BANGALORE NORTH UNIVERSITY

A PROJECT REPORT ON

"MASSMEDIA ADVERTISEMENT MANAGEMENT SYSTEM"

Submitted By

TEJAS.R

(R1911939)

A 5th SEM project report submitted in partial fulfillment of the requirements the award of degree in

BACHELOR OF COMPUTER APPLICATIONS

Under the Guidance of

Mr Satyajit Ray

(Dept. of Computer Science)

SDC COLLEGE KOLAR

Smt.DANAMMA CHANNABASAVAIAH COLLEGE OF ARTS,COMMERECE,SCIENCE & MANAGEMENT STUDIES

NH-75,KOLAR BYPASS NEAR KODIRAMASANDRA KOLAR-563101

(NAAC Accredited with 'B' Grade & affiliated to Bangalore North University)

Smt.DANAMMA CHANNABASAVAIAH COLLEGE OF ARTS, COMMERCE,SCIENCE & MANAGEMENT STUDIES,KOLAR-563101

CERTIFICATE

Certificate that project report titled **"Massmedia Advertisement Management System"** Based on the project work jointly carried out by **TEJAS.R** Bearing Registration number **R1911939** and respectively of **5th semester BCA**. This report is submitted in partial fullfillment of the requirements for the award of the **Degree in Bachelor of Computer Applications** by **Bangalore North University**,duringthe **Academic Year 2022.**

HOD	Guide
Mr.SATYAJIT RAY	**Mr.SATYAJIT RAY**
(Dept.of computer science)	**(Dept.computer science)**

Examiners:

1.

2.

Smt.DANAMMA CHANNABASAVAIAH COLLEGE OF ARTS,COMMERCE,SCIENCE & MANAGEMENT STUDIES,KOLAR-563101

ACKNOWLEDGEMENT

We take this opportunity to express our deep sense of gratitude to our founder chairperson **Smt. Usha Gangadhar.**

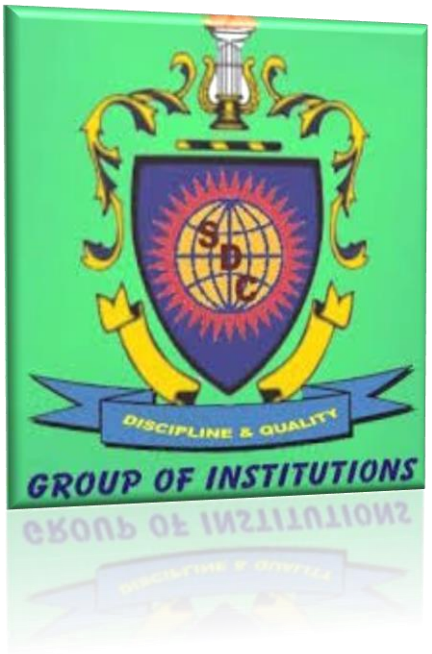
Prof.Pushpalatha,Principal for her valuable guidance ,keen interest and help during the course of study.

We express our sincere thanks to our recpected **HOD Mr.Satyajit Ray** for providing all necessary help during our project work.

Last but not the least we would like to thank our **friends,staff** and **all others** who have directly and indirectly helped us in the successful completion of this project

TEAM MEMBER
TEJAS.R (R1911939)

Smt.DANAMMA CHANNABASAVAIAH COLLEGE OF ARTS, COMMERCE,SCIENCE & MANAGEMENT STUDIES,KOLAR - 563101



UNDERTAKING

I **TEJAS.R (R1911939)** studying in 5th Semester BCA,hereby undertake that project has been carried out by us as a part of fulfillment of the reuirements for the award of the degree as prescribed by Banglore North University.The project was carried out under the guidance of **Mr.SATYAJIT RAY.**

Signature of the Students:

TEJAS.R
(R1911939)

CONTENTS

Synopsis

Project Title :-

MASSMEDIA ADVERTISEMENT MANAGEMENT SYSTEM

Submitting By :-

TEJAS.R (R1911939)

RAJENDRA.V (R1911901)

Software Requirements :-

Operating System : Windows 10 64-bit operating system

Front End : Visual Basic .NET (2015)

Back End : SQL SERVER DATABASE

Hardware Requirements:-

Processor : Intel(R) Pentium CPU @ 2GHz

Moniter : any display unit

Hard Disk : 200GB and above

RAM : 2.00GB

TITLE OF THE PROJECT :

MASSMEDIA ADVERTISEMENT MANAGEMENT SYSTEM

AIM OF THE PROJECT:

The project **"MASSMEDIA ADVERTISEMENT MANAGEMENT SYSTEM"** is in which gives public facility that provides mail services ,including accepting of letters and parcels, providing post office boxes and selling postage stamps, packaging, and stationery.

The project " **MASSMEDIA ADVERTISEMENT MANAGEMENT SYSTEM**" contains 10 modules

1. LOGIN

a. Username
b. Password

2. ADBooking

a. RONo
b. Categorie
c. Publisher
d. Type
e. Size
f. DOP
g. Edition
h. image

3. ADBilling

a. RONo
b. Bill Date
c. Client Name
d. Mobile No

e. Bill Amount
f. Payment Mode
g. Ref No
h. Image

4. ADRecord

a. RONo
b. Date of Publish

5. ADCalculator

a. Categorie
b. Publisher
c. Type
d. Size
e. Total

6. ADReport

a. From Year
b. To Year

REPORT:

It gives the data report about the Advertisement Management details.

INTRODUCTION

INTRODUCTION

ABOUT THE PROJECT :

The project deals with the **"MASSMEDIA ADVERTISEMENT MANAGEMENT SYSTEM".**
The data grouped differently to process and handle them effectively .

This is a menu driven, user friendly software with a provision to modify the required in the future. The software has been developed in an easy and understandable way.

This software has been developed using VISUAL BASIC .NET As front end and SQL Server as back end. The operating system used to develop this project in windows 10. One can work with different files at the same time, which saves space and also handling time during the process of searching and processing of data.
MASSMEDIA ADVERTISEMENT MANAGEMENT SYSTEM is a type of properly management system software which facilitates the management of **ADVERTISEMENT** operations and functions.

Feasibility Report

Feasibility report

Understanding feasibility:

Feasibility study means the analysis of problem to determine if it can be solved effectively. In other words, it is the study of the possibilities of the proposed system it studies the work ability, impact on the organization ability to meet user's need and efficient use of resources.

Three aspects in which the system has to be feasible are:-

Economic feasibility:

The economic analysis checks for the high investment incurred on the system. It evaluates development and Implementing charges for the proposed "**Massmedia Advertisement Management System**". The software used for the development is easily available at minimal cost and the database applied is freely available hence it results in low cost implementation.

Technical feasibility:

This aspect concentrates on the concept of using Computer Meaning, "Mechanization" of human works. Thus, the automated solution leads to the need for a technical feasibility study. The focus on the platform used database management and users for that software. The proposed system doesn't require an in-depth technical knowledge as the system development is simple and easy to understand. The software (JAVASCRIPT) used makes the system user friendly (GUI). The result obtained should be true in the real time conditions.

Behavioral feasibility:

Behavioral feasibility deals with the runtime performance of the software the proposed system must score higher than the present in the behavioral study. The software should have end user in mind when the system is designed while designing software the programmer should be aware of the conditions user's knowledge input, output, calculations etc. The software contains only a minimum no. of bugs. Care should be also taken to avoid non-working means and buttons.

SOFTWARE DEVELOPMENT LIFE CYCLE

SOFTWARE DEVELOPMENT LIFE CYCLE (SDLC)

MODELING:

During the evaluation and solution synthesis activity, the analyst creates models of the system in an effort to better understand data and control flow.

The model serves has a foundation for the software design and basis for the creation of specification for the software. For the better understanding of data and control flow we use Data Flow Diagram

The principle stages of SDLC are:

- **Requirement Analysis and Definition:**

After feasibility study has been performed, the requirement analysis which includes the software system's services. constraints and goals is established after consulting the system users. Then the exact requirements are defined in detail.

- **System and Software Design:**

Once the requirements for a system have been documented the software system must be designed to meet them. The system design process divides the

requirements into either hardware or software systems along with the overall system architecture. Software design involves designing abstraction and their relationships.

- **Integration and System Testing:**

All the modules or individual programs that have been developed and tested individually are integrated and tested has a whole system to ensure the software requirements have been meet. After system testing the software is delivered to the coustmer.

- **Operation and Maintainance:**

It meets the new requirements , if any .Software training and support is also important, since software is effective only if used correctly
After feasibility study has been performed the requirement analysis which includes the software system's services. constraints and goals is established after consulting the system users. Then the exact requirements are defined in detail.

- **System and Software Design:**

Once the requirements of a system have been documented the software system must be designed to meet them. The system design process divides the requirements into either hardware or software system along with the
Over all architecture. System design involves designing abstraction and their relationships

- **Implementation and Unit Testing:**

This process produces the actual code has a set of programs. Testing is an integral and important phase of software development process. Unit testing ensures that defects are recognized quickly and that each unit meets its specification.

- **Integration and System Testing:**

All the modules or individual programs that have been developed and tested individually are integrated and tested has a whole system to ensure the software requirements have been meet. After system testing the software is deliverd to the customer.

- **Operation and System Testing:**

This phase involves installation ,customization, testing and evaluation Maintainance involves correcting errors, which were missed in the earlier stages. It also involves changes or improvement of the system units so that it meets the new requirements, if any. Software training and support is also important, since software is effective only if used correctly.

Software process models:

Software process model is a simple description of software process. Types of Software process models:

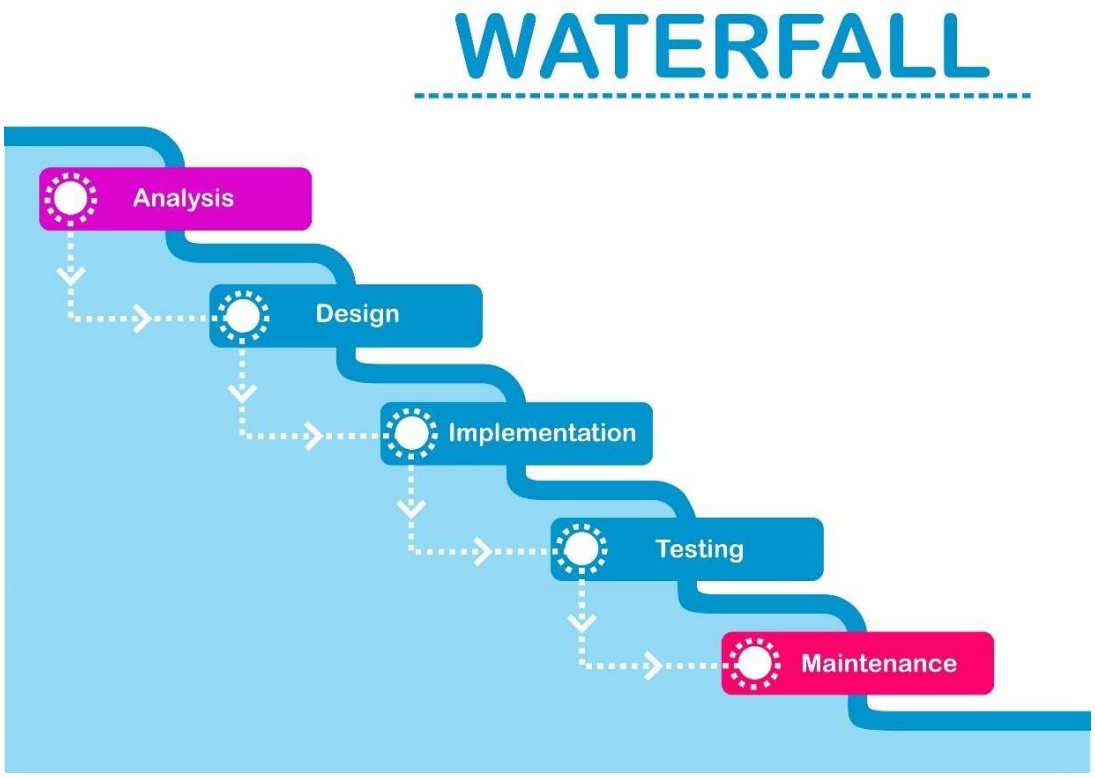
➢ Waterfall Model

➢ Spiral Model

➢ Iterative Enhancement Model

WATERFALL FALL MODEL:

The simplest model is the Waterfall Model or linear sequential model that states that the phases at organized in a linear order.



❖ **Requirement analysis and specification:** Requirement analysis is taken up after a feasibility study has been performed to define the costs and benefits of a software system.

❖ **Design:** Once the requirements for a system have been documented software design a software system to meet them.

❖ **Coding and Unit Testing:** This phase produces the actual code that will be delivered to the customer.

❖ **Integration and System testing:** All the modules that have been developed before and tested individually are put together (integrated) in this phase and tested as a whole.

❖ **Delivery and maintenance:** Once the system passes all the tests it is delivered to the customer and enters the maintenance phase.

Advantages of Waterfall model:

❖ The project requires the fulfillment of one phase, before proceeding to the next.

❖ The Waterfall model is a straight forward method and lets one know easily what stages is in progress.

❖ The Waterfall model is an oldest and most widely used model in the field of software development there are certain advantages of this model which makes it one of the most widely used models.

❖ Being a linear model, it is very simple to implement the amount of resource required to implement this model or minimal.

❖ After every major stage of software coding, testing is done to check the correct running of the code.

Disadvantages of Waterfall Model:

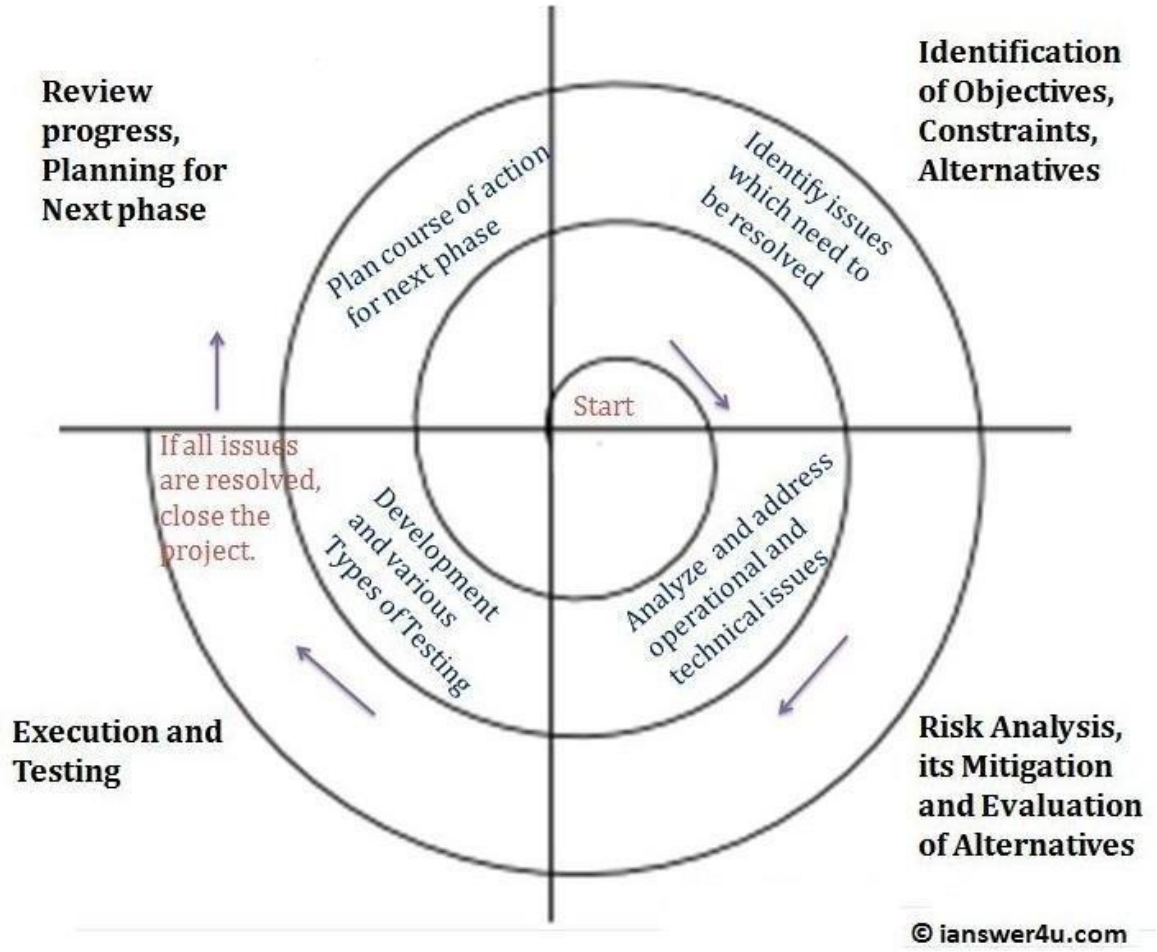
❖ Ironically, the biggest disadvantage is one of its greatest advantages. User or developer cannot go back a step if the design phase has gone wrong, things can get very complicated in the implemented phase.

So, this in short was all about Waterfall model, its advantages and disadvantages. Inspire of the cons, the many pros of this model ensure that it remains one con in many pros of this model. Model ensures that it remains one of the most popular models used in the field of software development.

SPIRAL MODEL:

The spiral model is similar to the increment model, with more emphasis placed on risk analysis.The spiral model has four phases:

- Planning
- Risk Analysis
- Engineering
- Evaluation



A software product repeatedly passes through the these phases in iterations (called spirals in this model).the baseline spirals ,starting in the planning phase ,requirements are gathered and risk is assessed. Each subsequent spiral builds on the baseline spirals. Requirements are gathered during the planning phase. In the risk analysis phase, a process is undertaken to identify risk and alternate solutions. A prototype is produced at the end of the risk analysis phase.

Software is produced in the engineering phase, along with the testing at the end of the phase. The evaluation phase allows the customer to evaluate the output of the project to date before the project continues to the next spiral.

The spiral model is the combination of water fall model and prototype. It also includes the features of the risk identification and risk analysis. The spiral model consist several phases, each phase is a spiral or loop represents some specific task. Once one spiral is over control goes to next spiral. It is divided into four phases which are as follows:

- Objective setting
- Identifying and risk analysis
- Development and validation
- Planning for next

- **Objective Setting:**

The objective setting includes requirements analysis, constraints, target and restrictions of the system along with risk monitoring.

- **Identifying and risk analysis:**

Once the risk is identified,then the methods are implemented to access analyze and reduce the risk.

- **Development and validation:**

During this phase, the spiral of the system is designed, coded and tested along with the various testing stratagies.

- **Planning for next iteration:**

This phase includes the future planning for next iteration based on prototype and risk management.

Advantages:

- Reduces the risk in a system development.
- The risk is analyzed and eliminated accordingly.
- Prototype gives the option of evolution.
- It reduces the chances of system failure.
- The spiral model is compact and iterative.
- The spiral model has planning strategy for easy spiral.

Disadvantages:

- It is expensive compare to waterfall model.
- It requires more skilled persons.
- The risk assessment and analysis sometime is not accepted by the clients.

FEASIBILITY STUDY

FEASIBILITY STUDY

All projects are feasible given unlimited resoures and infinite time. But the development of computer based system is likely to be plagued by scarcity of resources and difficult completion dates. A system , which is ill conceived , if recognized early will avert month or year of efforts, thousands of dollars professionals embarrassment etc.

While discussing about feasibility we need to concentrate on the points given below:

- Economic feasibility
- Technical feasibility
- Operational feasibility

Economic feasibility : Economic justification is generally the bottom line considering for most system. Economic justification includes aboard range of concerns that includes cost *benefits* analysis.

Computer department develop the proposed system. The system will be developed and operate in the existing hardware and software infrastructure. So there is no need of obtaining hardware and software for the system.

The purpose system will increase the performance, which in turn may be expected increase profit. Existing employes will be training in using and operation the employed system.

Thus elemenating the need of recruiting new employes in the organization. Thus, the project is economicaly feasibility for the development for the company.

Technical feasibility:

The technical feasibility is the most difficult area to encounter at this stage.

It is essential that the process of analysis an definition be conducted in parallel with an assessment of technical feasibility.

It centers on the existing computer system and to what extends can support the proposed system. Through information in manual is enormous, it is handled easily by VISUAL BASIC .NET information in the table form is easy to access and manage,

computer department is already equipped with suitable system and it can be utilized for a development of the system. Hence it is to be technically feasible to develop a new system.

Operational feasibility:

The proposed system offers greater level of user friend less combined with greater processing speed.

Therefore,

Float of staff can reduce. Since the processing speed is very high compared to that of manually on the management can

Take care timely actions depending on the information obtained. Since the work is also reduced, the arrangement conveys that the project is operationally feasible. Hence the stage of the system study and problem formulation places an important role in the system development life cycle.

SOFTWARE REQUIREMENT SPECIFICATIONS

A software requirements specification (SRS) is a document that describes what the software will do and how it will be expected to perform. It also describes the functionality the product needs to fulfill all stakeholders (business, users) needs.

The software requirements specification document lists sufficient and necessary requirements for the project development. To derive the requirements, the developer needs to have clear and thorough understanding of the products under development. This is achieved through detailed and continuous communications with the project team and customer throughout the software development process.

GOALS

- Facilitating reviews
- Describing the scope of work
- Providing a reference to software designers (i.e., navigation aids, document structure)
- Providing a framework for testing primary and secondary use cases
- Including features to customer requirements
- Providing a platform for ongoing refinement (via incomplete specs or questions)

System Features and Requirements

- Functional Requirements
- External Interface Requirements

FUNCTIONAL REQUIREMENTS:

General description of inputs and outputs. The system has basically a menu driven input format.

The officer has to choose from the menu, the appropriate options.

EXTERNAL INTERFACE REQUIREMENTS:

User interface:

User interface is of the most important parts of the effective software. A menu driven system is to be developed using which the user does it.

The interface should reinforce.

SOFTWARE DESIGN & REQUIREMENTS

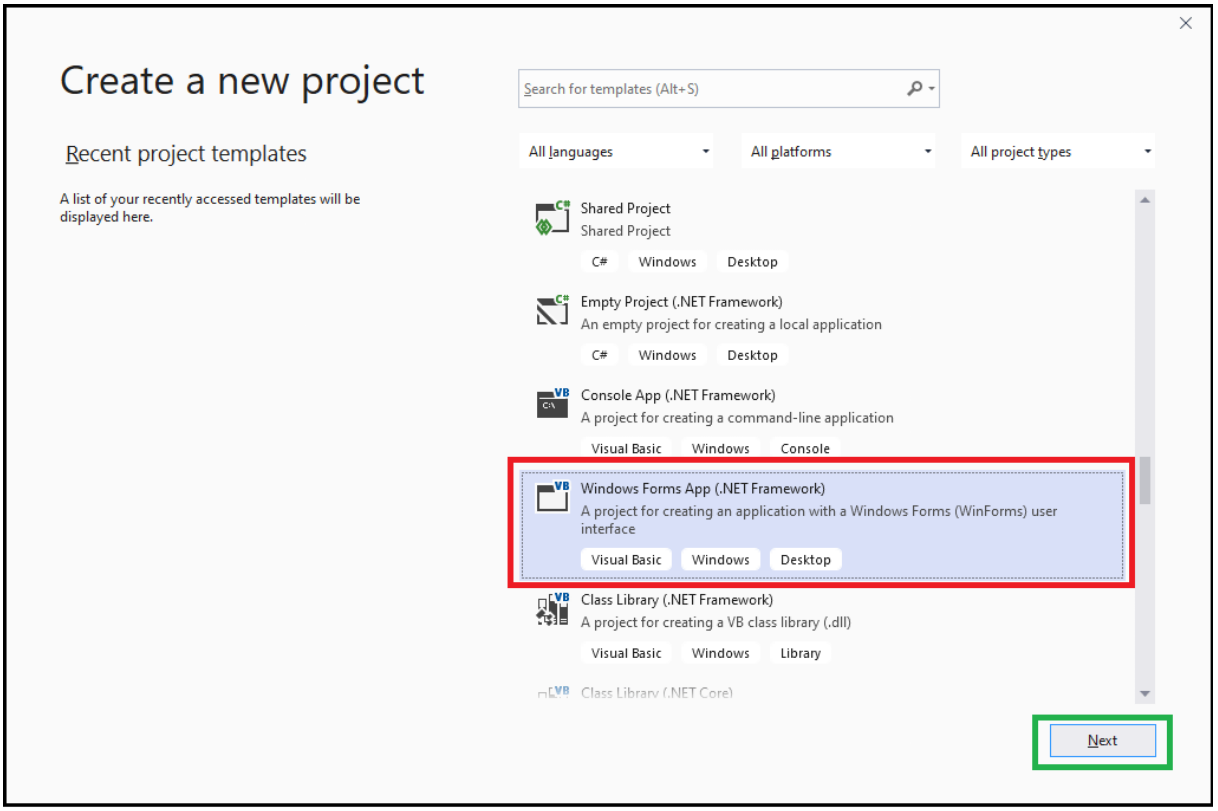
SYSTEM DESIGN AND REQUIREMENTS

Visual Basic, originally called **Visual Basic .NET** (**VB.NET**), is a multi-paradigm, object-oriented programming language, implemented on .NET, Mono, and the .NET Framework. Microsoft launched VB.NET in 2002 as the successor to its original Visual Basic language, the last version of which was Visual Basic 6.0. Although the ".NET" portion of the name was dropped in 2005, this article uses "Visual Basic [.NET]" to refer to all Visual Basic languages released since 2002, in order to distinguish between them and the classic Visual Basic. Along with C# and F#, it is one of the three main languages targeting the .NET ecosystem. As of March 11, 2020, Microsoft announced that evolution of the VB.NET language has concluded.

Microsoft's integrated development environment (IDE) for developing in Visual Basic is Visual Studio. Most Visual Studio editions are commercial; the only exceptions are Visual Studio Express and Visual Studio Community, which are freeware. In addition, the .NET Framework SDK includes a freeware command-line compiler called vbc.exe. Mono also includes a command-line VB.NET compiler.

Visual Basic is often used in conjunction with the Windows Forms GUI library to make desktop apps for Windows. Programming for Windows Forms with Visual Basic involves dragging and dropping controls on a form using a GUI designer and writing corresponding code for each control.

VISUAL BASIC WILL START AND THIS DIALOG BOX APPEARS



Some features of Visual Basic :

i. Full set of controles to draw the application.
ii. Lots of icons and pictures for usage.
iii. Response to mouse and keyboard.
iv. Clipboard and printer access.
v. Full array of mathematical, string handling and graphic functions.
vi. Can handle fixed and dynamic variable and control array.
vii. Sequential and random access files.
viii. Use full debugger and error handling facilities.
ix. Power full database access tools.
x. Package and development wizard
xi. Package and Development Wizard makes distributing your applications simple.

Versions of Visual Basic:

- The original visual basic for DOS and visual basic for windows were introduced in the year1991.
- Visual Basic 3 was released in 1993.
- Visual Basic 4 was released in 1995.
- Visual Basic 5 was released in 1996.
- Visual Basic 6.0 was released in 1998.

- **2002 (VB 7.0)**

The first version, Visual Basic .NET, relies on .NET Framework 1.0. The most important feature is managed code, which contrasts with the classic Visual Basic.

- **2003 (VB 7.1)**

Visual Basic .NET 2003 was released with .NET Framework 1.1. New features included support for the .NET Compact Framework and a better VB upgrade wizard. Improvements were also made to the performance and reliability of .NET IDE (particularly the background compiler) and runtime. In addition, Visual Basic .NET 2003 was available in the Visual Studio.NET Academic Edition, distributed to a certain number of scholars from each country without cost.

- **2005 (VB 8.0)**

After Visual Basic .NET 2003, Microsoft dropped ".NET" from the name of the product, calling the next version Visual Basic 2005.

For this release, Microsoft added many features intended to reinforce Visual Basic .NET's focus as a rapid application development platform and further differentiate it from C#., including:

- **2008 (VB 9.0)**

Visual Basic 9.0 was released along with .NET Framework 3.5 on November 19, 2007.

For this release, Microsoft added many features, including:

- **2010 (VB 10.0)**

In April 2010, Microsoft released Visual Basic 2010. Microsoft had planned to use Dynamic Language Runtime (DLR) for that release but shifted to a co-evolution strategy between Visual Basic and sister language C# to bring both languages into closer parity with one another. Visual Basic's innate ability to interact dynamically with CLR and COM objects has been enhanced to work with dynamic languages built on the DLR such as IronPython and IronRuby. The Visual Basic compiler was improved

to infer line continuation in a set of common contexts, in many cases removing the need for the " _" line continuation characters. Also, existing support of inline Functions was complemented with support for inline Subs as well as multi-line versions of both Sub and Function lambdas.

- **2012 (VB 11.0)**

Visual Basic 2012 was released alongside .NET Framework 4.5. Major features introduced in this version include.

- **2013 (VB 12.0)**

Visual Basic 2013 was released alongside .NET Framework 4.5.1 with Visual Studio 2013. Can also build .NET Framework 4.5.2 applications by installing Developer Pack.

- **2015 (VB 14.0)**

Visual Basic 2015 (code named VB "14.0") was released with Visual Studio 2015. Language features include a new "?." operator to perform inline null checks, and a new string interpolation feature is included to format strings inline.

- **2017 (VB 15.0)**

Visual Basic 2017 (code named VB "15.0") was released with Visual Studio 2017. Extends support for new Visual Basic 15 language features with revision 2017, 15.3, 15.5, 15.8. Introduces new refactorings that allow organizing source code with one action.

- **2019 (VB 16.0)**

Visual Basic 2019 (code named VB "16.0") was released with Visual Studio 2019. It is the first version of Visual Basic focused on .NET Core.

Special Features of VB.NET

- Variables can be declared using the WithEvents construct.
- An Object can be selected from the Class Name drop down list
- Property methods may take parameters and can be passed as reference
- Support for optional parameters
- Exception handling and filtering becomes logical with Catch … When … clause
- Auto-wire up of events, VB.NET has the Handles syntax for events
- Use of With … End With structure for Marshalling of objects
- Allows inline date declarations
- The My namespace as a shortcut to commonly used APIs
- Very powerful COM components and interoperability

- Importing of Namespaces at project level avoiding the need to import them for every file in the project
- Allows Definition of conditional compiler constants
- Allows late binding with Option Strict off
- The ability to implement interfaces with methods of different names
- Enums can be defined inside interfaces
- Code compilation in the background

Applications built using Visual Basic .Net will run with windows 7,8,10 And Windows11.

For now just click on open–we are starting a new project. Later, once you have saved same projects, they can be opened using the existing and recent tabs, the visual basic development environment will start.

Drawing the user interface and setting properties

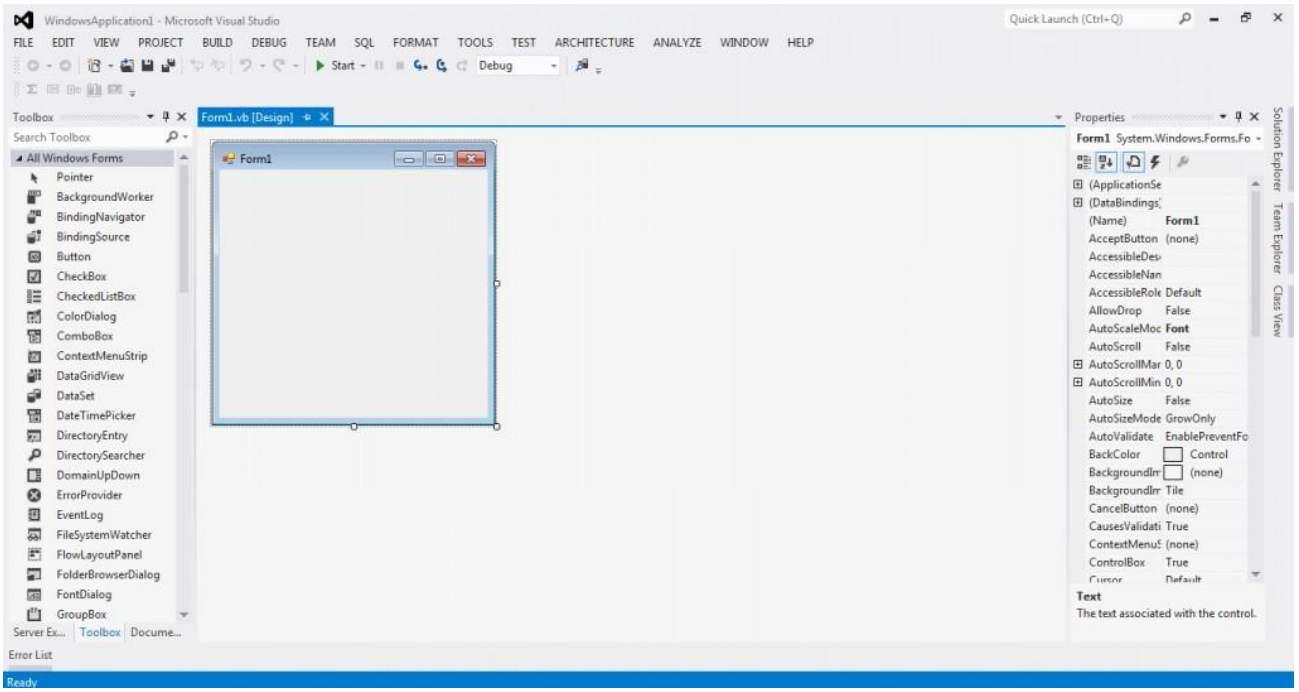
Visual basic operates in 3 modes:

- Design mode – used to build an application.
- Run mode–used to run the applications.
- Break mode– application halted and debugger is available.

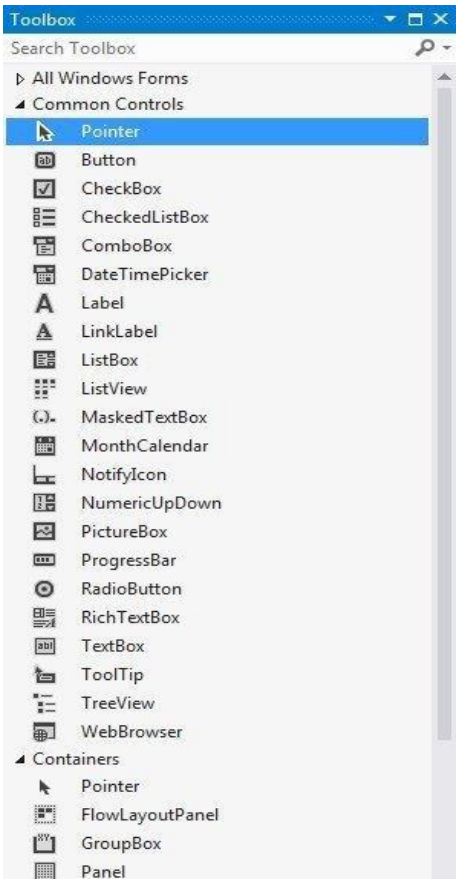
We focus on hear a design mode.

Each window can be viewed by selecting menu options, depressing functions keys or using the tool bar. Use the method you feel most comfortable with.

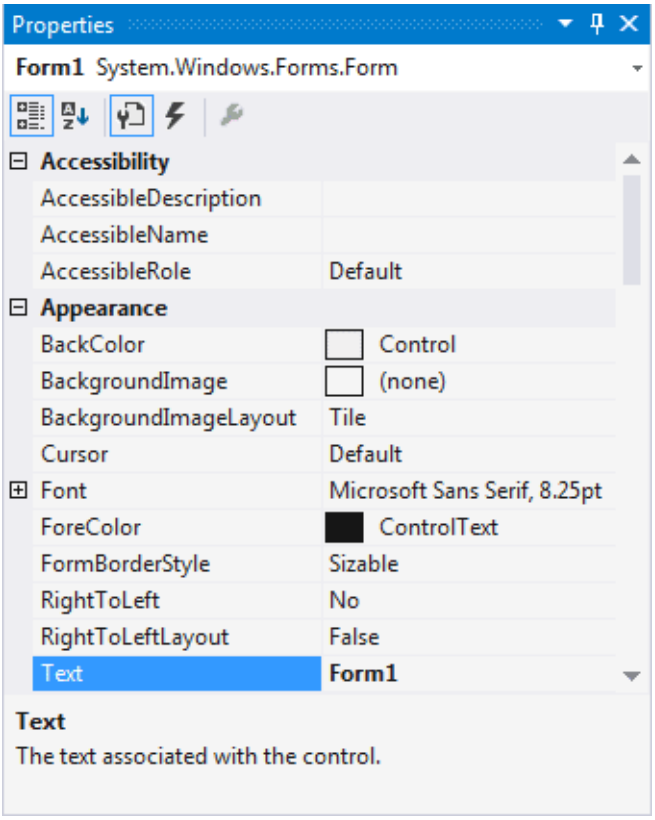
- The main window consists of the title bar and tool bar. At title bar indicates the project name, current visual basic operating mode, and the current form. The menu bar has drop down menus from which you control the operation of the environment. The tool bar has button that provides shortcuts to sum of the menu options. The main window also shows the location of the current from relative to the upper left corner of the screen and the width and length of Massmedia Adverticement Management System current form. The form window is central to developing Visual Basic application. It is where you draw your application.



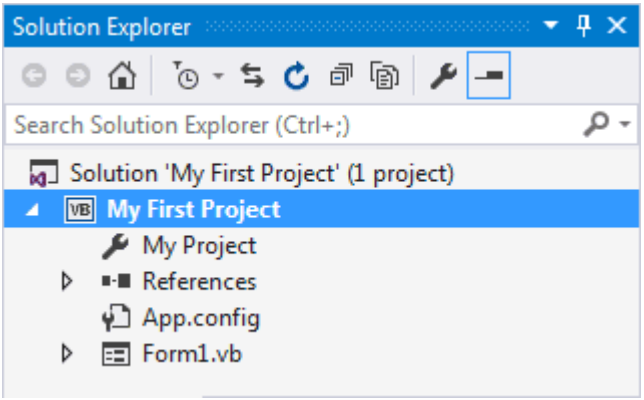
The **toolbar** is the selection menu bar controls used in your computer application help with any control is available by clicking the control. The property window is used to establish initial property values for object .The drop down box at the top of the window lists all objects in the current form. Two views are available: alphabetic properties for the currently selected object.



The Properties window is **used to display properties for objects selected** in the two main types of windows available in the Visual Studio integrated development environment (IDE). These two types of windows are: Tool windows such as Solution Explorer, Class View, and Object browser.



The Solution Explorer window contains a list of the items in the current solution. A solution can contain multiple projects, and each project can contain multiple items. The Solution Explorer displays a **hierarchical list of all the components**, organized by project. The Solution Explorer displays a list of all forms and modules making up your application. You can also obtain a view of the form or code windows form the Solution Explorer.



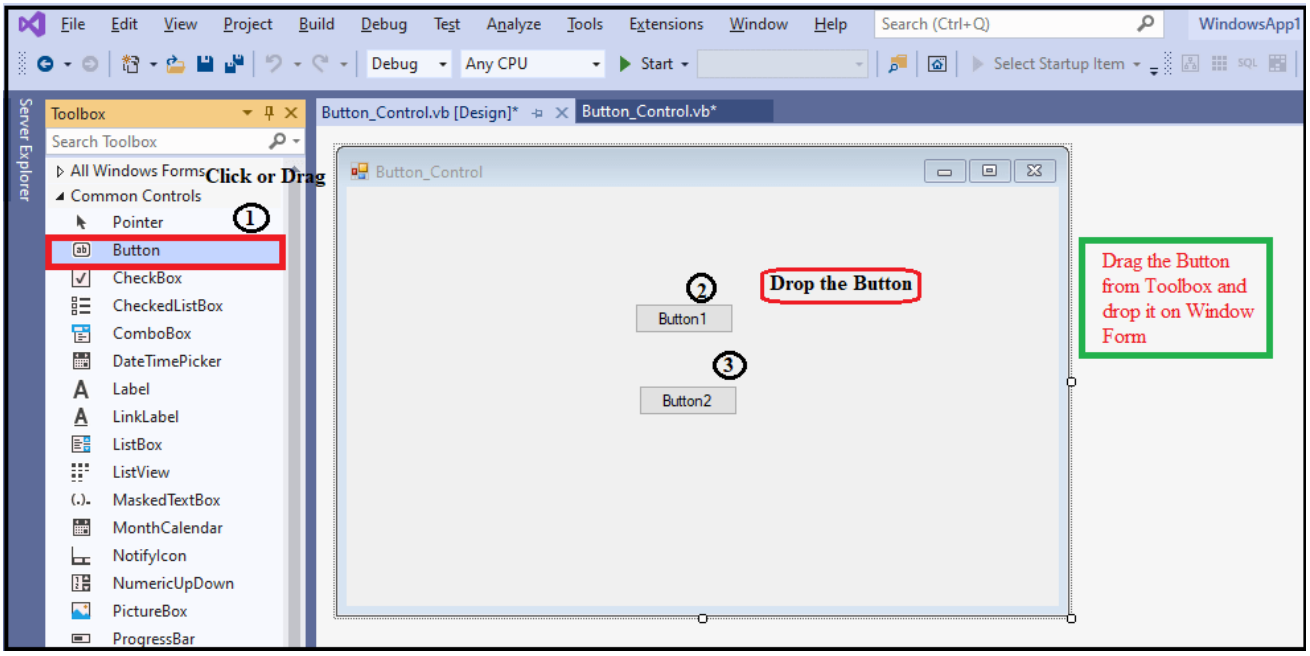
As mentioned, the user interface is drawn in the form window. There are two ways to place controls on a form.

i. Double click the tool in the toolbox and it is created with a default size on the form. You can then move it or resize.

ii. Click the tool in the tool box then move the mouse pointer to the cross hair at the upper left corner of where you want the control to be press the left mouse button and hold it down while dragging the cursor towards the lower right corner.

iii. To move a control you have drawn, click the object in the form window and drag it to the new location. Release the mouse button.

iv. To resize a control, click the objects so that is selected and sizing handles appear. Use this handle store size the objects.



Microsoft SQL Server (2019)

Introduction to Microsoft SQL Server (2019)

Microsoft SQL Server is a relational database management system developed by Microsoft. As a database server, it is a software product with the primary function of storing and retrieving data as requested by other software applications—which may run either on the same computer or on another computer across a network (including the Internet). Microsoft markets at least a dozen different editions of Microsoft SQL Server, aimed at different audiences and for workloads ranging from small single-machine applications to large Internet-facing applications with many concurrent users.

Milestones

- MS SQL Server for OS/2 began as a project to port Sybase SQL Server onto OS/2 in 1989, by Sybase, Ashton-Tate, and Microsoft.
- SQL Server 4.2 for NT is released in 1993, marking the entry onto Windows NT.
- SQL Server 6.0 is released in 1995, marking the end of collaboration with Sybase; Sybase would continue developing their own variant of *SQL Server*, Sybase Adaptive Server Enterprise, independently of Microsoft.
- SQL Server 7.0 is released in 1998, marking the conversion of the source code from C to C++.

- SQL Server 2005, released in 2005, finishes the complete revision of the old Sybase code into Microsoft code.
- SQL Server 2012, released in 2012, adds columnar in-memory storage aka xVelocity.
- SQL Server 2017, released in 2017, adds Linux support for these Linux platforms: Red Hat Enterprise Linux, SUSE Linux Enterprise Server, Ubuntu & Docker Engine.

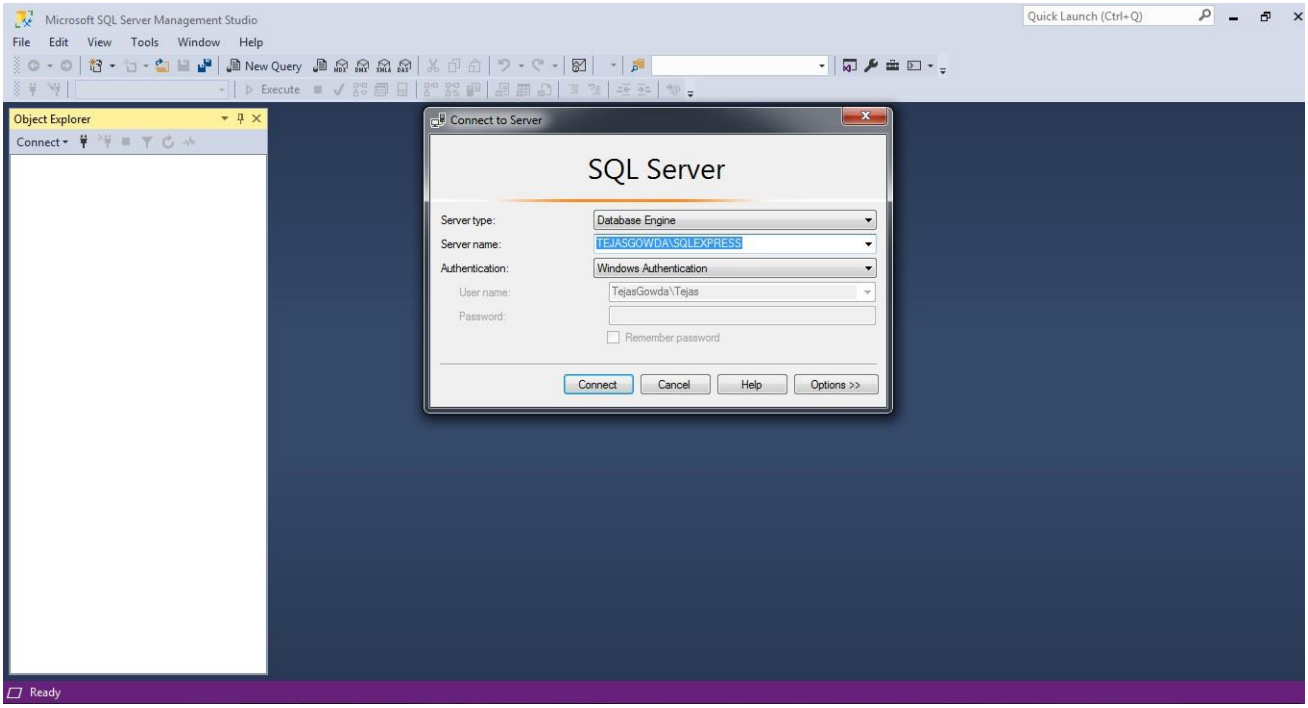
Currently

As of May 2020, the following versions are supported by Microsoft:

- SQL Server 2012
- SQL Server 2014
- SQL Server 2016
- SQL Server 2017
- SQL Server 2019

From SQL Server 2016 onward, the product is supported on x64 processors only.

The current version is Microsoft SQL Server 2019, released November 4, 2019.



Editions

- **Enterprise**

  SQL Server Enterprise Edition includes both the core database engine and add-on services, with a range of tools for creating and managing a SQL Server cluster. It can manage databases as large as 524 petabytes and address 12 terabytes of memory and supports 640 logical processors (CPU cores).

- **Standard**

  SQL Server Standard edition includes the core database engine, along with the stand-alone services. It differs from Enterprise edition in that it supports fewer active instances (number of nodes in a cluster) and does not include some high-availability functions such as hot-add memory (allowing memory to be added while the server is still running), and parallel indexes.

- **Web**

  SQL Server Web Edition is a low-TCO option for Web hosting.

- **Business Intelligence**

  Introduced in SQL Server 2012 and focusing on Self Service and Corporate Business Intelligence. It includes the Standard Edition capabilities and Business Intelligence tools: PowerPivot, Power View, the BI Semantic Model, Master Data Services, Data Quality Services and xVelocity in-memory analytics.

- **Workgroup**

  SQL Server Workgroup Edition includes the core database functionality but does not include the additional services. Note that this edition has been retired in SQL Server 2012.

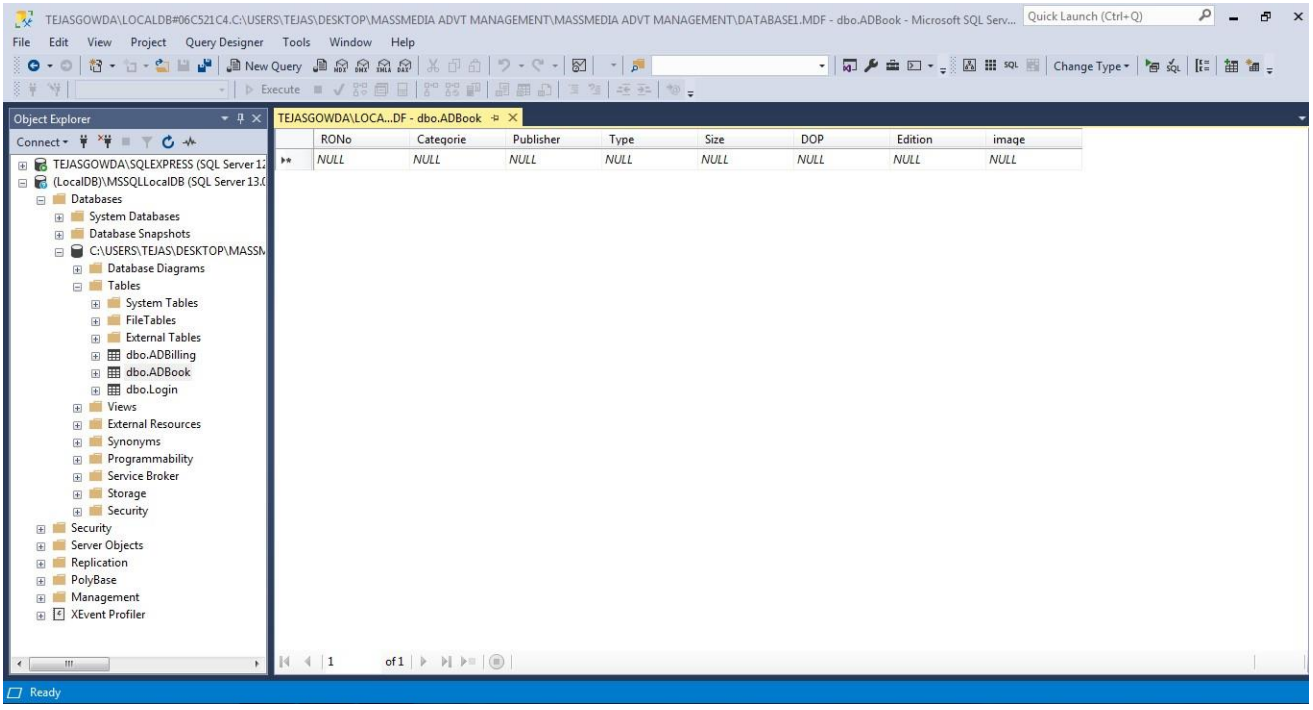
- **Express**

  SQL Server Express Edition is a scaled down, free edition of SQL Server, which includes the core database engine. While there are no limitations on the number of databases or users supported, it is limited to using one processor, 1 GB memory and 10 GB database files (4 GB database files prior to SQL Server Express 2008 R2). It is intended as a replacement for MSDE. Two additional editions provide a superset of features not in the original Express Edition. The first is *SQL Server Express with Tools*, which includes SQL Server Management

Studio Basic. *SQL Server Express with Advanced Services* adds full-text search capability and reporting services.

Architecture

The protocol layer implements the external interface to SQL Server. All operations that can be invoked on SQL Server are communicated to it via a Microsoft-defined format, called Tabular Data Stream (TDS). TDS is an application layer protocol, used to transfer data between a database server and a client. Initially designed and developed by Sybase Inc. for their Sybase SQL Server relational database engine in 1984, and later by Microsoft in Microsoft SQL Server, TDS packets can be encased in other physical transport dependent protocols, including TCP/IP, named pipes, and shared memory. Consequently, access to SQL Server is available over these protocols. In addition, the SQL Server API is also exposed over web services.



Visual Studio

Microsoft Visual Studio includes native support for data programming with Microsoft SQL Server. It can be used to write and debug code to be executed by SQL CLR. It also includes a data designer that can be used to graphically create, view or edit database schemas. Queries can be created either visually or using code. SSMS 2008 onwards, provides intellisense for SQL queries as well.

SQL Server Management Studio

SQL Server Management Studio is a GUI tool included with SQL Server 2005 and later for configuring, managing, and administering all components within Microsoft SQL Server. The tool includes both script editors and graphical tools that work with objects and features of the server. SQL Server Management Studio replaces Enterprise Manager as the primary management interface for Microsoft SQL Server since SQL Server 2005. A version of SQL Server Management Studio is also available for SQL Server Express Edition, for which it is known as SQL Server Management Studio Express (SSMSE).

A central feature of SQL Server Management Studio is the Object Explorer, which allows the user to browse, select, and act upon any of the objects within the server. It can be used to visually observe and analyze query plans and optimize the database performance, among others. SQL Server Management Studio can also be used to create a new database, alter any existing database schema by adding or modifying tables and indexes, or analyze performance. It includes the query windows which provide a GUI based interface to write and execute queries.

View database in Server Explorer

1. On the View menu, click Server Explorer.
2. Right-click Data Connections, and then click Add connection.
3. In the Data Link Properties dialog box, click localhost in the Select or enter a server name box.
4. Click Windows NT Integrated Security to log on to the server.
5. Click Select the database on the server, and then select *Northwind* database from the list.
6. Click Test Connection to validate the connection, and then click OK.
7. In the Server Explorer, click to expand the Data Connections tree so that the Employees table node expands. The properties of individual fields appear in the Properties window.

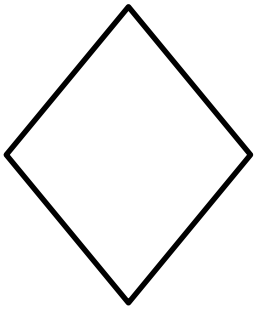
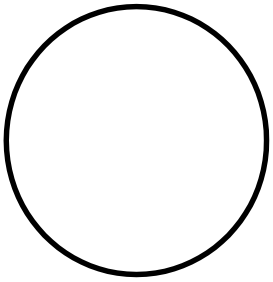
ER DIAGRAM

ER DIAGRAM

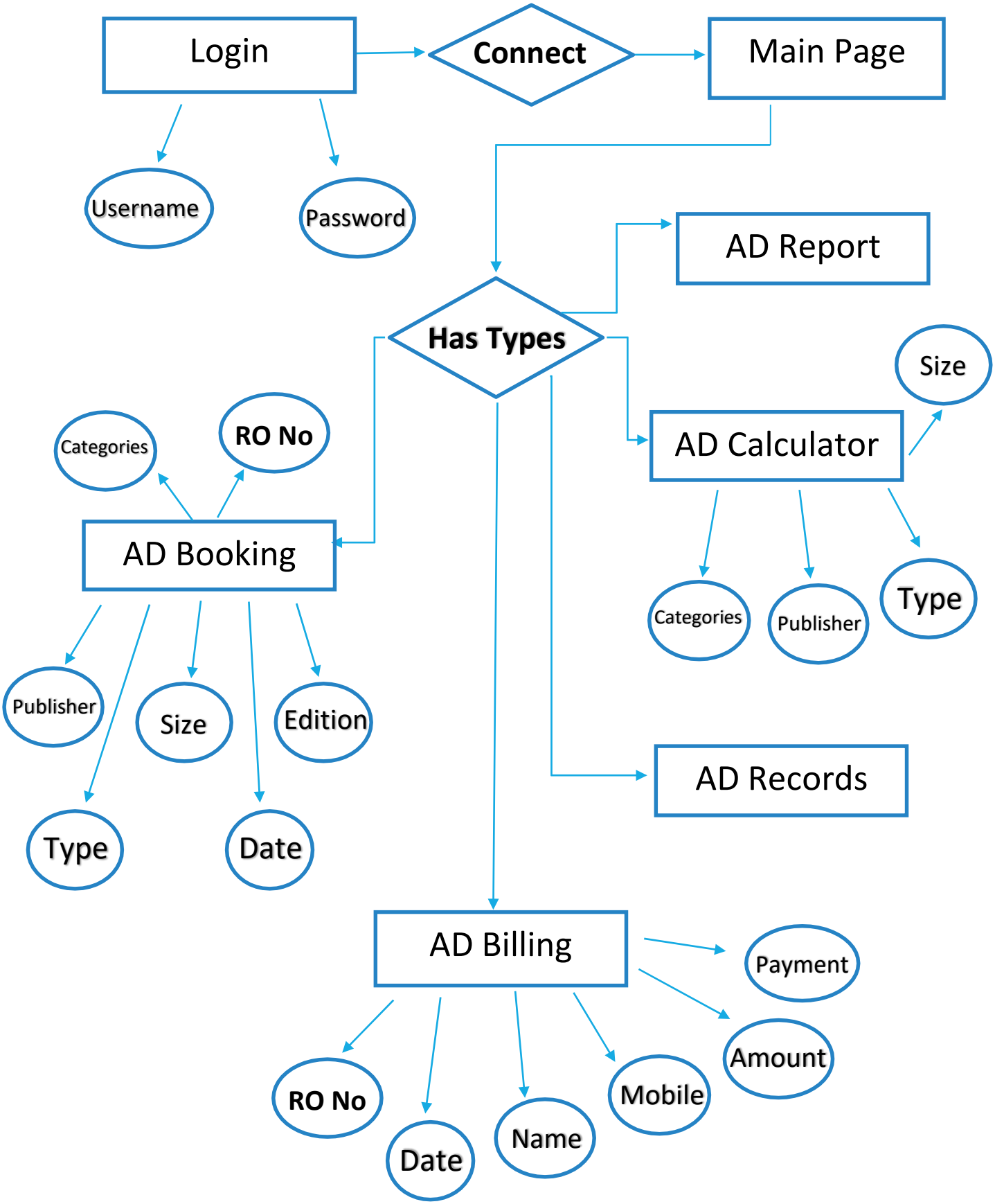
The overall logical structure of a database can be expressed graphically by an **ER DIAGRAM.**

THE FOLLOWING ARE USED TO DRAW AN ER DIAGRAM

RECTANGLE: This represents entity sets.
ELLIPSES: This represents attributes.
DIAMOND: this represents relationship sets.
LINES: which link attribute to entity sets and entity sets to relationship sets.
DOUBLE ELLIPSE: which represents multi-valued attributes.
DASHED ELLIPSE: which represents derived attributes

SYMBOLS	MEANING
	ENTITY
	RELATIONSHIP
	ATTRIBUTES

ER DIAGRAM

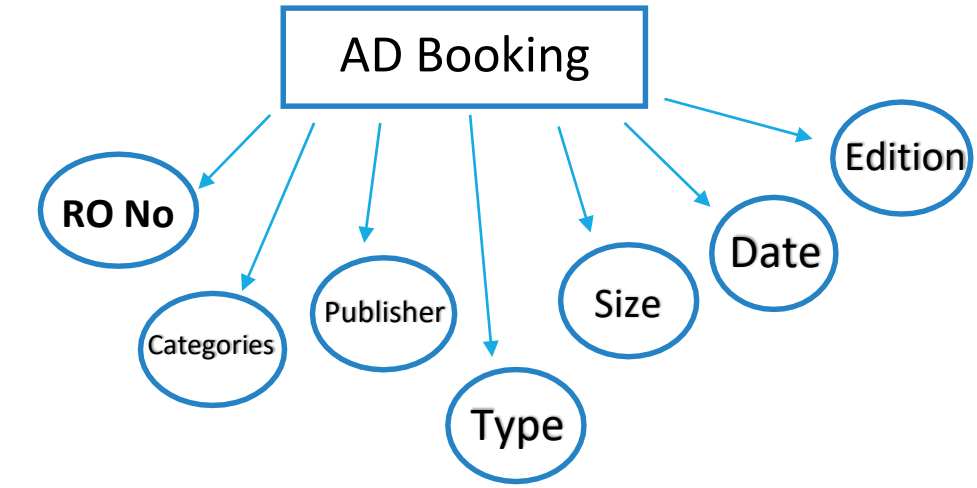

Login
Connect
Main Page
Username
Password
AD Report
Has Types
Size
AD Calculator
Categories
RO No
AD Booking
Categories
Publisher
Type
Publisher
Size
Edition
Type
Date
AD Records
AD Billing
Payment
Amount
RO No
Date
Name
Mobile

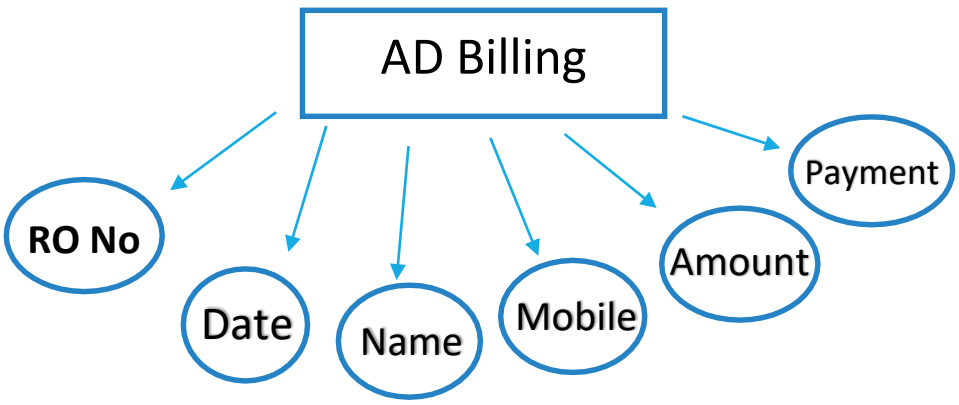
DATA FLOW DIAGRAM

Level 0:

Level 1:

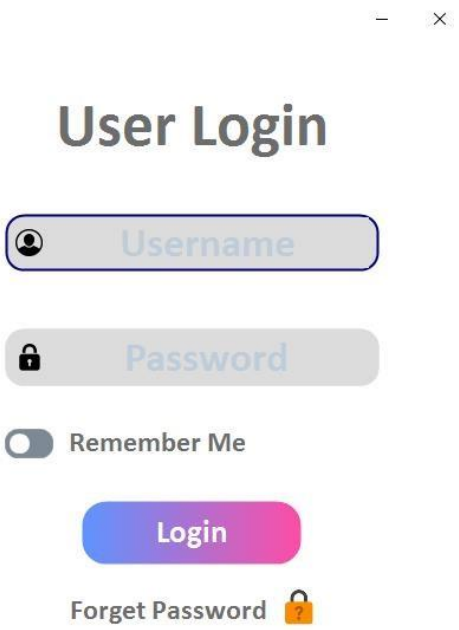
Level 3:





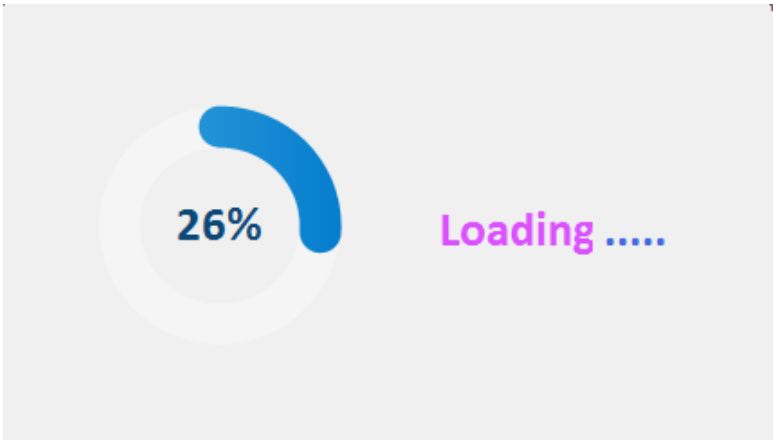
SNAPSHOTS

Login Form



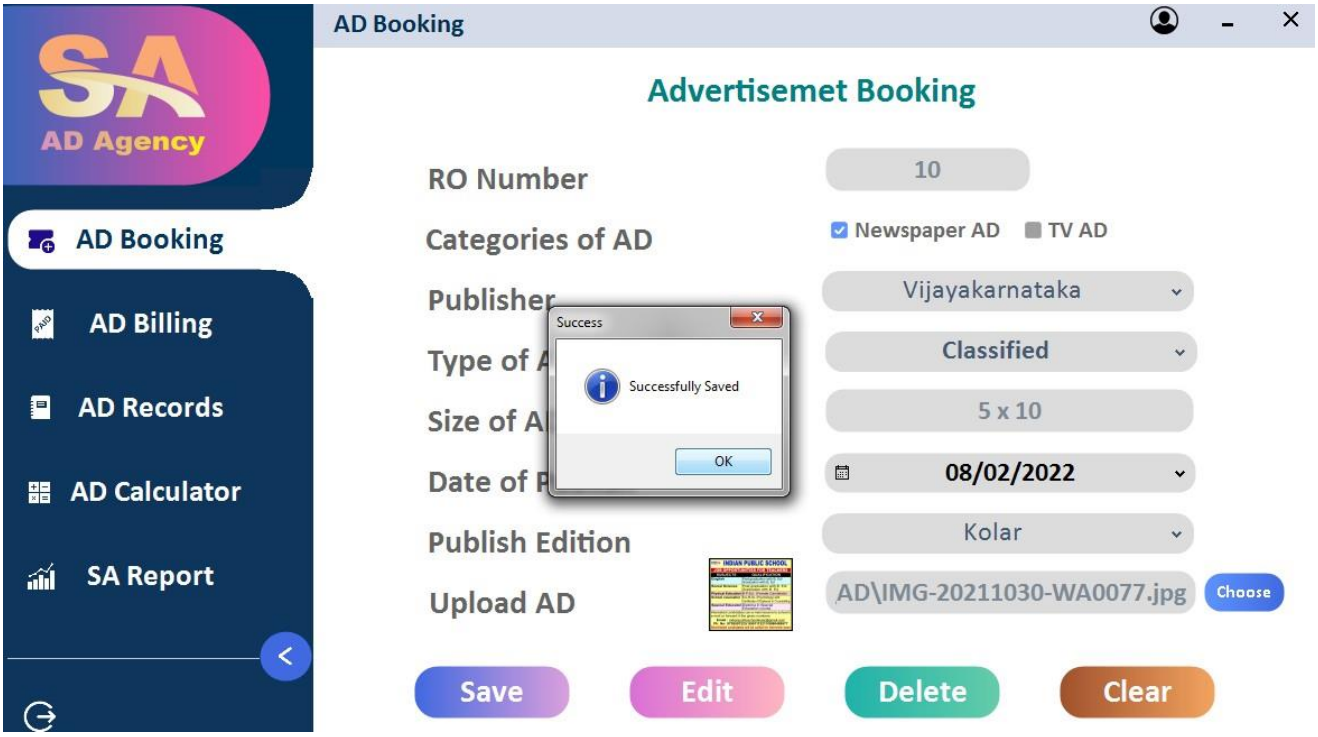
Change Password

Processing Form

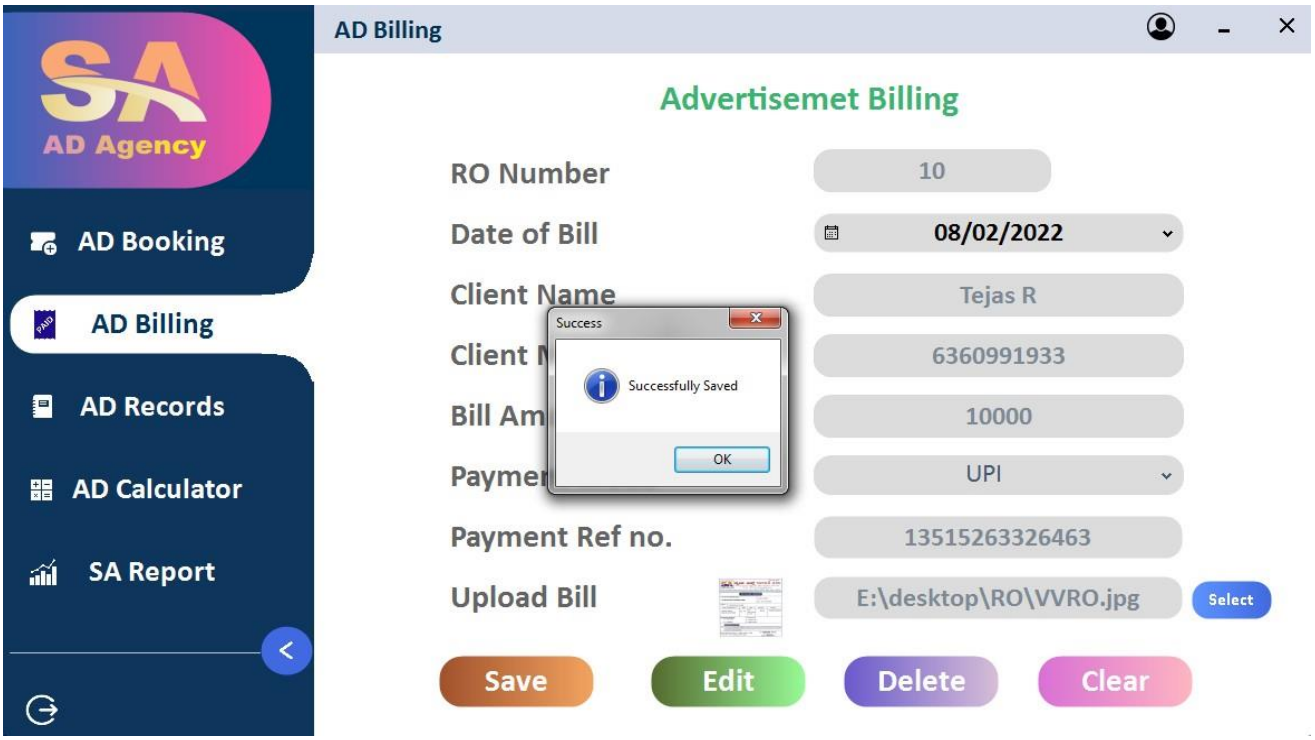


Main Page

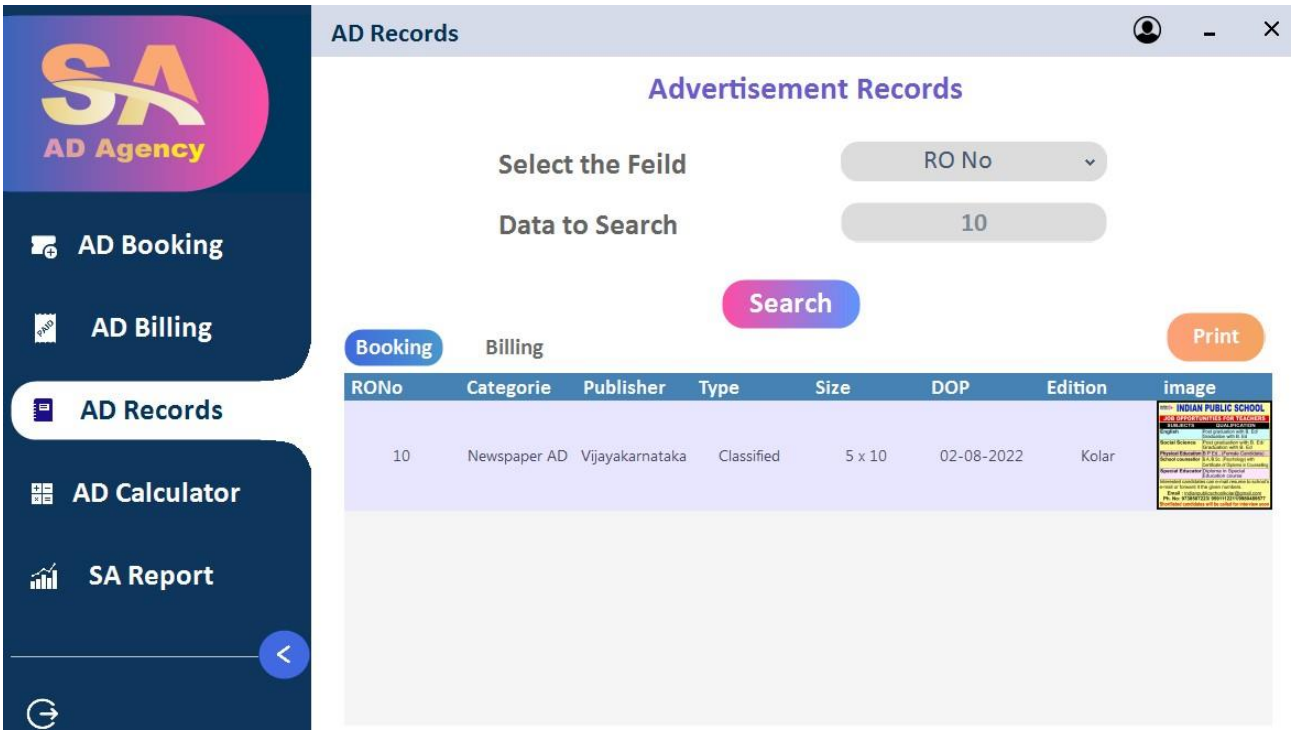
AD Booking



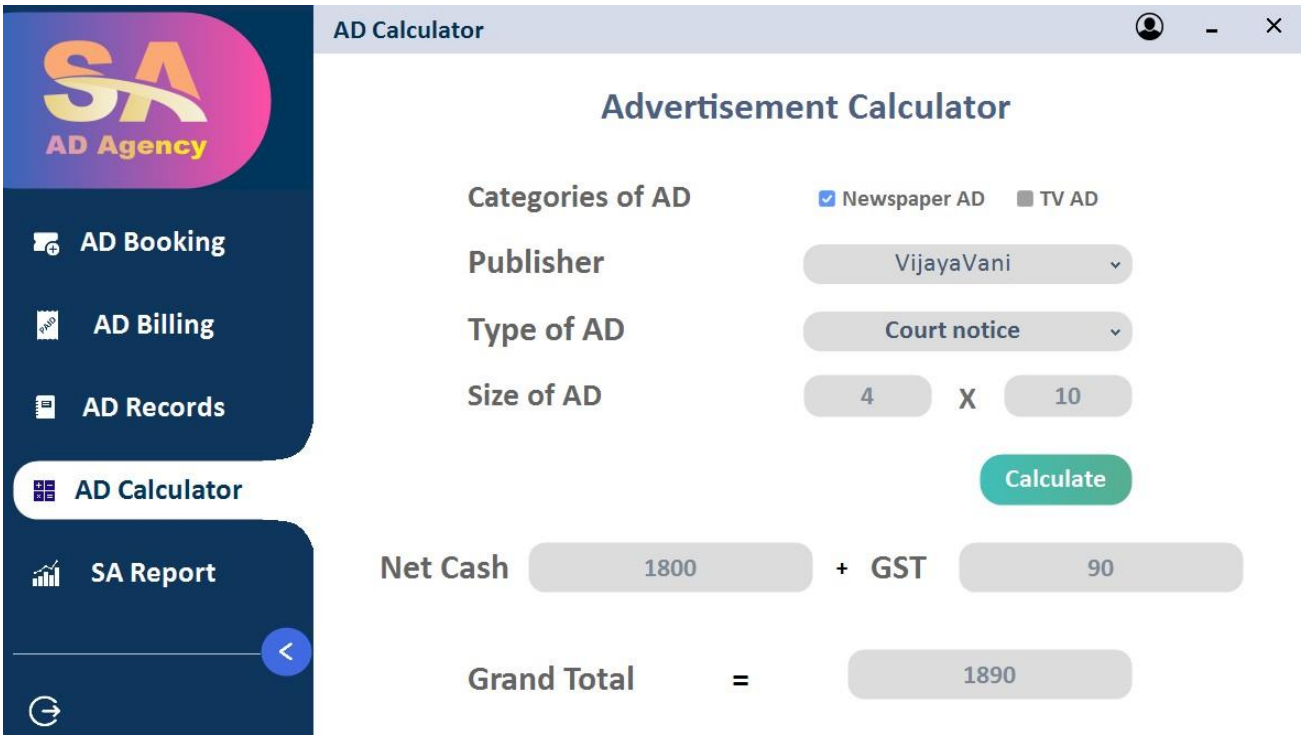
AD Billing



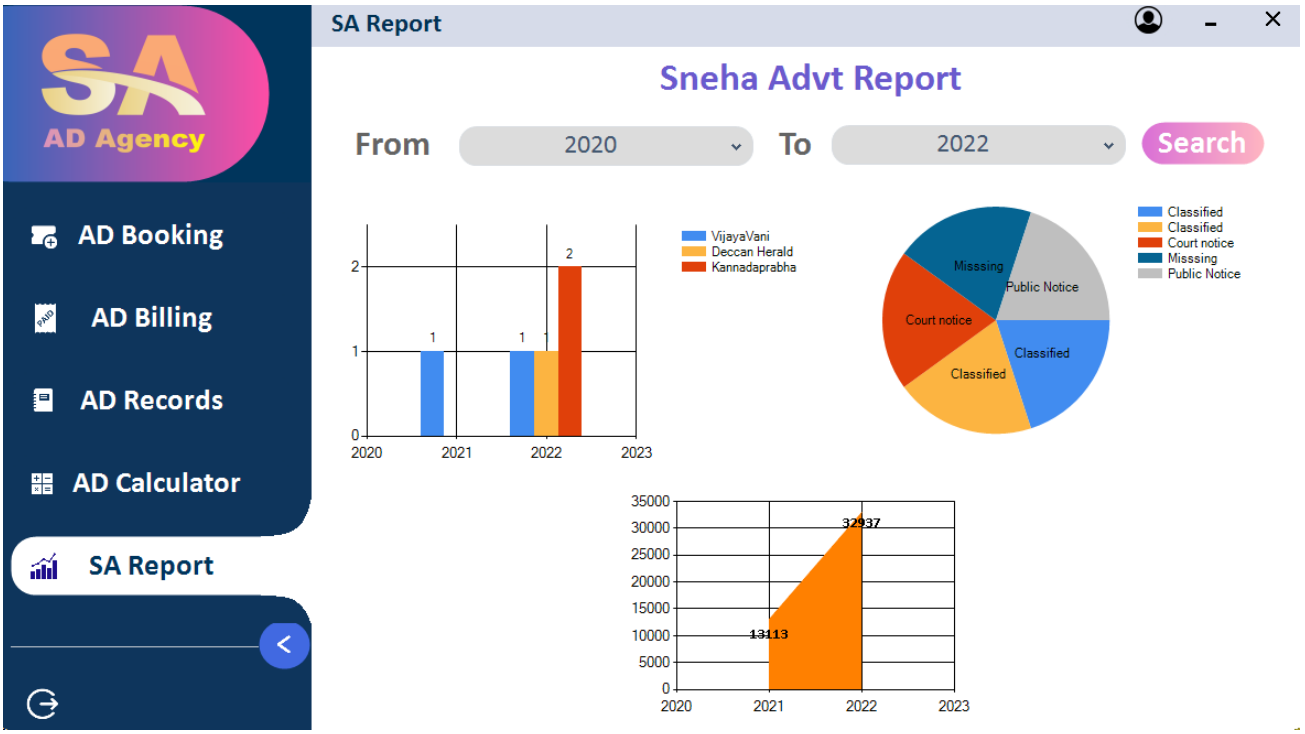
AD Records



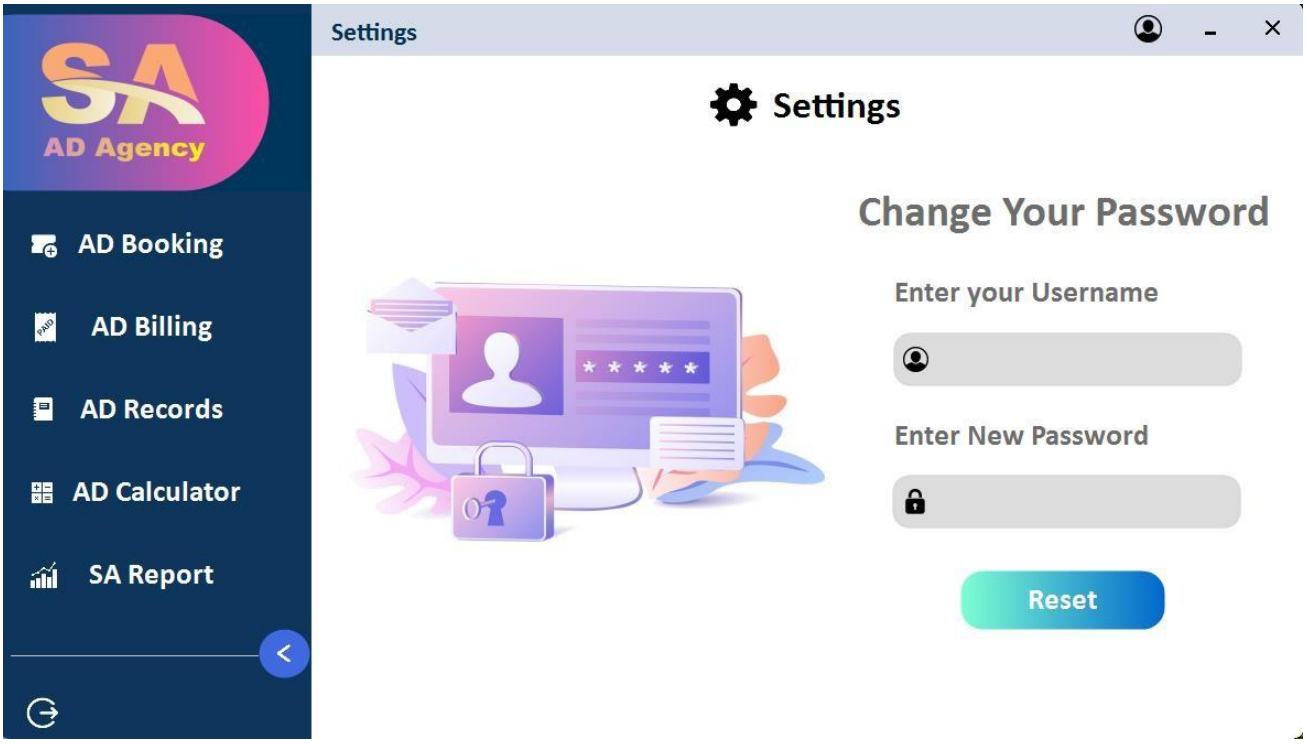
AD Calculator



SA Report



SA Settings



CODING

LOGIN FORM

```
Imports System.Data
Imports System.Data.SqlClient
Public Class LoginForm
    Dim con As SqlConnection = New SqlConnection("Data
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\Tejas\Documents\Massm
edia Advt Management\Massmedia Advt Management\MyDatabase.mdf;Integrated
Security=True")
    Private Sub LoginForm_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        LoginPanel.Visible = True
        Forgetpanel.Visible = False
        TextBox1.PlaceholderText = "Username"
        TextBox2.PlaceholderText = "Password"
        TextBox3.PlaceholderText = "Username"
        TextBox4.PlaceholderText = "New Password"
        If My.Settings.check = True Then
            TextBox1.Text = My.Settings.username
            TextBox2.Text = My.Settings.password
        Else
            'do nothing
        End If
    End Sub
    Private Sub ForgetButton_Click(sender As Object, e As EventArgs) Handles
ForgetButton.Click
        LoginPanel.Visible = False
        Forgetpanel.Visible = True
    End Sub
    Private Sub Backtologin_Click(sender As Object, e As EventArgs) Handles
Backtologin.Click
        Forgetpanel.Visible = False
        LoginPanel.Visible = True
    End Sub
    Private Sub LoginButton_Click(sender As Object, e As EventArgs) Handles
LoginButton.Click
        If Checkbox.Checked = True Then
            My.Settings.check = True
        Else
            My.Settings.check = False
        End If
        If Checkbox.Checked = True Then
            My.Settings.username = TextBox1.Text
            My.Settings.password = TextBox2.Text
        Else
```

```
            My.Settings.username = ""
            My.Settings.password = ""
        End If
        Dim cmd As SqlCommand = New SqlCommand("select * from login where
Username='" + TextBox1.Text + "'and Password='" + TextBox2.Text + "'", con)
        Dim sda As SqlDataAdapter = New SqlDataAdapter(cmd)
        Dim dt As DataTable = New DataTable()
        sda.Fill(dt)
        If (dt.Rows.Count > 0) Then
            Me.Hide()
            loginsuccess.Show()
        Else
            Me.Hide()
            LoginFailed.Show()
        End If
    End Sub
    Private Sub Forgetsubmit_Click(sender As Object, e As EventArgs) Handles
Forgetsubmit.Click
        If TextBox3.Text = "" Or TextBox4.Text = "" Then
            MsgBox("Enter Valid Data")
        Else
            Dim cmd4 As SqlCommand = New SqlCommand("Select * From login where
Username='" + TextBox3.Text + "'", con)
            Dim myreader3 As SqlDataReader
        con.Open()
        myreader3 = cmd4.ExecuteReader()
            If (myreader3.Read() = False) Then
                MessageBox.Show("Username not exist", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Information)
            Else
                con.Close()
                If MessageBox.Show("Are you sure need to Reset", "Confirmation",
MessageBoxButtons.YesNo, MessageBoxIcon.Information) Then
                    Dim cmd6 As SqlCommand = New SqlCommand("UPDATE login set
Password='" + TextBox4.Text + "' WHERE Username='" + TextBox3.Text + "'", con)
                    con.Open()
                    cmd6.ExecuteNonQuery()
                    MessageBox.Show("Password Changed Successfully ", "Success",
MessageBoxButtons.OK, MessageBoxIcon.Information)
                    con.Close()
                    TextBox3.Text = ""
                    TextBox4.Text = ""
                End If
```

```
        End If
      End If
    End Sub
End Class
```

Processing Form

```
Imports System.ComponentModel
Public Class loginsuccess
    Dim len As Integer
    Dim counter As Integer
    Dim txt As String
    Private Sub ForgetButton_Click(sender As Object, e As EventArgs) Handles
ForgetButton.Click
        Me.Hide()
        MainPage.Show()
    End Sub
    Private Sub Guna2ControlBox1_Click(sender As Object, e As EventArgs) Handles
Guna2ControlBox1.Click
        Me.Hide()
        LoginForm.Show()
    End Sub
    Private Sub Panel2_Paint(sender As Object, e As PaintEventArgs) Handles Panel2.Paint
        If CircleProgressBar1.Value >= 100 Then
            Panel2.Hide()
        End If
    End Sub
    Private Sub Timer1_Tick(sender As Object, e As EventArgs) Handles Timer1.Tick
        CircleProgressBar1.Increment(1)
        counter += 1
        If counter > len Then
            counter = 0
            Label3.Text = ""
        Else
            Label3.Text = txt.Substring(0, counter)
            If Label3.ForeColor = Color.Navy Then
                Label3.ForeColor = Color.Aquamarine
            Else
                Label3.ForeColor = Color.RoyalBlue
            End If
        End If
    End Sub
    Private Sub loginsuccess_Load(sender As Object, e As EventArgs) Handles MyBase.Load
```

```
        counter = 0
        len = Label3.Text.Length
        txt = Label3.Text
        Timer1.Start()
    End Sub
End Class
```

Main Page

```
Imports System.Data.SqlClient
Public Class MainPage
    Dim len As Integer
    Dim counter As Integer
    Dim txt As String
    Dim con As SqlConnection = New SqlConnection("Data
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\Tejas\Documents\Massm
edia Advt Management\Massmedia Advt Management\MyDatabase.mdf;Integrated
Security=True")
    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        Panel2.Controls.Add(AdBooking1)
        AdBooking1.BringToFront()
        AdBooking1.Show()
        Button1.Checked = True
        slider.Location = New Point(Button1.Location.X + 226, Button1.Location.Y - 37)
        slider.Visible = True
        slider.BringToFront()
        If Button1.Checked = True Then
            Label1.Text = Button1.Text
        End If
    End Sub
    Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
        Panel2.Controls.Add(AdBilling1)
        AdBilling1.BringToFront()
        AdBilling1.Show()
        Button2.Checked = True
        slider.Location = New Point(Button2.Location.X + 226, Button2.Location.Y - 37)
        slider.Visible = True
        slider.BringToFront()
        If Button2.Checked = True Then
            Label1.Text = Button2.Text
        End If
    End Sub
    Private Sub Button3_Click(sender As Object, e As EventArgs) Handles Button3.Click
```

```
        Panel2.Controls.Add(AdRecords1)
        AdRecords1.BringToFront()
        AdRecords1.Show()
        Button3.Checked = True
        slider.Location = New Point(Button3.Location.X + 227, Button3.Location.Y - 37)
        slider.Visible = True
        slider.BringToFront()
        If Button3.Checked = True Then
            Label1.Text = Button3.Text
        End If
    End Sub
    Private Sub Button4_Click(sender As Object, e As EventArgs) Handles Button4.Click
        Panel2.Controls.Add(AdCalculator1)
        AdCalculator1.BringToFront()
        AdCalculator1.Show()
        Button4.Checked = True
        slider.Location = New Point(Button4.Location.X + 227, Button4.Location.Y - 37)
        slider.Visible = True
        slider.BringToFront()
        If Button4.Checked = True Then
            Label1.Text = Button4.Text
        End If
    End Sub
    Private Sub Button5_Click(sender As Object, e As EventArgs) Handles Button5.Click
        Panel2.Controls.Add(SaReport1)
        SaReport1.BringToFront()
        SaReport1.Show()
        Button5.Checked = True
        slider.Location = New Point(Button5.Location.X + 227, Button5.Location.Y - 37)
        slider.Visible = True
        slider.BringToFront()
        If Button5.Checked = True Then
            Label1.Text = Button5.Text
        End If
    End Sub
    Private Sub Guna2PictureBox1_Click(sender As Object, e As EventArgs) Handles
Guna2PictureBox1.Click
        Panel2.Controls.Add(SaMainpage1)
        SaMainpage1.BringToFront()
        SaMainpage1.Show()
        slider.Visible = False
        slider.Hide()
        Button1.Checked = False
```

```
        Button2.Checked = False
        Button3.Checked = False
        Button4.Checked = False
        Button5.Checked = False
        Label1.Text = "Sneha Adverticers MainPage"
    End Sub
    Private Sub MainPage_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        Panel2.Controls.Add(SaMainpage1)
        SaMainpage1.BringToFront()
        SaMainpage1.Show()
        Guna2CircleButton2.Visible = False
        Guna2PictureBox2.Visible = False
        counter = 0
        len = Label8.Text.Length
        txt = Label8.Text
        Timer1.Start()
    End Sub
   Private Sub PictureBox1_Click(sender As Object, e As EventArgs) Handles
PictureBox1.Click
        Application.Restart()
    End Sub
    Private Sub Guna2ControlBox1_Click(sender As Object, e As EventArgs) Handles
Guna2ControlBox1.Click
        Application.Exit()
    End Sub
    Private Sub Guna2GradientTileButton1_Click(sender As Object, e As EventArgs) Handles
Guna2GradientTileButton1.Click
        Panel2.Controls.Add(SaMainpage1)
        SaMainpage1.BringToFront()
        SaMainpage1.Show()
        Label1.Text = "Sneha Adverticers MainPage"
        Button1.Checked = False
        Button2.Checked = False
        Button3.Checked = False
        Button4.Checked = False
        Button5.Checked = False
        slider.Visible = False
        slider.Hide()
    End Sub
    Private Sub Guna2CircleButton1_Click(sender As Object, e As EventArgs) Handles
Guna2CircleButton1.Click
        Panel1.Width = 68
        Guna2CircleButton2.Visible = True
```

```
        Guna2PictureBox1.Visible = False
        Guna2PictureBox2.Visible = True
        Guna2Transition1.ShowSync(Panel1)
    End Sub
    Private Sub Guna2CircleButton2_Click(sender As Object, e As EventArgs) Handles
Guna2CircleButton2.Click
        Panel1.Width = 285
        Guna2CircleButton2.Visible = False
        Guna2PictureBox1.Visible = True
        Guna2PictureBox2.Visible = False
        Guna2Transition1.ShowSync(Panel1)
    End Sub
    Private Sub Guna2GradientCircleButton1_Click(sender As Object, e As EventArgs)
Handles Guna2GradientCircleButton1.Click
        Userpanel.BringToFront()
    End Sub
    Private Sub Guna2GradientCircleButton1_MouseDoubleClick(sender As Object, e As
MouseEventArgs) Handles Guna2GradientCircleButton1.MouseDoubleClick
        Userpanel.SendToBack()
    End Sub
    Private Sub Guna2GradientButton3_Click(sender As Object, e As EventArgs) Handles
Guna2GradientButton3.Click
        Application.Restart()
    End Sub
    Private Sub Guna2GradientButton4_Click(sender As Object, e As EventArgs) Handles
Guna2GradientButton4.Click
        Guna2Panel1.BringToFront()
        Label1.Text = "Settings"
    End Sub
    Private Sub Forgetsubmit_Click(sender As Object, e As EventArgs) Handles
Forgetsubmit.Click
        If TextBox3.Text = "" Or TextBox4.Text = "" Then
            MsgBox("Enter Valid Data")
        Else
            Dim cmd4 As SqlCommand = New SqlCommand("Select * From login where
Username='" + TextBox3.Text + "'", con)
            Dim myreader3 As SqlDataReader
            con.Open()
            myreader3 = cmd4.ExecuteReader()
            If (myreader3.Read() = False) Then
                MessageBox.Show("Username not exist", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Information)
            Else
```

```
            con.Close()
            If MessageBox.Show("Are you sure need to Reset", "Confirmation",
MessageBoxButtons.YesNo, MessageBoxIcon.Information) Then
                Dim cmd6 As SqlCommand = New SqlCommand("UPDATE login set
Password='" + TextBox4.Text + "' WHERE Username='" + TextBox3.Text + "'", con)
                con.Open()
                cmd6.ExecuteNonQuery()
                MessageBox.Show("Password Changed Successfully ", "Success",
MessageBoxButtons.OK, MessageBoxIcon.Information)
                con.Close()
                TextBox3.Text = ""
                TextBox4.Text = ""
            End If
        End If
    End If
  End Sub
  Private Sub Timer1_Tick(sender As Object, e As EventArgs) Handles Timer1.Tick
    counter += 1
    If counter > len Then
      counter = 0
      Label8.Text = ""
    Else
      Label8.Text = txt.Substring(0, counter)
      If Label8.ForeColor = Color.Navy Then
        Label8.ForeColor = Color.Aquamarine
      Else
        Label8.ForeColor = Color.RoyalBlue
      End If
   End If
  End Class
```

AD Booking

```
Public Class ADBooking
  Dim categ As String
  Dim con As SqlConnection = New SqlConnection("Data
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\Tejas\Documents\Massm
edia Advt Management\Massmedia Advt Management\MyDatabase.mdf;Integrated
Security=True")

Private Sub Guna2GradientButton1_Click(sender As Object, e As EventArgs) Handles
Guna2GradientButton1.Click
```

```
        If TextBox1.Text = "" Then
            MessageBox.Show("RO No required!!", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Information)
            TextBox1.BorderColor = Color.Red
            TextBox1.BorderThickness  =  1
            Exit Sub
        ElseIf ComboBox1.SelectedItem = "" Then
            MessageBox.Show("Publisher required!!", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Information)
            Exit Sub
        ElseIf ComboBox2.SelectedItem = "" Then
            MessageBox.Show("Type of AD required !!", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Information)
            Exit Sub
        ElseIf TextBox2.Text = "" Then
            MessageBox.Show("Enter size of AD", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Information)
            Exit Sub
        ElseIf ComboBox3.SelectedItem = "" Then
            MessageBox.Show("Publisher Edition required!!", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Information)
            Exit Sub
        ElseIf TextBox4.Text = "" Then
            MessageBox.Show("Need to Upload AD !!", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Information)
            Exit Sub
        Else
            Dim cmd2 As SqlCommand = New SqlCommand("Select * From ADBook where
RONo='" + TextBox1.Text + "'", con)
            Dim myreader As SqlDataReader
            con.Close()
            con.Open()
            myreader = cmd2.ExecuteReader()
            If (myreader.Read()) Then
                MessageBox.Show("RO Number already exist", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Information)
```

```
        Else
            con.Close()
            Try
                If MessageBox.Show("Are you sure need to Save", "Confirmation",
MessageBoxButtons.YesNo, MessageBoxIcon.Information) = DialogResult.Yes Then
                    Dim cmd As SqlCommand = New SqlCommand("insert into ADBook values
      ('" + TextBox1.Text + "','" + categ + "','" + ComboBox1.SelectedItem.ToString() + "','" +
ComboBox2.SelectedItem.ToString() + "','" + TextBox2.Text + "','" +
Guna2DateTimePicker1.Value + "','" + ComboBox3.SelectedItem.ToString() + "',@image)",
con)
                    Dim mstr As New MemoryStream
                    PictureBox1.Image.Save(mstr, PictureBox1.Image.RawFormat)
                    cmd.Parameters.Add("@image", SqlDbType.Image).Value = mstr.ToArray
                    con.Open()
                    cmd.ExecuteNonQuery()
                    MessageBox.Show("Successfully Saved", "Success", MessageBoxButtons.OK,
MessageBoxIcon.Information)
                    con.Close()
                    'Dim result As String
                    'Dim cmd1 = New SqlCommand("Select MAX(RONo) from ADBook", con)
                    'con.Open()
                    'result = cmd1.ExecuteScalar().ToString() + 1
                    'TextBox1.Text = result
                    'con.Close()
                    TextBox2.Clear()
                    TextBox4.Clear()
                    CheckBox1.Checked = False
                    CheckBox1.Checked = False
                    ComboBox1.SelectedItem = Nothing
                    ComboBox2.SelectedItem = Nothing
                    PictureBox1.Image = Nothing
                    ComboBox3.SelectedItem = Nothing
                Else
                    'do nothing
                End If
            Catch ex As Exception
```

```
                MsgBox(ex.Message)
            End Try
            End If
        End If
    End Sub

    Private Sub CheckBox1_CheckedChanged(sender As Object, e As EventArgs) Handles
CheckBox1.CheckedChanged
        If CheckBox1.Checked = True Then
            ComboBox1.Items.AddRange(New String() {"Vijayakarnataka", "VijayaVani",
"Kannadaprabha", "HosaDigantha", "Prajavani", "Udayavani", "Kolarapatrike", "Indian
Express", "Deccan Herald"})
            ComboBox2.Items.AddRange(New String() {"Classified", "Court notice", "Public
Notice", "Govt AD", "Wishes", "Misssing "})
        Else
            ComboBox1.Items.Clear()
            ComboBox2.Items.Clear()
        End If
        categ = "Newspaper AD"
    End Sub
    Private Sub CheckBox2_CheckedChanged(sender As Object, e As EventArgs) Handles
CheckBox2.CheckedChanged
        If CheckBox2.Checked = True Then
            ComboBox1.Items.AddRange(New String() {"TV9", "Suvarna news", "Public TV",
"TV5", "Digvijaya"})
            ComboBox2.Items.AddRange(New String() {"1 min", "2 min", "3 min", "4 min"})
        Else
            ComboBox1.Items.Clear()
            ComboBox2.Items.Clear()
        End If
        categ = "TV AD"
    End Sub
    Private Sub Guna2GradientButton4_Click(sender As Object, e As EventArgs) Handles
Guna2GradientButton4.Click
        OpenFileDialog1.Filter = "PDF File|*.pdf|JPG|*.jpg"
        If OpenFileDialog1.ShowDialog = DialogResult.OK Then
```

```
            TextBox4.Text = OpenFileDialog1.FileName
            PictureBox1.Image = Image.FromFile(OpenFileDialog1.FileName)
        End If
    End Sub
    Private Sub TextBox1_KeyPress(sender As Object, e As KeyPressEventArgs) Handles
TextBox1.KeyPress
        If Not Char.IsNumber(e.KeyChar) And Not e.KeyChar = Chr(Keys.Delete) And Not
e.KeyChar = Chr(Keys.Back) And Not e.KeyChar = Chr(Keys.Space) Then
            e.Handled = True
            MessageBox.Show("RO No should consist of numbers only", "Error",
MessageBoxButtons.OK, MessageBoxIcon.Information)
        End If
    End Sub
    Private Sub Guna2GradientButton2_Click(sender As Object, e As EventArgs) Handles
Guna2GradientButton2.Click
        If TextBox1.Text = "" Then
            MessageBox.Show("RO No required!!", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Information)
            Exit Sub
        ElseIf ComboBox1.SelectedItem = "" Then
            MessageBox.Show("Publisher required!!", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Information)
            Exit Sub
        ElseIf ComboBox2.SelectedItem = "" Then
            MessageBox.Show("Type of AD required !!", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Information)
            Exit Sub
        ElseIf TextBox2.Text = "" Then
            MessageBox.Show("Enter size of AD", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Information)
            Exit Sub
        ElseIf ComboBox3.SelectedItem = "" Then
            MessageBox.Show("Publisher Edition required!!", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Information)
            Exit Sub
        ElseIf TextBox4.Text = "" Then
```

```
        MessageBox.Show("Need to Upload AD !!", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Information)
        Exit Sub
      Else
        Dim cmd2 As SqlCommand = New SqlCommand("Select * From ADBook where
RONo='" + TextBox1.Text + "'", con)
        Dim myreader As SqlDataReader
        con.Open()
        myreader = cmd2.ExecuteReader()
        If (myreader.Read() = False) Then
          MessageBox.Show("RO Number does not exist", "Error",
MessageBoxButtons.OK, MessageBoxIcon.Information)
        Else
          con.Close()
          Try
            If MessageBox.Show("Are you sure need to Edit", "Confirmation",
MessageBoxButtons.YesNo, MessageBoxIcon.Information) = DialogResult.Yes Then
              Dim cmd1 As SqlCommand = New SqlCommand("UPDATE ADBook set
Categorie='" + categ + "',Publisher='" + ComboBox1.SelectedItem.ToString() + "',Type='" +
ComboBox2.SelectedItem.ToString() + "',Size='" + TextBox2.Text + "',DOP='" +
Guna2DateTimePicker1.Value + "',Edition='" + ComboBox3.SelectedItem.ToString() +
"',image=@image Where RONo ='" + TextBox1.Text + "'", con)
              Dim mstr As New MemoryStream
              PictureBox1.Image.Save(mstr, PictureBox1.Image.RawFormat)
              cmd1.Parameters.Add("@image", SqlDbType.Image).Value = mstr.ToArray
              con.Open()
              cmd1.ExecuteNonQuery()
              MessageBox.Show("Update Successfully Saved", "Success",
MessageBoxButtons.OK, MessageBoxIcon.Information)
              con.Close()
              'Dim result As String
              'Dim cmd8 = New SqlCommand("Select MAX(RONo) from ADBook", con)
              'con.Open()
              'result = cmd8.ExecuteScalar().ToString()
              'TextBox1.Text = result + 1
              'con.Close()
```

```
                TextBox2.Clear()
                TextBox4.Clear()
                CheckBox1.Checked = False
                CheckBox1.Checked = False
                ComboBox1.SelectedItem = Nothing
                ComboBox2.SelectedItem = Nothing
                PictureBox1.Image = Nothing
                ComboBox3.SelectedItem = Nothing
            Else
                'do nothing
            End If
        Catch ex As Exception
            MsgBox(ex.Message)
        End Try
    End If
  End If
 End Sub
 Private Sub Guna2GradientButton3_Click(sender As Object, e As EventArgs) Handles
Guna2GradientButton3.Click
  If TextBox1.Text = "" Then
    MessageBox.Show("RO No required!!", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Information)
    Exit Sub
  Else
    Dim cmd2 As SqlCommand = New SqlCommand("Select * From ADBook where
RONo='" + TextBox1.Text + "'", con)
    Dim myreader As SqlDataReader
    con.Open()
    myreader = cmd2.ExecuteReader()
    If (myreader.Read() = False) Then
      MessageBox.Show("RO Number does not exist", "Error",
MessageBoxButtons.OK, MessageBoxIcon.Information)
    Else
      con.Close()
      Try
```

```
            If MessageBox.Show("Are you sure need to Delete", "Confirmation",
MessageBoxButtons.YesNo, MessageBoxIcon.Information) = DialogResult.Yes Then
                Dim cmd1 As SqlCommand = New SqlCommand("Delete from ADBook where
RONo ='" + TextBox1.Text + "'", con)
                con.Open()
                cmd1.ExecuteNonQuery()
                MessageBox.Show("Data Deleted Successfully", "Success",
MessageBoxButtons.OK, MessageBoxIcon.Information)
                con.Close()
                Dim result As String
                Dim cmd3 = New SqlCommand("Select MAX(RONo) from ADBook", con)
                con.Open()
                result = cmd3.ExecuteScalar().ToString() + 1
                TextBox1.Text = result
                con.Close()
                TextBox2.Clear()
                TextBox4.Clear()
                CheckBox1.Checked = False
                CheckBox1.Checked = False
                ComboBox1.SelectedItem = Nothing
                ComboBox2.SelectedItem = Nothing
                PictureBox1.Image = Nothing
                ComboBox3.SelectedItem = Nothing
            Else
                'do nothing
            End If
        Catch ex As Exception
            MsgBox(ex.Message)
        End Try
    End If
  End If
End Sub
Private Sub TextBox1_Load(sender As Object, e As EventArgs) Handles TextBox1.Load
  'Dim result As String
  'Dim cmd = New SqlCommand("Select MAX(RONo) from ADBook", con)
  'con.Open()
```

```
        'result = cmd.ExecuteScalar().ToString() + 1
        'TextBox1.Text = result
        'con.Close()
    End Sub
    Private Sub Guna2GradientButton5_Click(sender As Object, e As EventArgs) Handles
Guna2GradientButton5.Click
        TextBox2.Clear()
        TextBox4.Clear()
        CheckBox1.Checked = False
        CheckBox1.Checked = False
        ComboBox1.SelectedItem = Nothing
        ComboBox2.SelectedItem = Nothing
        PictureBox1.Image = Nothing
        ComboBox3.SelectedItem = Nothing
    End Sub
End Class
```

AD Billing

```
Public Class ADBilling
    Dim con As SqlConnection = New SqlConnection("Data
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\Tejas\Documents\Massm
edia Advt Management\Massmedia Advt Management\MyDatabase.mdf;Integrated
Security=True")
    Private Sub Guna2GradientButton2_Click(sender As Object, e As EventArgs) Handles
Guna2GradientButton2.Click
        If TextBox1.Text = "" Then
            MessageBox.Show("RO No required!!", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Information)
            TextBox1.BorderColor = Color.Red
            TextBox1.BorderThickness =  1
            Exit Sub
        ElseIf Guna2DateTimePicker1.Value.ToString = "" Then
            MessageBox.Show("Bill Date Required !!", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Information)
            Exit Sub
        ElseIf TextBox2.Text = "" Then
```

```
        MessageBox.Show("Client Name Required !!", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Information)
        Exit Sub
    ElseIf TextBox3.Text = "" Then
        MessageBox.Show("Client Mobile Number Required ", "Error",
MessageBoxButtons.OK, MessageBoxIcon.Information)
        Exit Sub
    ElseIf ComboBox1.SelectedItem = "" Then
        MessageBox.Show("Payment Mode Required!!", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Information)
        Exit Sub
    ElseIf TextBox4.Text = "" Then
        MessageBox.Show("Payment Reference No. Required !!", "Error",
MessageBoxButtons.OK, MessageBoxIcon.Information)
        Exit Sub
    ElseIf TextBox5.Text = "" Then
        MessageBox.Show("Need to Upload Bill !!", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Information)
        Exit Sub
    Else
        Dim cmd As SqlCommand = New SqlCommand("Select * From ADBilling where
RONo='" + TextBox1.Text + "'", con)
        Dim myreader As SqlDataReader
        con.Close()
        con.Open()
        myreader = cmd.ExecuteReader()
        If (myreader.Read()) Then
            MessageBox.Show("RO Number already exist", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Information)
        Else
            con.Close()
            Dim cmd1 As SqlCommand = New SqlCommand("Select * From ADBook where
RONo='" + TextBox1.Text + "'", con)
            Dim myreader1 As SqlDataReader
            con.Open()
            myreader1 = cmd1.ExecuteReader()
```

```
            If (myreader1.Read() = False) Then
                MessageBox.Show(" AD not Booked ", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Information)
            Else
                con.Close()
                Try
                    If MessageBox.Show("Are you sure need to Save", "Confirmation",
MessageBoxButtons.YesNo, MessageBoxIcon.Information) = DialogResult.Yes Then
                        Dim cmd3 As SqlCommand = New SqlCommand("insert into ADBilling
values
      ('" + TextBox1.Text + "','" + Guna2DateTimePicker1.Value + "','" + TextBox2.Text +
"','" + TextBox3.Text + "','" + TextBox4.Text + "','" + ComboBox1.SelectedItem.ToString() +
"','" + TextBox5.Text + "',@image)", con)
                        Dim mstr As New MemoryStream
                        PictureBox1.Image.Save(mstr, PictureBox1.Image.RawFormat)
                        cmd3.Parameters.Add("@image", SqlDbType.Image).Value = mstr.ToArray
                        con.Open()
                        cmd3.ExecuteNonQuery()
                        MessageBox.Show("Successfully Saved", "Success",
MessageBoxButtons.OK, MessageBoxIcon.Information)
                        con.Close()
                        TextBox2.Clear()
                        TextBox3.Clear()
                        ComboBox1.SelectedItem = Nothing
                        TextBox4.Clear()
                        TextBox5.Clear()
                        PictureBox1.Image = Nothing
                    Else
                        'do nothing
                    End If
                Catch ex As Exception
                    MsgBox(ex.Message)
                End Try
            End If
        End If
    End If
```

```
    End Sub

    Private Sub Guna2GradientButton4_Click(sender As Object, e As EventArgs) Handles
Guna2GradientButton4.Click
        OpenFileDialog1.Filter = "JPG|*.jpg"
        If OpenFileDialog1.ShowDialog = DialogResult.OK Then
            TextBox6.Text = OpenFileDialog1.FileName
            PictureBox1.Image = Image.FromFile(OpenFileDialog1.FileName)
        End If
    End Sub
    Private Sub Guna2GradientButton1_Click(sender As Object, e As EventArgs) Handles
Guna2GradientButton1.Click
        If TextBox1.Text = "" Then
            MessageBox.Show("RO No required!!", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Information)
            Exit Sub
        ElseIf Guna2DateTimePicker1.Value.ToString = "" Then
            MessageBox.Show("Bill Date Required !!", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Information)
            Exit Sub
        ElseIf TextBox2.Text = "" Then
            MessageBox.Show("Client Name Required !!", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Information)
            Exit Sub
        ElseIf TextBox3.Text = "" Then
            MessageBox.Show("Client Mobile Number Required ", "Error",
MessageBoxButtons.OK, MessageBoxIcon.Information)
            Exit Sub
        ElseIf ComboBox1.SelectedItem = "" Then
            MessageBox.Show("Payment Mode Required!!", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Information)
            Exit Sub
        ElseIf TextBox4.Text = "" Then
            MessageBox.Show("Payment Reference No. Required !!", "Error",
MessageBoxButtons.OK, MessageBoxIcon.Information)
            Exit Sub
```

```
        ElseIf TextBox5.Text = "" Then
            MessageBox.Show("Need to Upload Bill !!", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Information)
            Exit Sub
        Else
            Dim cmd4 As SqlCommand = New SqlCommand("Select * From ADBilling where
RONo='" + TextBox1.Text + "'", con)
            Dim myreader3 As SqlDataReader
            con.Close()
            con.Open()
            myreader3 = cmd4.ExecuteReader()
            If (myreader3.Read() = False) Then
                MessageBox.Show("RO Number does not exist", "Error",
MessageBoxButtons.OK, MessageBoxIcon.Information)
            Else
                con.Close()
                Try
                    If MessageBox.Show("Are you sure need to Edit", "Confirmation",
MessageBoxButtons.YesNo, MessageBoxIcon.Information) Then
                        Dim cmd6 As SqlCommand = New SqlCommand("UPDATE ADBilling set [Bill
Date]='" + Guna2DateTimePicker1.Value + "',[Client Name]='" + TextBox2.Text + "',[Client
Mobile No.]='" + TextBox3.Text + "',[Bill Amount]='" + TextBox4.Text + "',[Payment
Mode]='" + ComboBox1.SelectedItem.ToString() + "',[Payment Ref No.]='" +
TextBox5.Text + "',[Image]=@image Where RONo='" + TextBox1.Text + "'", con)
                        Dim mstr As New MemoryStream
                        PictureBox1.Image.Save(mstr, PictureBox1.Image.RawFormat)
                        cmd6.Parameters.Add("@image", SqlDbType.Image).Value = mstr.ToArray
                        con.Open()
                        cmd6.ExecuteNonQuery()
                        MessageBox.Show("Update Successfully Saved", "Success",
MessageBoxButtons.OK, MessageBoxIcon.Information)
                        con.Close()
                        TextBox2.Clear()
                        TextBox3.Clear()
                        ComboBox1.SelectedItem = Nothing
                        TextBox4.Clear()
```

```
                TextBox5.Clear()
                PictureBox1.Image = Nothing
            Else
                'do nothing
            End If
        Catch ex As Exception
            MsgBox(ex.Message)
        End Try
    End If
  End If
End Sub
Private Sub Guna2GradientButton3_Click(sender As Object, e As EventArgs) Handles
Guna2GradientButton3.Click
  If TextBox1.Text = "" Then
    MessageBox.Show("RO No required!!", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Information)
    Exit Sub
  Else
    Dim cmd7 As SqlCommand = New SqlCommand("Select * From ADBilling where
RONo='" + TextBox1.Text + "'", con)
    Dim myreader4 As SqlDataReader
    con.Close()
    con.Open()
    myreader4 = cmd7.ExecuteReader()
    If (myreader4.Read() = False) Then
      MessageBox.Show("RO Number does not exist", "Error",
MessageBoxButtons.OK, MessageBoxIcon.Information)
    Else
      con.Close()
      Try
        If MessageBox.Show("Are you sure need to Delete", "Confirmation",
MessageBoxButtons.YesNo, MessageBoxIcon.Information) = DialogResult.Yes Then
          Dim cmd8 As SqlCommand = New SqlCommand("Delete from ADBilling
where RONo ='" + TextBox1.Text + "'", con)
          con.Open()
          cmd8.ExecuteNonQuery()
```

```
                MessageBox.Show("Data Deleted Successfully", "Success",
MessageBoxButtons.OK, MessageBoxIcon.Information)
                con.Close()
                TextBox2.Clear()
                TextBox3.Clear()
                ComboBox1.SelectedItem = Nothing
                TextBox4.Clear()
                TextBox5.Clear()
                PictureBox1.Image = Nothing
              Else
                'do nothing
              End If
            Catch ex As Exception
              MsgBox(ex.Message)
            End Try
          End If
        End If
    End Sub
    Private Sub Guna2DateTimePicker1_ValueChanged(sender As Object, e As EventArgs)
Handles Guna2DateTimePicker1.ValueChanged
        Guna2DateTimePicker1.CustomFormat = "dd/MM/yyyy"
    End Sub
    Private Sub TextBox1_KeyPress(sender As Object, e As KeyPressEventArgs) Handles
TextBox1.KeyPress
        If Not Char.IsNumber(e.KeyChar) And Not e.KeyChar = Chr(Keys.Delete) And Not
e.KeyChar = Chr(Keys.Back) And Not e.KeyChar = Chr(Keys.Space) Then
          e.Handled = True
          MessageBox.Show("RO No should consist of numbers only", "Error",
MessageBoxButtons.OK, MessageBoxIcon.Information)
        End If
    End Sub
    Private Sub TextBox2_KeyPress(sender As Object, e As KeyPressEventArgs) Handles
TextBox2.KeyPress
        If Not Char.IsLetter(e.KeyChar) And Not e.KeyChar = Chr(Keys.Delete) And Not
e.KeyChar = Chr(Keys.Back) And Not e.KeyChar = Chr(Keys.Space) Then
          e.Handled = True
```

```
        MessageBox.Show("Client name should consist of Letters only", "Error",
MessageBoxButtons.OK, MessageBoxIcon.Information)
    End If
  End Sub
  Private Sub TextBox3_KeyPress(sender As Object, e As KeyPressEventArgs) Handles
TextBox3.KeyPress
    If Not Char.IsNumber(e.KeyChar) And Not e.KeyChar = Chr(Keys.Delete) And Not
e.KeyChar = Chr(Keys.Back) And Not e.KeyChar = Chr(Keys.Space) Then
      e.Handled = True
      MessageBox.Show("Mobile Number should consist of numbers only", "Error",
MessageBoxButtons.OK, MessageBoxIcon.Information)
    End If
  End Sub
  Private Sub Guna2GradientButton5_Click(sender As Object, e As EventArgs) Handles
Guna2GradientButton5.Click
    TextBox2.Clear()
    TextBox3.Clear()
    ComboBox1.SelectedItem = Nothing
    TextBox4.Clear()
    TextBox5.Clear()
    PictureBox1.Image = Nothing
  End Sub
End Class
```

AD Calculator

```
Public Class ADCalculator
  Private Sub CheckBox1_CheckedChanged(sender As Object, e As EventArgs) Handles
CheckBox1.CheckedChanged
    If CheckBox1.Checked = True Then
      ComboBox1.Items.AddRange(New String() {"Vijayakarnataka", "VijayaVani",
"Kannadaprabha", "HosaDigantha", "Prajavani", "Udayavani", "Kolarapatrike", "Indian
Express", "Deccan Herald"})
      ComboBox2.Items.AddRange(New String() {"Classified", "Court notice", "Public
Notice", "Govt AD", "Wishes", "Misssing "})
    Else
      ComboBox1.Items.Clear()
```

```
            ComboBox2.Items.Clear()
        End If
    End Sub
    Private Sub CheckBox2_CheckedChanged(sender As Object, e As EventArgs) Handles
CheckBox2.CheckedChanged
        If CheckBox2.Checked = True Then
            Label9.Text = "Timing of AD"
            ComboBox1.Items.AddRange(New String() {"TV9", "Suvarna news", "Public TV",
"TV5", "Digvijaya"})
            ComboBox2.Items.AddRange(New String() {"Full Screen", "Sliding", "Highlights",
"Border AD"})
        Else
            ComboBox1.Items.Clear()
            ComboBox2.Items.Clear()
        End If
    End Sub
    Private Sub Guna2GradientButton3_Click(sender As Object, e As EventArgs) Handles
Guna2GradientButton3.Click
        If ComboBox1.SelectedItem = "" Then
            MessageBox.Show("Publisher required!!", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Information)
            Exit Sub
        ElseIf ComboBox2.SelectedItem = "" Then
            MessageBox.Show("Type of AD required !!", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Information)
            Exit Sub
        ElseIf TextBox2.Text = "" Then
            MessageBox.Show("Enter the Correct Size!!", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Information)
            Exit Sub
        ElseIf TextBox3.Text = "" Then
            MessageBox.Show("Enter the Correct Size!!", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Information)
            Exit Sub
        Else
            Dim a, b, n, g, t As Integer
```

```
a = TextBox2.Text
b = TextBox3.Text
If ComboBox1.SelectedItem = "Vijayakarnataka" Then
   If ComboBox2.SelectedItem = "Classified" Then
      n = a * b * 10
   ElseIf ComboBox2.SelectedItem = "Court notice" Then
      n = a * b * 60
   ElseIf ComboBox2.SelectedItem = "Public Notice" Then
      n = a * b * 60
   ElseIf ComboBox2.SelectedItem = "Govt AD" Then
      n = a * b * 99
   ElseIf ComboBox2.SelectedItem = "Wishes" Then
      n = a * b * 60
   ElseIf ComboBox2.SelectedItem = "Misssing" Then
      n = a * b * 55
   End If
End If
If ComboBox1.SelectedItem = "VijayaVani" Then
   If ComboBox2.SelectedItem = "Classified" Then
      n = a * b * 20
   ElseIf ComboBox2.SelectedItem = "Court notice" Then
      n = a * b * 45
   ElseIf ComboBox2.SelectedItem = "Public Notice" Then
      n = a * b * 45
   ElseIf ComboBox2.SelectedItem = "Govt AD" Then
      n = a * b * 95
   ElseIf ComboBox2.SelectedItem = "Wishes" Then
      n = a * b * 65
   ElseIf ComboBox2.SelectedItem = "Misssing" Then
      n = a * b * 65
   End If
End If
If ComboBox1.SelectedItem = "Kannadaprabha" Then
   If ComboBox2.SelectedItem = "Classified" Then
      n = a * b * 20
   ElseIf ComboBox2.SelectedItem = "Court notice" Then
```

```
            n = a * b * 45
        ElseIf ComboBox2.SelectedItem = "Public Notice" Then
            n = a * b * 45
        ElseIf ComboBox2.SelectedItem = "Govt AD" Then
            n = a * b * 87
        ElseIf ComboBox2.SelectedItem = "Wishes" Then
            n = a * b * 50
        ElseIf ComboBox2.SelectedItem = "Misssing" Then
            n = a * b * 50
        End If
    End If
    If ComboBox1.SelectedItem = "HosaDigantha" Then
        If ComboBox2.SelectedItem = "Classified" Then
            n = a * b * 20
        ElseIf ComboBox2.SelectedItem = "Court notice" Then
            n = a * b * 20
        ElseIf ComboBox2.SelectedItem = "Public Notice" Then
            n = a * b * 20
        ElseIf ComboBox2.SelectedItem = "Govt AD" Then
            n = a * b * 65
        ElseIf ComboBox2.SelectedItem = "Wishes" Then
            n = a * b * 30
        ElseIf ComboBox2.SelectedItem = "Misssing" Then
            n = a * b * 30
        End If
    End If
    If ComboBox1.SelectedItem = "Prajavani" Then
        If ComboBox2.SelectedItem = "Classified" Then
            n = a * b * 20
        ElseIf ComboBox2.SelectedItem = "Court notice" Then
            n = a * b * 65
        ElseIf ComboBox2.SelectedItem = "Public Notice" Then
            n = a * b * 65
        ElseIf ComboBox2.SelectedItem = "Govt AD" Then
            n = a * b * 98
        ElseIf ComboBox2.SelectedItem = "Wishes" Then
```

```
        n = a * b * 80
    ElseIf ComboBox2.SelectedItem = "Misssing" Then
        n = a * b * 80
    End If
End If
If ComboBox1.SelectedItem = "Udayavani" Then
    If ComboBox2.SelectedItem = "Classified" Then
        n = a * b * 30
    ElseIf ComboBox2.SelectedItem = "Court notice" Then
        n = a * b * 40
    ElseIf ComboBox2.SelectedItem = "Public Notice" Then
        n = a * b * 40
    ElseIf ComboBox2.SelectedItem = "Govt AD" Then
        n = a * b * 88
    ElseIf ComboBox2.SelectedItem = "Wishes" Then
        n = a * b * 50
    ElseIf ComboBox2.SelectedItem = "Misssing" Then
        n = a * b * 50
    End If
End If
If ComboBox1.SelectedItem = "Kolarapatrike" Then
    If ComboBox2.SelectedItem = "Classified" Then
        n = a * b * 20
    ElseIf ComboBox2.SelectedItem = "Court notice" Then
        n = a * b * 15
    ElseIf ComboBox2.SelectedItem = "Public Notice" Then
        n = a * b * 15
    ElseIf ComboBox2.SelectedItem = "Govt AD" Then
        n = a * b * 30
    ElseIf ComboBox2.SelectedItem = "Wishes" Then
        n = a * b * 20
    ElseIf ComboBox2.SelectedItem = "Misssing" Then
        n = a * b * 20
    End If
End If
If ComboBox1.SelectedItem = "Indian Express" Then
```

```
    If ComboBox2.SelectedItem = "Classified" Then
      n = a * b * 20
    ElseIf ComboBox2.SelectedItem = "Court notice" Then
      n = a * b * 80
    ElseIf ComboBox2.SelectedItem = "Public Notice" Then
      n = a * b * 80
    ElseIf ComboBox2.SelectedItem = "Govt AD" Then
      n = a * b * 99
    ElseIf ComboBox2.SelectedItem = "Wishes" Then
      n = a * b * 100
    ElseIf ComboBox2.SelectedItem = "Misssing" Then
      n = a * b * 100
    End If
  End If
  If ComboBox1.SelectedItem = "Deccan Herald" Then
    If ComboBox2.SelectedItem = "Classified" Then
      n = a * b * 20
    ElseIf ComboBox2.SelectedItem = "Court notice" Then
      n = a * b * 65
    ElseIf ComboBox2.SelectedItem = "Public Notice" Then
      n = a * b * 65
    ElseIf ComboBox2.SelectedItem = "Govt AD" Then
      n = a * b * 98
    ElseIf ComboBox2.SelectedItem = "Wishes" Then
      n = a * b * 80
    ElseIf ComboBox2.SelectedItem = "Misssing" Then
      n = a * b * 80
    End If
  End If
  If ComboBox1.SelectedItem = "TV9" Then
    If ComboBox2.SelectedItem = "Full Screen" Then
      n = a * b * 90
    ElseIf ComboBox2.SelectedItem = "Sliding" Then
      n = a * b * 60
    ElseIf ComboBox2.SelectedItem = "Highlights" Then
      n = a * b * 40
```

```
    ElseIf ComboBox2.SelectedItem = "Border AD" Then
      n = a * b * 20
    End If
  End If
  If ComboBox1.SelectedItem = "Suvarna news" Then
    If ComboBox2.SelectedItem = "Full Screen" Then
      n = a * b * 70
    ElseIf ComboBox2.SelectedItem = "Sliding" Then
      n = a * b * 40
    ElseIf ComboBox2.SelectedItem = "Highlights" Then
      n = a * b * 25
    ElseIf ComboBox2.SelectedItem = "Border AD" Then
      n = a * b * 15
    End If
  End If
  If ComboBox1.SelectedItem = "Public TV" Then
    If ComboBox2.SelectedItem = "Full Screen" Then
      n = a * b * 80
    ElseIf ComboBox2.SelectedItem = "Sliding" Then
      n = a * b * 50
    ElseIf ComboBox2.SelectedItem = "Highlights" Then
      n = a * b * 30
    ElseIf ComboBox2.SelectedItem = "Border AD" Then
      n = a * b * 20
    End If
  End If
  If ComboBox1.SelectedItem = "TV5" Then
    If ComboBox2.SelectedItem = "Full Screen" Then
      n = a * b * 60
    ElseIf ComboBox2.SelectedItem = "Sliding" Then
      n = a * b * 25
    ElseIf ComboBox2.SelectedItem = "Highlights" Then
      n = a * b * 40
    ElseIf ComboBox2.SelectedItem = "Border AD" Then
      n = a * b * 15
    End If
```

```
        End If
        If ComboBox1.SelectedItem = "Digvijaya" Then
          If ComboBox2.SelectedItem = "Full Screen" Then
            n = a * b * 40
          ElseIf ComboBox2.SelectedItem = "Sliding" Then
            n = a * b * 25
          ElseIf ComboBox2.SelectedItem = "Highlights" Then
            n = a * b * 15
          ElseIf ComboBox2.SelectedItem = "Border AD" Then
            n = a * b * 10
          End If
        End If
        g = n * 5 / 100
        t = n + g
        Guna2TextBox1.Text = n
        Guna2TextBox2.Text = g
        Guna2TextBox3.Text = t
      End If
    End Sub
End Class
```

AD Records

```
Imports System.Data
Imports System.Data.SqlClient
Public Class ADRecords
  Dim con As SqlConnection = New SqlConnection("Data
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\Tejas\Documents\Massm
edia Advt Management\Massmedia Advt Management\MyDatabase.mdf;Integrated
Security=True")
  Dim table As New DataTable()
  Dim table1 As New DataTable()
  Private Sub Guna2GradientButton1_Click(sender As Object, e As EventArgs) Handles
Guna2GradientButton1.Click
    If ComboBox1.SelectedItem = "RO No" Then
      TextBox1.BringToFront()
      Guna2DataGridView1.DataSource.Clear()
      Dim cmd1 As SqlCommand = New SqlCommand("Select * from ADBook Where
RONo = '" + TextBox1.Text + "'", con)
```

```
            cmd1.Connection = con
            Dim adapter As New SqlDataAdapter(cmd1)
            adapter.Fill(table)
        ElseIf ComboBox1.SelectedItem = "DOP" Then
            Guna2DataGridView1.DataSource.Clear()
            Dim cmd4 As SqlCommand = New SqlCommand("Select * from ADBook Where DOP
= '" + Guna2DateTimePicker1.Value + "'", con)
            cmd4.Connection = con
            Dim adapter As New SqlDataAdapter(cmd4)
            adapter.Fill(table)
        End If
        Guna2DataGridView2.DataSource.Clear()
        Dim cmd5 As SqlCommand = New SqlCommand("Select * from ADBilling Where
RONo = '" + TextBox1.Text + "'", con)
        cmd5.Connection = con
        Dim adapter1 As New SqlDataAdapter(cmd5)
        adapter1.Fill(table1)
    End Sub

    Private Sub ADRecords_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        Guna2DataGridView1.AutoSizeColumnsMode =
DataGridViewAutoSizeColumnsMode.Fill
        Guna2DataGridView1.RowTemplate.Height = 100
        Dim img As New DataGridViewImageColumn
        Guna2DataGridView1.DataSource = table
        img = Guna2DataGridView1.Columns(7)
        img.ImageLayout = DataGridViewImageCellLayout.Stretch
        Guna2DataGridView1.Columns(5).DefaultCellStyle.Format = "dd/MM/yyyy"
        Guna2DataGridView2.AutoSizeColumnsMode =
DataGridViewAutoSizeColumnsMode.Fill
        Guna2DataGridView2.RowTemplate.Height = 100
        Dim imga As New DataGridViewImageColumn
        Guna2DataGridView2.DataSource = table1
        imga = Guna2DataGridView2.Columns(7)
        imga.ImageLayout = DataGridViewImageCellLayout.Stretch
    End Sub
    Private Sub ComboBox1_SelectedIndexChanged(sender As Object, e As EventArgs)
Handles ComboBox1.SelectedIndexChanged
        If ComboBox1.SelectedItem = "DOP" Then
            Guna2DateTimePicker1.BringToFront()
        Else
            Guna2DateTimePicker1.SendToBack()
        End If
```

```
    End Sub
    Private Sub Guna2GradientButton2_Click(sender As Object, e As EventArgs) Handles
Guna2GradientButton2.Click
        Guna2DataGridView1.BringToFront()
        Guna2GradientButton4.BringToFront()
    End Sub
    Private Sub Guna2GradientButton3_Click(sender As Object, e As EventArgs) Handles
Guna2GradientButton3.Click
        Guna2DataGridView2.BringToFront()
        Guna2GradientButton5.BringToFront()
    End Sub
    Private Sub Guna2GradientButton4_Click(sender As Object, e As EventArgs) Handles
Guna2GradientButton4.Click
        PrintPreviewDialog1.ShowDialog()
    End Sub
    Private Sub PrintDocument1_PrintPage(sender As Object, e As
Printing.PrintPageEventArgs) Handles PrintDocument1.PrintPage
        Dim bm As New Bitmap(Me.Guna2DataGridView1.Width,
Me.Guna2DataGridView1.Height)
        Guna2DataGridView1.DrawToBitmap(bm, New Rectangle(0, 0,
Me.Guna2DataGridView1.Width, Me.Guna2DataGridView2.Height))
        e.Graphics.DrawImage(bm, 0, 0)
    End Sub
    Private Sub Guna2GradientButton5_Click(sender As Object, e As EventArgs) Handles
Guna2GradientButton5.Click
        PrintPreviewDialog2.ShowDialog()
    End Sub
    Private Sub PrintDocument2_PrintPage(sender As Object, e As
Printing.PrintPageEventArgs) Handles PrintDocument2.PrintPage
        Dim bm As New Bitmap(Me.Guna2DataGridView2.Width,
Me.Guna2DataGridView2.Height)
        Guna2DataGridView1.DrawToBitmap(bm, New Rectangle(0, 0,
Me.Guna2DataGridView1.Width, Me.Guna2DataGridView2.Height))
        e.Graphics.DrawImage(bm, 0, 0)
    End Sub
End Class
```

AD Reports

```
Imports System.Data
Imports System.Data.SqlClient
Imports  System.Configuration
Public Class SAReport
```

```
    Dim con As SqlConnection = New SqlConnection("Data
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\Tejas\Documents\Massm
edia Advt Management\Massmedia Advt Management\MyDatabase.mdf;Integrated
Security=True")
    Private Sub SAReport_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        ComboBox1.Text = "2020"
        ComboBox2.Text = "2021"
        loaddata()
        loaddata1()
        loaddata2()
    End Sub
    Private Sub loaddata()
        'Chart1.DataSource = getdata()
        'Chart1.Series("Publisher").Points.Clear()
        'Chart1.Series("Publisher").XValueMember = "year"
        'Chart1.Series("Publisher").YValueMembers = "Total"
        Chart1.DataBindCrossTable(getdata().DefaultView, "Publisher", "year", "Total",
"Label=Total")
    End Sub
    Private Function getdata() As DataTable
        Dim dtChartData As New DataTable()
        Using cmd As New SqlCommand("pubchart", con)
            cmd.CommandType = CommandType.StoredProcedure
            con.Open()
            cmd.Parameters.AddWithValue("@fromyear", ComboBox1.Text)
            cmd.Parameters.AddWithValue("@toyear", ComboBox2.Text)
            Dim reader As SqlDataReader = cmd.ExecuteReader()
            dtChartData.Load(reader)
            con.Close()
        End Using
        Return dtChartData
    End Function
    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        Chart1.Series.Clear()
        loaddata()
        loaddata1()
        loaddata2()
    End Sub
    Private Sub loaddata1()
        Chart2.DataSource = getdata1()
        Chart2.Series("Total").Points.Clear()
        Chart2.Series("Total").XValueMember = "date"
        Chart2.Series("Total").YValueMembers = "BillAmt"
```

```
    End Sub
    Private Function getdata1() As DataTable
        Dim dtChartData1 As New DataTable()
        Using cmd As New SqlCommand("totalchart", con)
            cmd.CommandType = CommandType.StoredProcedure
            con.Open()
            cmd.Parameters.AddWithValue("@fromyear", ComboBox1.Text)
            cmd.Parameters.AddWithValue("@toyear", ComboBox2.Text)
            Dim reader As SqlDataReader = cmd.ExecuteReader()
            dtChartData1.Load(reader)
            con.Close()
        End Using
        Return dtChartData1
    End Function
    Private Sub loaddata2()
        Chart3.DataSource = getdata2()
        Chart3.Series("Total").Points.Clear()
        Chart3.Series("Total").XValueMember = "Type"
        Chart3.Series("Total").YValueMembers = "Total"
        'Chart3.DataBindCrossTable(getdata2().DefaultView, "Type", "year", "Total",
"Label=Total")
    End Sub
    Private Function getdata2() As DataTable
        Dim dtChartData2 As New DataTable()
        Using cmd As New SqlCommand("typechart", con)
            cmd.CommandType = CommandType.StoredProcedure
            con.Open()
            cmd.Parameters.AddWithValue("@fromyear", ComboBox1.Text)
            cmd.Parameters.AddWithValue("@toyear", ComboBox2.Text)
            Dim reader As SqlDataReader = cmd.ExecuteReader()
            dtChartData2.Load(reader)
            con.Close()
        End Using
        Return dtChartData2
    End Function
    Private Sub Guna2GradientButton1_Click(sender As Object, e As EventArgs) Handles
Guna2GradientButton1.Click
        Reportview.Show()
    End Sub
    Private Sub Guna2GradientButton2_Click(sender As Object, e As EventArgs) Handles
Guna2GradientButton2.Click
        Reportview1.Show()
    End Sub End Class
```

Tables

LOGIN TABLE

Table	Column	Data type	Length
Login	Username	NVarchar	50
	Password	NVarchar	50

Advertisement Booking

Table	Column	Data type	Length
Advertisement Booking	RO No	INT	50
	Categorie	NVARCHAR	50
	Publisher	NVARCHAR	50
	Type	NVARCHAR	50
	Size	NVARCHAR	50
	DOP	NVARCHAR	50
	Edition	NVARCHAR	50
	image	IMAGE	

Advertisement Billing

Table	Column	Data type	Length
Advertisement Billing	RO No	INT	50
	Bill Date	NVARCHAR	50
	Client Name	NVARCHAR	50
	Mobile No	INT	10
	Bill Amount	INT	50
	Payment Mode	NVARCHAR	50
	Ref No	NVARCHAR	50
	Image	IMAGE	

Testing

TESTING

SYSTEM TESTING

Software testing involves the execution of a software component or system component to evaluate one or more properties of interest. In general, these properties indicate the extent to which the component or system under test:

- meets the requirements that guided its design and development,
- responds correctly to all kinds of inputs,
- performs its functions within an acceptable time,
- is sufficiently usable,
- can be installed and run in its intended environments
- achieves the general result its stakeholder's desire. Software testing can be conducted as soon as executable software (even if partially complete) exists. The overall approach to software development often determines when and how testing is conducted. For example, in a phased process, most testing occurs after system requirements have been defined and then implemented in testable programs. In contrast, under an agile approach, requirements, programming, and testing are often done concurrently.

Software testing methods are traditionally divided into white- and blackbox testing. These two approaches are used to describe the point of view that the tester takes when designing test cases. A hybrid approach called grey-box testing may also be applied to software testing methodology. With the concept of grey-box testing—which develops tests from specific design elements—gaining prominence, this "arbitrary distinction" between black- and white-box testing has faded somewhat.

WHITE BOX TESTING:

While white-box testing can be applied at the unit, integration, and system levels of the software testing process, it is usually done at the unit level. It can test paths within a unit, paths between units during integration, and between subsystems during a system–level test. Though this method of test design can uncover many errors or problems, it might not detect unimplemented parts of the specification or missing requirements.

BLACK BOX TESTING

Black-box testing (also known as functional testing) treats the software as a "black box," examining functionality without any knowledge of internal implementation, without seeing the source code. The testers are only aware of what the software is supposed to do, not how it does it. Black-box testing methods include: equivalence partitioning, boundary value analysis, all-pairs testing, state transition tables, decision table testing, fuzz testing, model-based testing, use case testing, exploratory testing, and specification-based testing.

TESTING STRATEGIES

A strategy for software testing must accommodate low level tests that are necessary to verify that all small source code segment has been correctly implemented as well as high level tests that validate major system functions against customer requirements.

Testing fundamentals

Testing is a process of executing program with the intent of finding error. A good test case is one that has high probability of finding an undiscovered error. If testing is conducted successfully it uncovers the errors in the software. Testing cannot show the absence of defects, it can only show that software defects present.

Unit testing

Unit testing is a software development process that involves a synchronized application of a broad spectrum of defect prevention and detection strategies in order to reduce software development risks, time, and costs. It is performed by the software developer or engineer during the construction phase of the software development life cycle. Unit testing aims to eliminate construction errors before code is promoted to additional testing; this strategy is intended to increase the quality of the resulting software as well as the efficiency of the overall development process. Depending on the organization's expectations for software development, unit testing might include static code analysis, data-flow analysis, metrics analysis, peer code reviews, code coverage analysis and other software testing practices.

Integration testing

integration testing is any type of software testing that seeks to verify the interfaces between components against a software design. Software components may be integrated in an iterative way or all together. Normally the former is considered a better practice since it allows interface issues to be located more quickly and fixed. Integration testing works to expose defects in the interfaces and interaction between integrated components (modules). Progressively larger groups of tested software components corresponding to elements of the architectural design are integrated and tested until the software works as a system.

Integration tests usually involve a lot of code, and produce traces that are larger than those produced by unit tests. This has an impact on the ease of localizing the fault when an integration test fails. To overcome this issue, it has been proposed to automatically cut the large tests in smaller pieces to improve fault localization.

System testing

System testing tests a completely integrated system to verify that the system meets its requirements. For example, a system test might involve testing a login interface, then creating and editing an entry, plus sending or printing results, followed by summary processing or deletion (or archiving) of entries, then logoff.

Acceptance testing

Commonly this level of Acceptance testing includes the following four types:

- User acceptance testing
- Operational acceptance testing
- Contractual and regulatory acceptance testing
- Alpha and beta testing

User acceptance testing and Alpha and beta testing are described in the next #Testing types section.

Operational acceptance is used to conduct operational readiness (prerelease) of a product, service or system as part of a quality management system. OAT is a common type of non-functional software testing, used mainly in software development and software maintenance projects. This type of testing focuses on the operational readiness of the system to be supported, or to become part of the production environment.

In addition, the software testing should ensure that the portability of the system, as well as working as expected, does not also damage or partially corrupt its operating environment or cause other processes within that environment to become inoperative.

Test case

Test case are derived to ensure that all statements in the program have been executed at least once during testing and that all logical condition have been executed. Using white box testing methods, the software engineer can drive test cases that

- Guarantee that logical decisions on their true and false sides.
- Exercise all logical decision on their true and false sides.
- Execute all loops at their boundaries and within their operational bounds.
- Exercise internal data structure to assure their validity. Tools to special importance during acceptance testing include:

1. Test coverage analyzer: Records the control paths followed for each test case.

2. Timing analyzer: This is also called a profiler, reports the time spent in various regions of the code to improve system performance.

3. Coding standards: Static analyzer and standard checkers are used to inspect code for deviation from standard and guidelines.

The Package and Development Wizard

The package development wizard tool was finally used to create installation packages for the visual basic application that was developed and then installed them to the end-users computers. We must have a saved and compiled project before we can run this wizard.

OPTIONS ARE:

PACKAGE:

This takes you to the step in packaging your project. When you finished packaging your application, you can return to the main screen and deploy our packages.

DELAY:

This takes you to the first step in developing your package. You can use this part of the wizard if you have a packaged project that you want to distribute to floppy disks, to a network or local folder location, or over the internet.

MANAGE SCRIPT:

Displays the manage scripts dialog box where you can use rename, duplicate, delete packaging and deployment scripts.

EDUCATION AND TRAINING:

Next comes training the members and other users of the organization to make them familiar with using the interfaces. This doesn't involve much strain as the system is menu driven and help manuals are available at any moment. User manual is developed to educate the users about the various feature support the system and the corresponding screens and their usage. Apart from this one-day training is given to the users

Future Enhancement

Future Enhancement

- We can update this system as online application
- The main advantage of online application is that, a person can get the information from anywhere.
- They can also pay through banking.
- By the future technology people from anywhere can book for visiting the places.
- Finger print matching adds more strength to this system so that frauds can be solved easily.
- Face recognition technology can be applied which avoids the theft

Conclusion

Conclusion

The package was designed in such a way that future modifications can be done easily. The following conclusions can be deducted from the development of the project.

- Automation of the entire system improves the efficiency.
- It provides a friendly graphical user interface which proves to be better when compared to the existing system.
- It gives appropriate access to the authorized users depending on their permissions.
- If effectively overcomes the delay in communications.
- Updating of information becomes so easier.
- System security, data security and reliability are the striking features.
- The system has adequate scope for modification in future if it is necessary.

Bibliography

Bibliography

- www.codeproject.com
- www.stackoverflow.com
- www.social.msdn.microsoft.com