProgressBar1.Value = ProgressBar1.Maximum Then
            LOGIN.Show()
            Me.Hide()
            Timer1.Stop()
        Else
            ProgressBar1.Value = ProgressBar1.Value + 10
        End If
    End Sub
    Private Sub FlashLabel()
        Label1.ForeColor = Color.RoyalBlue
        Timer3.Start()
    End Sub
    Private Sub Timer2_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Timer2.Tick
        Call FlashLabel()
    End Sub
    Private Sub Timer3_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Timer3.Tick
        Label1.ForeColor = SystemColors.Control
        Timer3.Stop()
    End Sub
```

End Class

- **LOGIN FORM CODING**

```
Imports System.Data.SqlClient

Public Class LOGIN
    Private Sub LOGIN_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        Button1.Enabled = False
    End Sub
    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
        Dim A As Integer
        A = MsgBox("Do you want to exit", MsgBoxStyle.OkCancel, "Exit")
        If A = vbOK Then
            Application.Exit()
        End If
        If A = vbCancel Then
            Exit Sub
        End If
    End Sub
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
        Dim CON As New SqlConnection
        Dim CMD As New SqlCommand
        Dim DR As SqlDataReader
```

```
        CON.ConnectionString = "Server =ANGELDEVIL;Initial
Catalog=BankingSystem;Integrated Security=True"

        CON.Open()
        CMD.Connection = CON
        CMD.CommandText = "Select * From Login_Info"

        DR = CMD.ExecuteReader

        While DR.Read
            If DR.HasRows Then
                If TextBox1.Text = DR(0) And TextBox2.Text = DR(1) Then
                    Me.Hide()
                    Dim A As Integer
                    A = MsgBox("Login Successfully", MsgBoxStyle.OkOnly)
                    If A = vbOK Then
                        MAIN.Show()
                    End If
                    Return
                End If
            End If
        End While

    MsgBox("Invalid User Name or Password", MsgBoxStyle.Exclamation)

        TextBox1.Text = ""
        TextBox2.Text = ""
        Button1.Enabled = False
        TextBox1.Select()
        CMD.Dispose()
        CON.Close()
```

```
    End Sub
    Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button3.Click
        Me.Hide()
        CHANGE_PASSWORD_FORM.Show()
    End Sub

    Private Sub TextBox1_TextChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles TextBox1.TextChanged
        If TextBox1.Text = "" Then
            Button1.Enabled = False
        ElseIf TextBox2.Text = "" Then
            Button1.Enabled = False
        Else
            Button1.Enabled = True
        End If
    End Sub

    Private Sub TextBox2_TextChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles TextBox2.TextChanged
        If TextBox1.Text = "" Then
            Button1.Enabled = False
        ElseIf TextBox2.Text = "" Then
            Button1.Enabled = False
        Else
            Button1.Enabled = True
        End If
    End Sub
End Class
```

- **MDI FORM CODING**

```
Imports System.Windows.Forms

Public Class MAIN
    Private Sub ExitToolStripMenuItem1_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles ExitToolStripMenuItem1.Click
        Dim A As Integer
        A = MsgBox("Do You Want To Close", vbOKCancel)
        If A = vbOK Then
            Application.Exit()
        Else
            Return
        End If
    End Sub

    Private Sub DepositToolStripMenuItem1_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles DepositToolStripMenuItem1.Click
        NOMINAL_ACCOUNTS.MdiParent = Me
        NOMINAL_ACCOUNTS.Show()
    End Sub
    Private Sub DepositToolStripMenuItem_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles DepositToolStripMenuItem.Click
        DEPOSIT_FORM.MdiParent = Me
        DEPOSIT_FORM.Show()
    End Sub
    Private Sub WithdrawlToolStripMenuItem_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles WithdrawlToolStripMenuItem.Click
        WITHDRAWL.MdiParent = Me
        WITHDRAWL.Show()
```

```
    End Sub
    Private Sub HelpMenu_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles HelpMenu.Click
        ABOUT_US_FORM.MdiParent = Me
        ABOUT_US_FORM.Show()
    End Sub
    Private Sub FixedDepositToolStripMenuItem_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles FixedDepositToolStripMenuItem.Click
        FIXED_DEPOSIT.MdiParent = Me
        FIXED_DEPOSIT.Show()
    End Sub
    Private Sub AccountsToolStripMenuItem1_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles AccountsToolStripMenuItem1.Click
        ACCOUNTS_REPORT_FORM.MdiParent = Me
        ACCOUNTS_REPORT_FORM.Show()
    End Sub
    Private Sub SavingsToolStripMenuItem_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles SavingsToolStripMenuItem.Click
        DEPOSIT_REPORT.MdiParent = Me
        DEPOSIT_REPORT.Show()
    End Sub
    Private Sub CurrentToolStripMenuItem_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles CurrentToolStripMenuItem.Click
        FIXED_REPORT.MdiParent = Me
        FIXED_REPORT.Show()
    End Sub
    Private Sub RecurringToolStripMenuItem_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles RecurringToolStripMenuItem.Click
        RECURRING_REPORT.MdiParent = Me
        RECURRING_REPORT.Show()
    End Sub
```

```
    Private Sub WithdrawlToolStripMenuItem1_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles WithdrawlToolStripMenuItem1.Click
        WITHDRAWL_REPORT.MdiParent = Me
        WITHDRAWL_REPORT.Show()
    End Sub
    Private Sub LoanToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles LoanToolStripMenuItem.Click
        LOAN.MdiParent = Me
        LOAN.Show()
    End Sub
    Private Sub RecurringDepositToolStripMenuItem_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
RecurringDepositToolStripMenuItem.Click
        RECURRING_DEPOSIT.MdiParent = Me
        RECURRING_DEPOSIT.Show()
    End Sub
End Class
```

- **ACCOUNT OPENING FORM**

```
Imports System.Data.SqlClient
Imports System.IO

Public Class NOMINAL_ACCOUNTS

    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
        Me.Close()
    End Sub

    Private Sub NOMINAL_ACCOUNTS_Load(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles MyBase.Load

        '       Button1.Enabled = False

        TextBox1.ReadOnly = True
        TextBox1.Enabled = False
        TextBox2.ReadOnly = True
        TextBox2.Enabled = False

        Dim CON As New SqlConnection
        Dim CMD As New SqlCommand
        Dim DR As SqlDataReader

        CON.ConnectionString = "Server =ANGELDEVIL;Initial
Catalog=BankingSystem;Integrated Security=True"

        CON.Open()
        CMD.Connection = CON
```

```
CMD.CommandText = "Select * From Account_Info"
DR = CMD.ExecuteReader

While DR.Read
   TextBox1.Text = DR(0)
End While

DR.Close()

CMD.Dispose()
CON.Close()

TextBox1.Text = Val(TextBox1.Text) + 1

TextBox9.ReadOnly = True
TextBox9.Enabled = False
TextBox10.ReadOnly = True
TextBox10.Enabled = False
TextBox11.ReadOnly = True
TextBox11.Enabled = False
TextBox23.ReadOnly = True
TextBox23.Enabled = False

CON.Open()
CMD.Connection = CON
CMD.CommandText = "Select * From Branch_Info"
DR = CMD.ExecuteReader

While DR.Read
   ComboBox1.Items.Add(DR(0))
End While
```

```
        DR.Close()
        CMD.Dispose()
        CON.Close()

    End Sub

    Dim STRImage As String

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click

        IF TextBox1.Text = "" Or TextBox2.Text = "" Or TextBox3.Text = "" Or
TextBox4.Text = "" Or TextBox5.Text = "" Or TextBox6.Text = "" Or TextBox7.Text =
"" Or TextBox8.Text = "" Or TextBox9.Text = "" Or TextBox10.Text = "" Or
TextBox11.Text = "" Or TextBox12.Text = "" Or TextBox13.Text = "" Or
TextBox14.Text = "" Or TextBox15.Text = "" Or TextBox16.Text = "" Or
TextBox17.Text = "" Or TextBox18.Text = "" Or TextBox19.Text = "" Or
TextBox20.Text = "" Or TextBox21.Text = "" Or TextBox22.Text = "" Or
TextBox23.Text = "" Or ComboBox1.Text = "" Or ComboBox2.Text = "" Or
ComboBox3.Text = "" Or ComboBox4.Text = "" Or ComboBox5.Text = "" Or
ComboBox6.Text = "" Or ComboBox7.Text = "" Or ComboBox8.Text = "" Or
ComboBox9.Text = "" Or ComboBox10.Text = "" Or ComboBox11.Text = "" Or
ComboBox12.Text = "" Or RadioButton1.Checked = False Or RadioButton2.Checked =
False Or RadioButton3.Checked = False Or RadioButton4.Checked = False Or
RadioButton5.Checked = False Or RadioButton6.Checked = False Or
DateTimePicker1.Text = "" Or DateTimePicker2.Text = "" Or DateTimePicker3.Text =
"" Or DateTimePicker4.Text = "" Then

        MessageBox.Show("Plz fill the form")
```

```
        Exit Sub

        End If

        Dim CON As New SqlConnection
        Dim CMD As New SqlCommand

        CON.ConnectionString = "Server =ANGELDEVIL;Initial
Catalog=BankingSystem;Integrated Security=True"

        CON.Open()
        CMD.Connection = CON
        CMD.CommandType = CommandType.StoredProcedure
        CMD.CommandText = "InsertAccount_Info"


        'Try

        CMD.Parameters.Add("@Account_No", SqlDbType.Int).Value = TextBox1.Text
        CMD.Parameters.Add("@Branch_No", SqlDbType.VarChar).Value =
ComboBox1.Text
        CMD.Parameters.Add("@Branch_Name", SqlDbType.VarChar).Value =
TextBox2.Text
        If RadioButton1.Checked = True Then
            CMD.Parameters.Add("@Account_H_Type", SqlDbType.VarChar).Value =
RadioButton1.Text
        ElseIf RadioButton2.Checked = True Then
            CMD.Parameters.Add("@Account_H_Type", SqlDbType.VarChar).Value =
RadioButton2.Text
        ElseIf RadioButton3.Checked = True Then
```

```
        CMD.Parameters.Add("@Account_H_Type", SqlDbType.VarChar).Value =
RadioButton3.Text
    End If
    If RadioButton4.Checked = True Then
        CMD.Parameters.Add("@No_Account_H", SqlDbType.VarChar).Value =
RadioButton4.Text
    ElseIf RadioButton5.Checked = True Then
        CMD.Parameters.Add("@No_Account_H", SqlDbType.VarChar).Value =
RadioButton5.Text
    ElseIf RadioButton6.Checked = True Then
        CMD.Parameters.Add("@No_Account_H", SqlDbType.VarChar).Value =
RadioButton6.Text
    End If
    CMD.Parameters.Add("@Saluation_F", SqlDbType.VarChar).Value =
ComboBox2.Text
    CMD.Parameters.Add("@Name_P_O_F", SqlDbType.VarChar).Value =
TextBox3.Text
    CMD.Parameters.Add("@Fa_Name_F", SqlDbType.VarChar).Value =
TextBox6.Text
    CMD.Parameters.Add("@Gender_F", SqlDbType.VarChar).Value =
ComboBox5.Text
    CMD.Parameters.Add("@DOB_F", SqlDbType.DateTime).Value =
DateTimePicker1.Text
    CMD.Parameters.Add("@Age_F", SqlDbType.VarChar).Value = TextBox9.Text
    CMD.Parameters.Add("@Occupation_F", SqlDbType.VarChar).Value =
ComboBox8.Text

    'Catch ex As Exception
    'MessageBox.Show(ex.Message)
    'End Try
```

```
        'Try
        Dim FS As New FileStream(STRImage, FileMode.Open, FileAccess.Read)
        Dim BYTEARRAY(FS.Length) As Byte
        FS.Read(BYTEARRAY, 0, FS.Length)
        FS.Close()
        CMD.Parameters.AddWithValue("@Photo_F", BYTEARRAY)



        'Catch ex As Exception
        'MessageBox.Show(ex.Message)
        'End Try

        'Try
        Dim FS1 As New FileStream(STRImage, FileMode.Open, FileAccess.Read)
        Dim BYTEARRAY1(FS1.Length) As Byte
        FS1.Read(BYTEARRAY1, 0, FS1.Length)
        FS1.Close()
        CMD.Parameters.AddWithValue("@Sign_F", BYTEARRAY1)

        'Catch ex As Exception
        'MessageBox.Show(ex.Message)
        'End Try

        'Try
        CMD.Parameters.Add("@Address_F", SqlDbType.VarChar).Value =
TextBox12.Text
        CMD.Parameters.Add("@Ph_No_F", SqlDbType.VarChar).Value =
TextBox15.Text
```

```
        CMD.Parameters.Add("@Mob_No_F", SqlDbType.VarChar).Value =
TextBox18.Text

        CMD.Parameters.Add("@Saluation_S", SqlDbType.VarChar).Value =
ComboBox3.Text
        CMD.Parameters.Add("@Name_P_O_S", SqlDbType.VarChar).Value =
TextBox4.Text
        CMD.Parameters.Add("@Fa_Name_S", SqlDbType.VarChar).Value =
TextBox7.Text
        CMD.Parameters.Add("@Gender_S", SqlDbType.VarChar).Value =
ComboBox6.Text
        CMD.Parameters.Add("@DOB_S", SqlDbType.DateTime).Value =
DateTimePicker2.Text
        CMD.Parameters.Add("@Age_S", SqlDbType.VarChar).Value = TextBox10.Text
        CMD.Parameters.Add("@Occupation_S", SqlDbType.VarChar).Value =
ComboBox9.Text

        'Catch ex As Exception
        'MessageBox.Show(ex.Message)
        'End Try



        'Try
        Dim FS2 As New FileStream(STRImage, FileMode.Open, FileAccess.Read)
        Dim BYTEARRAY2(FS2.Length) As Byte
        FS2.Read(BYTEARRAY2, 0, FS2.Length)
        FS2.Close()
        CMD.Parameters.AddWithValue("@Photo_S", BYTEARRAY2)

        'Catch ex As Exception
        'MessageBox.Show(ex.Message)
```

```
        'End Try

        'Try
        Dim FS3 As New FileStream(STRImage, FileMode.Open, FileAccess.Read)
        Dim BYTEARRAY3(FS3.Length) As Byte
        FS3.Read(BYTEARRAY3, 0, FS3.Length)
        FS3.Close()
        CMD.Parameters.AddWithValue("@Sign_S", BYTEARRAY3)

        'Catch ex As Exception
        'MessageBox.Show(ex.Message)
        'End Try

        'Try
        CMD.Parameters.Add("@Address_S", SqlDbType.VarChar).Value =
TextBox13.Text
        CMD.Parameters.Add("@Ph_No_S", SqlDbType.VarChar).Value =
TextBox16.Text
        CMD.Parameters.Add("@Mob_No_S", SqlDbType.VarChar).Value =
TextBox19.Text

        CMD.Parameters.Add("@Saluation_T", SqlDbType.VarChar).Value =
ComboBox4.Text
        CMD.Parameters.Add("@Name_P_O_T", SqlDbType.VarChar).Value =
TextBox5.Text
        CMD.Parameters.Add("@Fa_Name_T", SqlDbType.VarChar).Value =
TextBox8.Text
        CMD.Parameters.Add("@Gender_T", SqlDbType.VarChar).Value =
ComboBox7.Text
        CMD.Parameters.Add("@DOB_T", SqlDbType.DateTime).Value =
DateTimePicker3.Text
```

```
        CMD.Parameters.Add("@Age_T", SqlDbType.VarChar).Value = TextBox11.Text
        CMD.Parameters.Add("@Occupation_T", SqlDbType.VarChar).Value =
ComboBox10.Text

        'Catch ex As Exception
        'MessageBox.Show(ex.Message)
        'End Try

        'Try
        Dim FS4 As New FileStream(STRImage, FileMode.Open, FileAccess.Read)
        Dim BYTEARRAY4(FS4.Length) As Byte
        FS4.Read(BYTEARRAY4, 0, FS4.Length)
        FS4.Close()
        CMD.Parameters.AddWithValue("@Photo_T", BYTEARRAY4)

        'Catch ex As Exception
        'MessageBox.Show(ex.Message)
        'End Try

        'Try
        Dim FS5 As New FileStream(STRImage, FileMode.Open, FileAccess.Read)
        Dim BYTEARRAY5(FS5.Length) As Byte
        FS5.Read(BYTEARRAY5, 0, FS5.Length)
        FS5.Close()
        CMD.Parameters.AddWithValue("@Sign_T", BYTEARRAY5)

        'Catch ex As Exception
        'MessageBox.Show(ex.Message)
        'End Try
```

```
        'Try
        CMD.Parameters.Add("@Address_T", SqlDbType.VarChar).Value =
TextBox14.Text
        CMD.Parameters.Add("@Ph_No_T", SqlDbType.VarChar).Value =
TextBox17.Text
        CMD.Parameters.Add("@Mob_No_T", SqlDbType.VarChar).Value =
TextBox20.Text
        CMD.Parameters.Add("@Account_Type", SqlDbType.VarChar).Value =
ComboBox11.Text
        CMD.Parameters.Add("@Witness_Name", SqlDbType.VarChar).Value =
TextBox21.Text

        'Catch ex As Exception
        'MessageBox.Show(ex.Message)
        'End Try

        'Try
        Dim FS6 As New FileStream(STRImage, FileMode.Open, FileAccess.Read)
        Dim BYTEARRAY6(FS6.Length) As Byte
        FS6.Read(BYTEARRAY6, 0, FS6.Length)
        FS6.Close()
        CMD.Parameters.AddWithValue("@Witness_Sign", BYTEARRAY6)

        'Catch ex As Exception
        'MessageBox.Show(ex.Message)
        'End Try

        'Try

        CMD.Parameters.Add("@Nominee_Rel", SqlDbType.VarChar).Value =
ComboBox12.Text
```

```
        CMD.Parameters.Add("@Nominee_Name", SqlDbType.VarChar).Value =
TextBox22.Text

        'Catch ex As Exception
        'MessageBox.Show(ex.Message)
        'End Try



        'Try
        Dim FS7 As New FileStream(STRImage, FileMode.Open, FileAccess.Read)
        Dim BYTEARRAY7(FS7.Length) As Byte
        FS7.Read(BYTEARRAY7, 0, FS7.Length)
        FS7.Close()
        CMD.Parameters.AddWithValue("@Nominee_Sign", BYTEARRAY7)

        'Catch ex As Exception
        'MessageBox.Show(ex.Message)
        'End Try

        'Try
        CMD.Parameters.Add("@Opening_Bal", SqlDbType.VarChar).Value =
TextBox23.Text
        CMD.Parameters.Add("@T_Date", SqlDbType.DateTime).Value =
DateTimePicker4.Text

        'Catch ex As Exception
        'MessageBox.Show(ex.Message)
        'End Try

        CMD.ExecuteNonQuery()
        CMD.Dispose()
```

```
CON.Close()

Dim A1 As Integer

A1 = MsgBox("Record Saved", MsgBoxStyle.OkOnly)

If A1 = vbOK Then

   TextBox1.Text = ""
   TextBox2.Text = ""
   TextBox3.Text = ""
   TextBox4.Text = ""
   TextBox5.Text = ""
   TextBox6.Text = ""
   TextBox7.Text = ""
   TextBox8.Text = ""
   TextBox9.Text = ""
   TextBox10.Text = ""
   TextBox11.Text = ""
   TextBox12.Text = ""
   TextBox13.Text = ""
   TextBox14.Text = ""
   TextBox15.Text = ""
   TextBox16.Text = ""
   TextBox17.Text = ""
   TextBox18.Text = ""
   TextBox19.Text = ""
   TextBox20.Text = ""
   TextBox21.Text = ""
   TextBox22.Text = ""
   TextBox23.Text = ""
```

```
        ComboBox1.Text = ""
        ComboBox2.Text = ""
        ComboBox3.Text = ""
        ComboBox4.Text = ""
        ComboBox5.Text = ""
        ComboBox6.Text = ""
        ComboBox7.Text = ""
        ComboBox8.Text = ""
        ComboBox9.Text = ""
        ComboBox10.Text = ""
        ComboBox11.Text = ""
        ComboBox12.Text = ""

        RadioButton1.Checked = False
        RadioButton2.Checked = False
        RadioButton3.Checked = False
        RadioButton4.Checked = False
        RadioButton5.Checked = False
        RadioButton6.Checked = False

        DateTimePicker1.Text = ""
        DateTimePicker2.Text = ""
        DateTimePicker3.Text = ""
        DateTimePicker4.Text = ""

    End If

End Sub
```

```
    Private Sub ComboBox1_SelectedIndexChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles ComboBox1.SelectedIndexChanged

        Dim CON As New SqlConnection
        Dim CMD As New SqlCommand
        Dim DR As SqlDataReader

        CON.ConnectionString = "Server =ANGELDEVIL;Initial
Catalog=BankingSystem;Integrated Security=True"

        CON.Open()
        CMD.Connection = CON
        CMD.CommandText = "Select * From Branch_Info"
        DR = CMD.ExecuteReader

        While DR.Read
            If ComboBox1.SelectedItem = DR(0) Then
                TextBox2.Text = DR(1)
            End If
        End While

        DR.Close()
        CMD.Dispose()
        CON.Close()

    End Sub

    Private Sub RadioButton2_CheckedChanged(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles RadioButton2.CheckedChanged

        GroupBox2.Enabled = True
```

RadioButton4.Enabled = False
RadioButton5.Enabled = True
RadioButton6.Enabled = True

ComboBox2.Enabled = True
ComboBox3.Enabled = True
ComboBox4.Enabled = True
TextBox3.Enabled = True
TextBox4.Enabled = True
TextBox5.Enabled = True
TextBox6.Enabled = True
TextBox7.Enabled = True
TextBox8.Enabled = True
ComboBox5.Enabled = True
ComboBox6.Enabled = True
ComboBox7.Enabled = True
DateTimePicker1.Enabled = True
DateTimePicker2.Enabled = True
DateTimePicker3.Enabled = True
TextBox9.Enabled = True
TextBox10.Enabled = True
TextBox11.Enabled = True
ComboBox8.Enabled = True
ComboBox9.Enabled = True
ComboBox10.Enabled = True
LinkLabel1.Enabled = True
LinkLabel2.Enabled = True
LinkLabel3.Enabled = True
PictureBox1.Enabled = True
PictureBox2.Enabled = True
PictureBox3.Enabled = True

```
LinkLabel4.Enabled = True
LinkLabel5.Enabled = True
LinkLabel6.Enabled = True
PictureBox4.Enabled = True
PictureBox5.Enabled = True
PictureBox6.Enabled = True
TextBox12.Enabled = True
TextBox13.Enabled = True
TextBox14.Enabled = True
TextBox15.Enabled = True
TextBox16.Enabled = True
TextBox17.Enabled = True
TextBox18.Enabled = True
TextBox19.Enabled = True
TextBox20.Enabled = True

If RadioButton2.Checked = True Then

   RadioButton4.Enabled = False
   RadioButton5.Enabled = True
   RadioButton6.Enabled = True

   If RadioButton5.Checked = True Then

      ComboBox2.Enabled = True
      ComboBox3.Enabled = True
      TextBox3.Enabled = True
      TextBox4.Enabled = True
      TextBox6.Enabled = True
      TextBox7.Enabled = True
      ComboBox5.Enabled = True
```

```
            ComboBox6.Enabled = True
            DateTimePicker1.Enabled = True
            DateTimePicker2.Enabled = True
            TextBox9.Enabled = True
            TextBox10.Enabled = True
            ComboBox8.Enabled = True
            ComboBox9.Enabled = True
            LinkLabel1.Enabled = True
            LinkLabel2.Enabled = True
            PictureBox1.Enabled = True
            PictureBox2.Enabled = True
            LinkLabel4.Enabled = True
            LinkLabel5.Enabled = True
            PictureBox4.Enabled = True
            PictureBox5.Enabled = True
            TextBox12.Enabled = True
            TextBox13.Enabled = True
            TextBox15.Enabled = True
            TextBox16.Enabled = True
            TextBox18.Enabled = True
            TextBox19.Enabled = True

        End If

    End If

End Sub

Private Sub RadioButton5_CheckedChanged(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles RadioButton5.CheckedChanged
```

```
        If RadioButton5.Checked = True Then

            ComboBox4.Enabled = False
            TextBox5.Enabled = False
            TextBox8.Enabled = False
            ComboBox7.Enabled = False
            DateTimePicker3.Enabled = False
            TextBox11.Enabled = False
            ComboBox10.Enabled = False
            LinkLabel3.Enabled = False
            PictureBox3.Enabled = False
            LinkLabel6.Enabled = False
            PictureBox6.Enabled = False
            TextBox14.Enabled = False
            TextBox17.Enabled = False
            TextBox20.Enabled = False

        End If

    End Sub

    Private Sub RadioButton1_CheckedChanged_1(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles RadioButton1.CheckedChanged

        If RadioButton1.Checked = True Then

            RadioButton4.Checked = True
            GroupBox2.Enabled = False
            ComboBox3.Enabled = False
            TextBox4.Enabled = False
            ComboBox4.Enabled = False
```

```
        TextBox5.Enabled = False
        TextBox7.Enabled = False
        TextBox8.Enabled = False
        ComboBox6.Enabled = False
        ComboBox7.Enabled = False
        DateTimePicker2.Enabled = False
        DateTimePicker3.Enabled = False
        TextBox10.Enabled = False
        TextBox11.Enabled = False
        ComboBox9.Enabled = False
        ComboBox10.Enabled = False
        LinkLabel2.Enabled = False
        PictureBox1.Enabled = False
        LinkLabel3.Enabled = False
        PictureBox2.Enabled = False
        LinkLabel5.Enabled = False
        PictureBox4.Enabled = False
        LinkLabel6.Enabled = False
        PictureBox5.Enabled = False
        TextBox13.Enabled = False
        TextBox14.Enabled = False
        TextBox16.Enabled = False
        TextBox17.Enabled = False
        TextBox19.Enabled = False
        TextBox20.Enabled = False

    End If

End Sub
```

```
    Private Sub RadioButton6_CheckedChanged(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles RadioButton6.CheckedChanged

        If RadioButton6.Checked = True Then

            RadioButton4.Enabled = False
            RadioButton5.Enabled = True
            RadioButton6.Enabled = True

            ComboBox2.Enabled = True
            ComboBox3.Enabled = True
            ComboBox4.Enabled = True
            TextBox3.Enabled = True
            TextBox4.Enabled = True
            TextBox5.Enabled = True
            TextBox6.Enabled = True
            TextBox7.Enabled = True
            TextBox8.Enabled = True
            ComboBox5.Enabled = True
            ComboBox6.Enabled = True
            ComboBox7.Enabled = True
            DateTimePicker1.Enabled = True
            DateTimePicker2.Enabled = True
            DateTimePicker3.Enabled = True
            TextBox9.Enabled = True
            TextBox10.Enabled = True
            TextBox11.Enabled = True
            ComboBox8.Enabled = True
            ComboBox9.Enabled = True
            ComboBox10.Enabled = True
            LinkLabel1.Enabled = True
```

```
            LinkLabel2.Enabled = True
            LinkLabel3.Enabled = True
            PictureBox1.Enabled = True
            PictureBox2.Enabled = True
            PictureBox3.Enabled = True
            LinkLabel4.Enabled = True
            LinkLabel5.Enabled = True
            LinkLabel6.Enabled = True
            PictureBox4.Enabled = True
            PictureBox5.Enabled = True
            PictureBox6.Enabled = True
            TextBox12.Enabled = True
            TextBox13.Enabled = True
            TextBox14.Enabled = True
            TextBox15.Enabled = True
            TextBox16.Enabled = True
            TextBox17.Enabled = True
            TextBox18.Enabled = True
            TextBox19.Enabled = True
            TextBox20.Enabled = True

        End If

    End Sub

    Private Sub RadioButton3_CheckedChanged(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles RadioButton3.CheckedChanged


        If RadioButton3.Checked = True Then
```

RadioButton4.Checked = True

GroupBox2.Enabled = False

ComboBox3.Enabled = False

TextBox4.Enabled = False

ComboBox4.Enabled = False

TextBox5.Enabled = False

TextBox7.Enabled = False

TextBox8.Enabled = False

ComboBox6.Enabled = False

ComboBox7.Enabled = False

DateTimePicker2.Enabled = False

DateTimePicker3.Enabled = False

TextBox10.Enabled = False

TextBox11.Enabled = False

ComboBox9.Enabled = False

ComboBox10.Enabled = False

LinkLabel2.Enabled = False

PictureBox1.Enabled = False

LinkLabel3.Enabled = False

PictureBox2.Enabled = False

LinkLabel5.Enabled = False

PictureBox4.Enabled = False

LinkLabel6.Enabled = False

PictureBox5.Enabled = False

TextBox13.Enabled = False

TextBox14.Enabled = False

TextBox16.Enabled = False

TextBox17.Enabled = False

TextBox19.Enabled = False

TextBox20.Enabled = False

```
        End If

    End Sub

    Private Sub DateTimePicker1_ValueChanged(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles DateTimePicker1.ValueChanged


        Dim Date1 As Date
        Dim A As Integer

        Date1 = Convert.ToDateTime(DateTimePicker1.Value)

        A = DateDiff(DateInterval.Year, Date1, DateTime.Now)

        TextBox9.Text = A

    End Sub
    'Dim count, count1, count2, count3, count4, count5, count6, count7 As Integer

    Dim count As Integer = 0
    Dim count1 As Integer = 0
    Dim count2 As Integer = 0
    Dim count3 As Integer = 0
    Dim count4 As Integer = 0
    Dim count5 As Integer = 0
    Dim count6 As Integer = 0
    Dim count7 As Integer = 0
```

```
    Private Sub LinkLabel1_LinkClicked(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.LinkLabelLinkClickedEventArgs) Handles
LinkLabel1.LinkClicked


        OpenFileDialog1.Filter = " Image files(*.jpg,*.bmp,*.gif)|*.jpg; *.gif*; *.bmp | all
files | * . * "

        If OpenFileDialog1.ShowDialog() = Windows.Forms.DialogResult.OK Then

            PictureBox1.Image = Image.FromFile(OpenFileDialog1.FileName)
            STRImage = OpenFileDialog1.FileName

        End If

        count = count + 1

    End Sub

    Private Sub LinkLabel2_LinkClicked(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.LinkLabelLinkClickedEventArgs) Handles
LinkLabel2.LinkClicked

        OpenFileDialog1.Filter = " Image files(*.jpg,*.bmp,*.gif)|*.jpg; *.gif*; *.bmp | all
files | * . * "

        If OpenFileDialog1.ShowDialog() = Windows.Forms.DialogResult.OK Then

            PictureBox2.Image = Image.FromFile(OpenFileDialog1.FileName)
            STRImage = OpenFileDialog1.FileName
```

```
        End If

        count1 = count1 + 1

    End Sub

    Private Sub LinkLabel3_LinkClicked(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.LinkLabelLinkClickedEventArgs) Handles
LinkLabel3.LinkClicked

        OpenFileDialog1.Filter = " Image files(*.jpg,*.bmp,*.gif)|*.jpg; *.gif*; *.bmp | all
files | * . * "

        If OpenFileDialog1.ShowDialog() = Windows.Forms.DialogResult.OK Then

            PictureBox3.Image = Image.FromFile(OpenFileDialog1.FileName)
            STRImage = OpenFileDialog1.FileName

        End If

        count2 = count2 + 1

    End Sub

    Private Sub LinkLabel4_LinkClicked(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.LinkLabelLinkClickedEventArgs) Handles
LinkLabel4.LinkClicked

        OpenFileDialog1.Filter = " Image files(*.jpg,*.bmp,*.gif)|*.jpg; *.gif*; *.bmp | all
files | * . * "
```

```
        If OpenFileDialog1.ShowDialog() = Windows.Forms.DialogResult.OK Then

            PictureBox4.Image = Image.FromFile(OpenFileDialog1.FileName)
            STRImage = OpenFileDialog1.FileName

        End If

        count3 = count3 + 1

    End Sub

    Private Sub LinkLabel5_LinkClicked(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.LinkLabelLinkClickedEventArgs) Handles
LinkLabel5.LinkClicked

        OpenFileDialog1.Filter = " Image files(*.jpg,*.bmp,*.gif)|*.jpg; *.gif*; *.bmp | all
files | * . * "

        If OpenFileDialog1.ShowDialog() = Windows.Forms.DialogResult.OK Then

            PictureBox5.Image = Image.FromFile(OpenFileDialog1.FileName)
            STRImage = OpenFileDialog1.FileName

        End If

        count4 = count4 + 1


    End Sub
```

```
    Private Sub LinkLabel6_LinkClicked(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.LinkLabelLinkClickedEventArgs) Handles
LinkLabel6.LinkClicked

        OpenFileDialog1.Filter = " Image files(*.jpg,*.bmp,*.gif)|*.jpg; *.gif*; *.bmp | all
files | * . * "

        If OpenFileDialog1.ShowDialog() = Windows.Forms.DialogResult.OK Then

            PictureBox6.Image = Image.FromFile(OpenFileDialog1.FileName)
            STRImage = OpenFileDialog1.FileName

        End If

        count5 = count5 + 1

    End Sub

    Private Sub LinkLabel7_LinkClicked(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.LinkLabelLinkClickedEventArgs) Handles
LinkLabel7.LinkClicked

        OpenFileDialog1.Filter = " Image files(*.jpg,*.bmp,*.gif)|*.jpg; *.gif*; *.bmp | all
files | * . * "

        If OpenFileDialog1.ShowDialog() = Windows.Forms.DialogResult.OK Then

            PictureBox7.Image = Image.FromFile(OpenFileDialog1.FileName)
            STRImage = OpenFileDialog1.FileName

        End If
```

```
        count6 = count6 + 1


    End Sub

    Private Sub LinkLabel8_LinkClicked(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.LinkLabelLinkClickedEventArgs) Handles
LinkLabel8.LinkClicked

        OpenFileDialog1.Filter = " Image files(*.jpg,*.bmp,*.gif)|*.jpg; *.gif*; *.bmp | all
files | * . * "

        If OpenFileDialog1.ShowDialog() = Windows.Forms.DialogResult.OK Then

            PictureBox8.Image = Image.FromFile(OpenFileDialog1.FileName)
            STRImage = OpenFileDialog1.FileName

        End If

        count7 = count7 + 1

    End Sub

    Private Sub DateTimePicker2_ValueChanged(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles DateTimePicker2.ValueChanged

        Dim Date1 As Date
        Dim A As Integer

        Date1 = Convert.ToDateTime(DateTimePicker2.Value)
```

```
        A = DateDiff(DateInterval.Year, Date1, DateTime.Now)

        TextBox10.Text = A

    End Sub

    Private Sub DateTimePicker3_ValueChanged(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles DateTimePicker3.ValueChanged

        Dim Date1 As Date
        Dim A As Integer

        Date1 = Convert.ToDateTime(DateTimePicker3.Value)

        A = DateDiff(DateInterval.Year, Date1, DateTime.Now)

        TextBox11.Text = A

    End Sub

    Private Sub ComboBox11_SelectedIndexChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles ComboBox11.SelectedIndexChanged

        If ComboBox11.SelectedItem = "Savings" Or ComboBox11.SelectedItem =
"Current" Then

            TextBox23.Text = "15000"

        Else
```

```
            TextBox23.Text = "0"
        End If



    End Sub
End Class
```

DEPOSIT FORM CODING

```
Imports System.Data.SqlClient

Public Class DEPOSIT

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click

        If TextBox1.Text = "" Or TextBox2.Text = "" Or TextBox3.Text = "" Or
TextBox4.Text = "" Or TextBox5.Text = "" Or TextBox6.Text = "" Or ComboBox1.Text
= "" Then
            MsgBox("Please Enter The Required Information",MsgBoxStyle.OkOnly)
            Return
        End If

        Dim CON As New SqlConnection
        Dim CMD As New SqlCommand

        CON.ConnectionString = "Server=ANGELDEVIL;Initial
Catalog=BankingSystem;Integrated Security=True"

        CON.Open()
        CMD.Connection = CON
        CMD.CommandType = CommandType.StoredProcedure
```

```
        CMD.CommandText = "InsertDeposit_Info"

        CMD.Parameters.Add("@Account_No", SqlDbType.Int).Value = ComboBox1.Text
        CMD.Parameters.Add("@Branch_No", SqlDbType.VarChar).Value =
TextBox1.Text
        CMD.Parameters.Add("@Depositor_Name", SqlDbType.VarChar).Value =
TextBox2.Text
        CMD.Parameters.Add("@Account_H_Type", SqlDbType.VarChar).Value =
TextBox3.Text
        CMD.Parameters.Add("@Deposit_Amt", SqlDbType.VarChar).Value =
TextBox5.Text
        CMD.Parameters.Add("@Deposit_Date", SqlDbType.DateTime).Value =
DateTimePicker1.Text

        CMD.ExecuteNonQuery()
        CMD.Dispose()
        CON.Close()

        MsgBox("Amount Deposited", MsgBoxStyle.OkOnly)

        Button2.Enabled = True

    End Sub

    Private Sub DEPOSIT_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load

        TextBox1.Enabled = False
        TextBox2.Enabled = False
        TextBox3.Enabled = False
        TextBox4.Enabled = False
```

```
        TextBox6.Enabled = False

        TextBox1.ReadOnly = True
        TextBox2.ReadOnly = True
        TextBox3.ReadOnly = True
        TextBox4.ReadOnly = True
        TextBox6.ReadOnly = True

        Button2.Enabled = False

        Dim CON As New SqlConnection
        Dim CMD As New SqlCommand

        Dim DR As SqlDataReader

        CON.ConnectionString = "Server=ANGELDEVIL;Initial
Catalog=BankingSystem;Integrated Security=True"

        CON.Open()
        CMD.Connection = CON
        CMD.CommandText = "Select * From Account_Info"
        DR = CMD.ExecuteReader

        While DR.Read
           ComboBox1.Items.Add(DR(0))
        End While

        DR.Close()
        CMD.Dispose()
        CON.Close()
```

```
    End Sub

    Private Sub ComboBox1_SelectedIndexChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles ComboBox1.SelectedIndexChanged

        Dim CON As New SqlConnection
        Dim CMD As New SqlCommand
        Dim DR As SqlDataReader

        CON.ConnectionString = "Server =ANGELDEVIL;Initial
Catalog=BankingSystem;Integrated Security=True"

        CON.Open()
        CMD.Connection = CON
        CMD.CommandText = "Select * From Account_Info"
        DR = CMD.ExecuteReader

        While DR.Read
            If ComboBox1.SelectedItem = DR(0) Then
                TextBox1.Text = DR(1)
                TextBox2.Text = DR(6) & "," & DR(18) & "," & DR(30)
                TextBox3.Text = DR(3)
                TextBox4.Text = DR(47)
            End If
        End While

        DR.Close()
        CMD.Dispose()
        CON.Close()
```

```
    End Sub

    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click

        Dim CON As New SqlConnection
        Dim CMD As New SqlCommand

        CON.ConnectionString = "Server=ANGELDEVIL;Initial
Catalog=BankingSystem;Integrated Security=True"

        CON.Open()
        CMD.Connection = CON
        CMD.CommandType = CommandType.StoredProcedure
        CMD.CommandText = "UpdateAccount_Info"

        CMD.Parameters.Add("@Account_No", SqlDbType.Int).Value = ComboBox1.Text
        CMD.Parameters.Add("@Opening_Bal", SqlDbType.VarChar).Value =
TextBox6.Text

        CMD.ExecuteNonQuery()
        CMD.Dispose()
        CON.Close()

        MsgBox("Previous Amount Updated", MsgBoxStyle.OkOnly)

        ComboBox1.Text = ""
        TextBox1.Text = ""
        TextBox2.Text = ""
        TextBox3.Text = ""
        TextBox4.Text = ""
```

```
        TextBox5.Text = ""
        TextBox6.Text = ""

        Button2.Enabled = False

    End Sub

    Private Sub TextBox5_TextChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles TextBox5.TextChanged
            TextBox6.Text = Val(TextBox4.Text) + Val(TextBox5.Text)
        End Sub

    Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button3.Click
        Me.Close()
    End Sub

End Class
```

- **WITHDRAWAL FORM CODING**

```
Imports System.Data.SqlClient

Public Class WITHDRAWL
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
        If TextBox1.Text = "" Or TextBox2.Text = "" Or TextBox3.Text = "" Or
TextBox4.Text = "" Or TextBox5.Text = "" Or TextBox6.Text = "" Or ComboBox1.Text
= "" Then
            MsgBox("Please Enter The Required Information", MsgBoxStyle.OkOnly)
            Return
```

```
        End If

        Dim CON As New SqlConnection
        Dim CMD As New SqlCommand

        CON.ConnectionString = "Server=ANGELDEVIL;Initial
Catalog=BankingSystem;Integrated Security=True"

        CON.Open()
        CMD.Connection = CON
        CMD.CommandType = CommandType.StoredProcedure
        CMD.CommandText = "InsertWithdrawl_Info"

        CMD.Parameters.Add("@Account_No", SqlDbType.Int).Value = ComboBox1.Text
        CMD.Parameters.Add("@Branch_No", SqlDbType.VarChar).Value =
TextBox1.Text
        CMD.Parameters.Add("@Withdrawee_Name", SqlDbType.VarChar).Value =
TextBox2.Text
        CMD.Parameters.Add("@Account_H_Type", SqlDbType.VarChar).Value =
TextBox3.Text
        CMD.Parameters.Add("@Withdrawl_Amt", SqlDbType.VarChar).Value =
TextBox5.Text
        CMD.Parameters.Add("@Withdrawl_Date", SqlDbType.DateTime).Value =
DateTimePicker1.Text

        CMD.ExecuteNonQuery()
        CMD.Dispose()
        CON.Close()

        MsgBox("Amount Withdrawn", MsgBoxStyle.OkOnly)
```

```
        Button2.Enabled = True
    End Sub
    Private Sub WITHDRAWL_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        TextBox1.Enabled = False
        TextBox2.Enabled = False
        TextBox3.Enabled = False
        TextBox4.Enabled = False
        TextBox6.Enabled = False

        TextBox1.ReadOnly = True
        TextBox2.ReadOnly = True
        TextBox3.ReadOnly = True
        TextBox4.ReadOnly = True
        TextBox6.ReadOnly = True

        Button2.Enabled = False

        Dim CON As New SqlConnection
        Dim CMD As New SqlCommand

        Dim DR As SqlDataReader

        CON.ConnectionString = "Server=ANGELDEVIL;Initial
Catalog=BankingSystem;Integrated Security=True"

        CON.Open()
        CMD.Connection = CON
        CMD.CommandText = "Select * From Account_Info"
        DR = CMD.ExecuteReader
```

```
        While DR.Read
            ComboBox1.Items.Add(DR(0))
        End While

        DR.Close()
        CMD.Dispose()
        CON.Close()
    End Sub
    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
        Dim CON As New SqlConnection
        Dim CMD As New SqlCommand

        CON.ConnectionString = "Server=ANGELDEVIL;Initial
Catalog=BankingSystem;Integrated Security=True"

        CON.Open()
        CMD.Connection = CON
        CMD.CommandType = CommandType.StoredProcedure
        CMD.CommandText = "UpdateAccount_Info"

        CMD.Parameters.Add("@Account_No", SqlDbType.Int).Value = ComboBox1.Text
        CMD.Parameters.Add("@Opening_Bal", SqlDbType.VarChar).Value =
TextBox6.Text

        CMD.ExecuteNonQuery()
        CMD.Dispose()
        CON.Close()

        MsgBox("Previous Amount Updated", MsgBoxStyle.OkOnly)
```

```
        ComboBox1.Text = ""
        TextBox1.Text = ""
        TextBox2.Text = ""
        TextBox3.Text = ""
        TextBox4.Text = ""
        TextBox5.Text = ""
        TextBox6.Text = ""

        Button2.Enabled = False
    End Sub
    Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button3.Click
        Me.Close()
    End Sub
    Private Sub ComboBox1_SelectedIndexChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles ComboBox1.SelectedIndexChanged
        Dim CON As New SqlConnection
        Dim CMD As New SqlCommand
        Dim DR As SqlDataReader

        CON.ConnectionString = "Server =ANGELDEVIL;Initial
Catalog=BankingSystem;Integrated Security=True"

        CON.Open()
        CMD.Connection = CON
        CMD.CommandText = "Select * From Account_Info"
        DR = CMD.ExecuteReader

        While DR.Read
            If ComboBox1.SelectedItem = DR(0) Then
                TextBox1.Text = DR(1)
```

```
            TextBox2.Text = DR(6) & ", " & DR(18) & ", " & DR(30)
            TextBox3.Text = DR(3)
            TextBox4.Text = DR(47)
        End If
    End While

    DR.Close()
    CMD.Dispose()
    CON.Close()
  End Sub
  Private Sub TextBox5_TextChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles TextBox5.TextChanged
    TextBox6.Text = Val(TextBox4.Text) - Val(TextBox5.Text)
  End Sub
End Class
```

- **LOAN FORM CODING**

```
Imports System.Data.SqlClient

Public Class LOAN
  Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    'If TextBox1.Text = "" Or TextBox2.Text = "" Or TextBox3.Text = "" Or
TextBox4.Text = "" Or TextBox5.Text = "" Or TextBox5.Text = "" Or ComboBox1.Text
= "" Or ComboBox2.Text = "" Or DateTimePicker1.Text = "" Then
    '   MsgBox("Please Enter The Required Information", MsgBoxStyle.OkOnly)
    '   Return
    'End If

    Dim CON As New SqlConnection
    Dim CMD As New SqlCommand
```

```
        CON.ConnectionString = "Server=ANGELDEVIL;Initial
Catalog=BankingSystem;Integrated Security=True"

        CON.Open()
        CMD.Connection = CON
        CMD.CommandType = CommandType.StoredProcedure
        CMD.CommandText = "InsertLoan_Info"

        CMD.Parameters.Add("@Account_No", SqlDbType.Int).Value = ComboBox1.Text
        CMD.Parameters.Add("@Branch_No", SqlDbType.VarChar).Value =
TextBox1.Text
        CMD.Parameters.Add("@Depositor_Name", SqlDbType.VarChar).Value =
TextBox2.Text
        CMD.Parameters.Add("@Account_H_Type", SqlDbType.VarChar).Value =
TextBox3.Text
        CMD.Parameters.Add("@Time_Span", SqlDbType.VarChar).Value =
ComboBox2.Text
        CMD.Parameters.Add("@ROI", SqlDbType.VarChar).Value = TextBox4.Text
        CMD.Parameters.Add("@Start_Date", SqlDbType.DateTime).Value =
DateTimePicker1.Text
        CMD.Parameters.Add("@Mature_Date", SqlDbType.VarChar).Value =
TextBox5.Text
        CMD.Parameters.Add("@Deposit_Amt", SqlDbType.VarChar).Value =
TextBox6.Text
        CMD.Parameters.Add("@Mature_Amt", SqlDbType.VarChar).Value =
TextBox7.Text

        CMD.ExecuteNonQuery()
        CMD.Dispose()
        CON.Close()
```

```
        MsgBox("Fixed Deposit Started", MsgBoxStyle.OkOnly)
    End Sub
    Private Sub LOAN _Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        TextBox1.Enabled = False
        TextBox2.Enabled = False
        TextBox3.Enabled = False
        TextBox4.Enabled = False
        TextBox5.Enabled = False
        TextBox7.Enabled = False
        DateTimePicker1.Enabled = False

        TextBox1.ReadOnly = True
        TextBox2.ReadOnly = True
        TextBox3.ReadOnly = True
        TextBox4.ReadOnly = True

        TextBox6.Enabled = False

        'Button2.Enabled = False

        Dim CON As New SqlConnection
        Dim CMD As New SqlCommand

        Dim DR As SqlDataReader

        CON.ConnectionString = "Server=ANGELDEVIL;Initial
Catalog=BankingSystem;Integrated Security=True"

        CON.Open()
```

```
        CMD.Connection = CON
        CMD.CommandText = "Select * From Account_Info Where Account_Type='
LOAN '"
        DR = CMD.ExecuteReader

        While DR.Read
            ComboBox1.Items.Add(DR(0))
        End While

        DR.Close()
        CMD.Dispose()
        CON.Close()

        CON.Open()
        CMD.Connection = CON
        CMD.CommandText = "Select * From Rate_Of_Interest_Info"
        DR = CMD.ExecuteReader

        While DR.Read
            ComboBox2.Items.Add(DR(0))
        End While

        DR.Close()
        CMD.Dispose()
        CON.Close()
    End Sub
    Private Sub ComboBox1_SelectedIndexChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles ComboBox1.SelectedIndexChanged
        Dim CON As New SqlConnection
        Dim CMD As New SqlCommand
        Dim DR As SqlDataReader
```

```
        CON.ConnectionString = "Server =ANGELDEVIL;Initial
Catalog=BankingSystem;Integrated Security=True"

        CON.Open()
        CMD.Connection = CON
        CMD.CommandText = "Select * From Account_Info"
        DR = CMD.ExecuteReader

        While DR.Read
            If ComboBox1.SelectedItem = DR(0) Then
                TextBox1.Text = DR(1)
                TextBox2.Text = DR(6) & " , " & DR(18) & " , " & DR(30)
                TextBox3.Text = DR(3)
                'TextBox4.Text = DR(47)
            End If
        End While

        DR.Close()
        CMD.Dispose()
        CON.Close()
    End Sub
    Private Sub TextBox5_TextChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles TextBox5.TextChanged
        'TextBox6.Text = Val(TextBox4.Text) + Val(TextBox5.Text)
    End Sub
    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
        Me.Close()
    End Sub
```

```
    Private Sub ComboBox2_SelectedIndexChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles ComboBox2.SelectedIndexChanged

        Dim CON As New SqlConnection
        Dim CMD As New SqlCommand
        Dim DR As SqlDataReader

        CON.ConnectionString = "Server=ANGELDEVIL;Initial
Catalog=BankingSystem;Integrated Security=True"

        CON.Open()
        CMD.Connection = CON
        CMD.CommandText = "Select * From Rate_Of_Interest_Info"
        DR = CMD.ExecuteReader

        While DR.Read
           If ComboBox2.SelectedItem = DR(0) Then
              TextBox4.Text = DR(1)
           End If
        End While

        DR.Close()
        CMD.Dispose()
        CON.Close()

        Dim a, b, c As String

        a = DateTimePicker1.Value.Date.Year
        c = DateTimePicker1.Value.Date.Month
        b = DateTimePicker1.Value.Date.Day  ' 19 '
```

```
        TextBox5.Text = b & "/ " & c & "/" & a + Val(ComboBox2.Text)

        TextBox6.Enabled = True
    End Sub
    Private Sub TextBox6_TextChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles TextBox6.TextChanged
        Dim A As Integer

        A = (Val(TextBox6.Text) * Val(ComboBox2.Text) * Val(TextBox4.Text)) / 100
        TextBox7.Text = TextBox6.Text + A
    End Sub
End Class
```

# TESTING AND DEBUGGING

## INTRODUCTION:-

The implementation phase of software development is concerned with translating design specification into source code. The preliminary goal of implementation is to write source code and internal documentation so that conformance of the code to its specifications can be easily verified, and so that debugging, testing and modifications are eased. This goal can be achieved by making the source code as clear and straightforword as possible. Simplicity, clarity and elegance are the hallmark of good programs, obscurity, cleverness, and complexity are indications of inadequate design and misdirected thinking.

Source code clarity is enhanced by structured coding techniques, by good coding style, by, appropriate supporting documents, by good internal comments, and by feature provided in modern programming languages.

The implementation team should be provided with a well-defined set of software requirement, an architectural design specification, and a detailed design description. Each team member must understand the objectives of implementation.

TERMS IN TESTING FUNDAMENTAL

**1. Error**

The term error is used in two ways. It refers to the difference between the actual output of software and the correct output, in this interpretation, error is essential a measure of the difference between actual and ideal. Error is also to used to refer to human action that result in software containing a defect or fault.

**2. Fault**

Fault is a condition that causes to fail in performing its required function. A fault is a basic reason for software malfunction and is synonymous with the commonly used term Bug.

**3. Failure**

Failure is the inability of a system or component to perform a required function according to its specifications. A software failure occurs if the behavior of the software is the different from the specified behavior. Failure may be caused due to functional or performance reasons.

**a. Unit Testing**

The term unit testing comprises the sets of tests performed by an individual programmer prior to integration of the unit into a larger system.

A program unit is usually small enough that the programmer who developed it can test it in great detail, and certainly in greater detail than will be possible when the unit is integrated into an evolving software product. In the unit testing the programs are tested separately, independent of each other. Since the check is done at the program level, it is also called program teasing.

**b. Module Testing**

A module and encapsulates related component. So can be tested without other system module.

**c. Subsystem Testing**

Subsystem testing may be independently design and implemented common problems are sub-system interface mistake in this checking we concentrate on it.

There are four categories of tests that a programmer will typically perform on a program unit.

1) Functional test
2) Performance test
3) Stress test
4) Structure test

**1) Functional Test**

Functional test cases involve exercising the code with Nominal input values for which expected results are known; as well as boundary values (minimum values, maximum values and values on and just outside the functional boundaries) and special values.

**2) Performance Test**

Performance testing determines the amount of execution time spent in various parts of the unit, program throughput, response time, and device utilization by the program unit. A certain amount of avoid expending too much effort on fine-tuning of a program unit that contributes little to the over all performance of the entire system. Performance testing is most productive at the subsystem and system levels.

**3) Stress Test**

Stress test are those designed to intentionally break the unit. A great deal can be learned about the strengths and limitations of a program by examining the manner in which a program unit breaks.

**4) Structure Test**

Structure tests are concerned with exercising the internal logic of a program and traversing particular execution paths. Some authors refer collectively to functional performance and stress testing as “black box” testing. While structure testing is referred

to as “white box” or “glass box” testing. The major activities in structural testing are deciding which path to exercise, deriving test date to exercise those paths, determining the test coverage criterion to be used, executing the test, and measuring the test coverage achieved when the test cases are exercised.

## DEBUGGING

Defect testing is intended to find areas where the program does not confirm to its specifications. Tests are designed to reveal the presence of defect in the system.When defect have been found in the program. There must be discovered and removed. This is called “Debugging”.

# FUTURE SCOPE OF THE PROJECT

This project can be handled in future by doing various modifications like: -

- We can go further for Online Banking.
- We can establish and start various Branches and available help centers for Account Holder's Queries.
- We can also deal through internet by creating web pages and a banking website for internet dealing.
- To attract Account Holder's we can offer various offers during festivals months.
- We can also deal in various types of Banking Transactions.
- To have more and more customer satisfaction we will emphasize more and more on our dealings.

# REFERENCES

[1] Iztok Fajfar “Start Programming Using HTML,CSS and JavaScript” issue 21 October 2015 Pub. Location New York.

[2] Preston Zhang “Practical Guide to Oracle SQL,T-SQL and MySQL” issue 21 November 2017 Pub. Location Boca Raton

[3]Mitsunori Ogihara “Fundamentals of Java Programming” issue 25 January 2016 Pub. Springer,Cham

[4]Richard Wiener “Fundamentals of OOP and Data Structures in Java” issue on June 2012 Pub. Cambridge University Press.