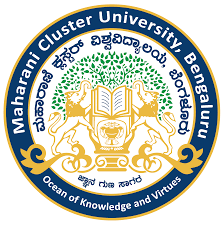
**GOVERNMENT OF KARNATAKA**

**SMT. V.H.D CENTRAL INSTITUTE OF HOME SCIENCE**

AFFILIATED TO MAHARANI CLUSTER UNIVERSITY

SHESHADRI ROAD, BANGALORE – 560001

****

**PROJECT REPORT**

**“ART GALLLERY”**

**A PARTIAL FULFILMENT OF BACHELOR OF COMPUTER APPLICATION PROJECT**

* **SUBMITTED BY:**

**PAVITHRA N (20US1087)**

**PAAVANA M RAO (20US1084)**

**SUSHMITHA A (20US1134)**

* **GUIDED BY:**

**Prof. GIRISH**

**CERTIFICATE**

This is to certify that the project entitled **“ART GALLERY”** submitted to Maharani Cluster University in Partial Fulfilment for the degree of bachelor of computer application in computer science is a bona fide original work carried out by **PAVITHRA N(20US1087), PAAVANA M (20US1084), SUSHMITHA A(20US1134)** under the guidance and Supervision during the year 2022-2023.

The project report as it satisfies in the academic requirement in respect of project work prescribed for BCA.

**Project Guide Head of Department**

**S G H Girish Prof Sumanth.S**

**Examiners Date**

**DECLARATION**

We **PAVITHRA N(20US1087), PAAVANA M RAO(20US1084), SUSHMITHA A(20US1134)** do here by declare that the project entitled **“ART GALLERY”** bona fide work carried out by us under the guidance of **Prof Girish**

This project, as presented in this report, is our original work and has not been presented for any other university award. This project has been submitted as Fulfillment of requirements for the degree of Bachelor of computer application of Bangalore central university.

**Place :** Bengaluru **Signature of the student Date:**

**ACKNOWLEDGEMENT**

I take this opportunity to express our deep sense of gratitude to our honorable, Principal H.N Vijayalakshmi Smt. V.H.D Central Institute of Home Science College for providing excellent academic climate in the college that made this endeavors possible.

My whole hearted admiration and deep sense of gratitude to our HOD of Computer Science Prof. Sumath S for his inspiration, valuable guidance encouragement, suggestions and overall help throughout for this successful completion of our project.

I am thankful to our beloved guide Prof. Girish of Computer Science who has been very kind and supportive during course of our project.

I would also like to express sincere and humble gratitude to our respectful lecturers and non-teaching staff of our Computer Science department.

I thank our beloved parents and all those who have directly or indirectly helped me.

|  |  |
| --- | --- |
| **1** | **INTRODUCTION** |
| 1.1 | Objective of the Project |
| 1.2 | Purpose of the Project |
| 1.3 | Scope of the Project |
| 1.4 | Benefits of the Project |
| 1.5 | Hardware and Software Requirements |
| 1.6 | Limitations of the Project |
| 1.7 | Project Life Cycle |
| **2** | **COMPONENTS ASSIGNED** |
| 2.1 | Existing System |
| 2.2 | Proposed System |
| **3** | **SYSTEM DEVELOPMENT LIFE CYCLE** |
| 3.1 | Introduction |
| 3.2 | Our Project Needed |
| 3.3 | Waterfall Model |
| **4** | **REQUIREMENT ANALYSIS** |
| 4.1 | Analysis of Study |
| 4.2 | Feasibility of Study |
| 4.3 | User Requirements |
| **5** | **SYSTEM DESIGN** |
| 5.1 | Data Flow Diagram |
| 5.2 | ER – Diagram |
| **6** | **IMPLEMENTATION** |
| **7** | **TESTING** |
| **8** | **MAINTAINANCE** |
| **9** | **SCREENSHOTS OF FORMS (Output)** |
| **10** | **CONCLUSION AND SCOPE** |
| **11** | **BIBLIOGRAPHY** |

**INDEX**

**CHAPTER – 01**

**INTRODUCTION**

Online **Art Gallery is** an online application, which is used to display and sell art works of artist irrespective of their nationality, gender, and other narrow consideration, through auction. Artist can register online for being a member in the art gallery and each artist can upload the digital copy of their art work under the respective categories. They can host their art work either for auction or for fixed price. The artist is liable to pay a fraction of the price of each art work to the web site to find the running fund for site.Art lovers have to go to the art exhibition to collect their favorite arts or painting. But now-a-days they are not getting enough time to go to the galleries and collect the arts and paintings.

* 1. **Objective of the Project**

Sometimes it will be difficult to maintain the details of the artists and their paintings details through the pen paper method. The art gallery management is the application that allows the art gallery owners to maintain the details of the artists and the details of their paintings with great ease.

* 1. **Purpose of the Project**
* **The primary purpose of an art gallery is to nurture visual artists, promote their work, and expose them to the public, collectors, media, and cultural institutions. Furthermore, the gallery works tirelessly and strategically to advance the artists careers and establish them in the professional art world both locally and globally.**
* To draw attention to and position their artists, galleries must organize in-person and online exhibitions of their work.
  1. **Scope of the Project**

This software project is aimed to create a platform where artists, exhibition organizers and customers can interact. Artists can display their artistic skills to attract the potential customers .The platform manages and displays exhibition details so that the art lovers can know about the upcoming exhibitions. The artists can maintain their stock of uploaded paintings details. The customers and admin can see the details of all the paintings by all the artists. An artist can see the stock of his own paintings. Customers can see the purchasing transactions done by them. The admin can see the purchase details of all the customers and evaluate the sales.

* 1. **Benefits of the Project**
* To reduce the hectic of maintaining the record of inventories.
* To reduce the cumbersome job of maintaining several documents.
* It will eliminate the delays in the generation of reports that which item has sold to whom.
* Searching will become more efficient and faster. It will also provide assurance to the customer that they can buy the art they like there would be no pressure.
* Overall, it will reduce the cost and time of the customer.
  1. **Hardware and Software Requirements**

**Software Requirements**

Operating System : Windows 10 Pro.

Language : VB.NET

Technologies : .NET framework

Data Bases : Microsoft Sql Server 2000

IDE : Visual Studio 2010

**Hardware Requirements:**

Processor : Any Processor above 500 MHz.

RAM : 128Mb.

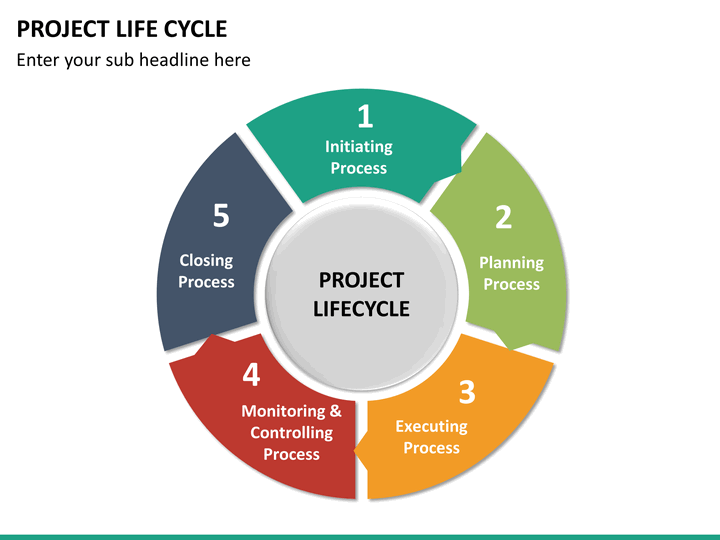
Hard Disk : 10 Gb.

Compact Disk : 650 Mb.

Input device : Standard Keyboard and Mouse.

Output device : High Resolution Monitor**.**

* 1. **Limitations**
* Off the shelf options/subscription based packages may not offer a theme in keeping with the style of images.
* Technical knowledge required to build/customize website.
* Ongoing subscription costs (hosting plans, etc.).
  1. **Project Life Cycle**



Project life cycle consist of **4 phase**s.

They are :

1. **Phase 1 :**Initiating Process
2. **Phase 2 :**Planning Process
3. **Phase 3 :**Executing Process
4. **Phase 4 :**Monitoring and Controlling Process
5. **Phase 5 :**ClosingProcess

**CHAPTER – 02**

**COMPONENTS ASSIGNED**

**2.1. Existing System**

Customer can also register and they can browse art works that are arranged in different categories scientifically. Each Customer can create their own gallery to see his favorite art works with out much difficult. And each user has the right to purchase an art work using the integrated payment gateway and participate in auction by submitting their bids. Qualified bidder should remit the amount using payment gateway and after each valid payment the art work will be shipped within some days.

The existing system has some disadvantages. They are as below:

* Lack of security data.
* More man power.
* Its time consuming.
* Consumes large volume of pare work.
* Needs manual calculations.
* No direct role for the higher officials.
* Damage of machines due to lack of attention.

To avoid all these limitations and make the working more accurately the system needs to be computerized in the effective manner

**2.2. Proposed System**

The artists can have their pencil sketches, glass paintings, oil paintings and many other forms of paintings that can be displayed in art gallery. Some of the art galleries will be having the paintings of some years ago. But at some point of time if the artist’s details are required, then you can get it easily if this application is used by the owner of the art gallery. Details of any artists and details of their paintings can be obtained with ease in just one mouse click. Some of the features that it can include in this online art gallery system are:

* **Artists and paintings database management:**The details of the artists like name, address, paintings, contact can be stored easily through this application.
* **Paintings can be sold**: The details of the paintings that are sold on each day can be stored easily through this application
* **Sales details**: The details of the sales that has taken place on a particular day can be obtained in just one mouse click through this application.

**Benefits of Proposed System:**

When the existing system is changed to proposed system, many benefits will be getting. Some of them are:

* Reduce time delay in processes (high speed more than manual processes)
* Reduce costs for the paper and stationary
* Reduce the record storage area
* Reduce redundancy of the data

To update the information, it just needs to change in one area and the rest will take care of the proposed system.

The reports can be generated quickly

Can be produce report which is daily, weekly and monthly of order record, total income and etc.

Can easy backup deal with storage data (mark, testing result and assignment)

Can search student ID or student name eg. Grade, Attendance and etc.

Smooth data flow between processes

Increase security for the information

Increase more privacy for the middle and top management

Calculation is more accurate

Easier to know detail of income (Student member and fee) and enquire about school from student

Ease of use and understand for school staff, teaching staff (if you can work this project, will follow user manual)

**CHAPTER – 03**

**SYSTEM DEVELOPMENT LIFE CYCLE**

**3.1. Introduction:**

A software development methodology is a framework that is used to structure, plan, and control the[process of developing](http://en.wikipedia.org/wiki/Software_development_process) an [information system](http://en.wikipedia.org/wiki/Information_system) definition of specific deliverables and artifacts that are created and completed by a project team to develop or maintain an application. A wide variety of such frameworks have evolved over the years, each with its own recognized strengths and weaknesses. One software development methodology framework is not necessarily suitable for use by all projects. Each of the available methodology frameworks are best suited to specific kinds of projects, based on various technical, organizational, project and team considerations.

There are a number of software development approaches that have been used since the origin of information technology. These software development approaches are: -

* Waterfall Approach
* Prototyping Approach
* Incremental Approach
* Spiral Approach

**3.2 Our Project Needed**

According to our project requirements, we are going to use the Waterfall Model to develop our software. Through this model we would be able to produce well-documented maintainable software in a manner that is very predictable and easy to understand. First of all the feasibility study is done. Once that part is over the requirement analysis and project planning begins. If system exists one and modification and addition of new module is needed, analysis of present system can be used as basic model.

The design starts after the requirement analysis is complete and the coding begins after the design is complete. Once the programming is completed, the testing is done. In this model the sequence of activities performed in a software development project are: -

* Requirement Analysis and Specification
* Design Implementation and testing
* Maintenance
  1. **WATERFALL MODEL**

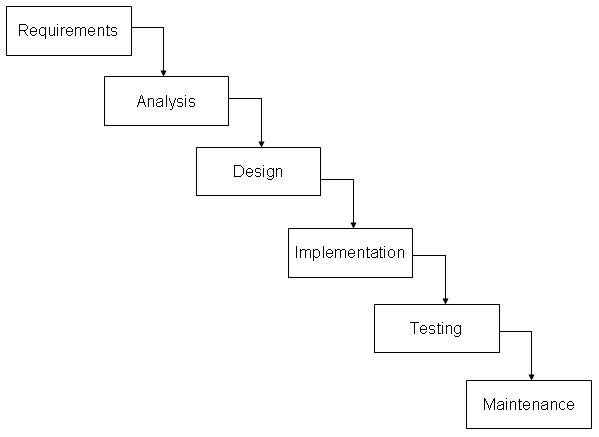


Fig: WATERFALL MODEL

**CHAPTER – 04**

**REQUIREMENT ANALYSIS**

**4.1 Analysis of Study**

System Analysis is a management technique, which helps us in designing a new system or improving an existing system

After analyzing the requirements of the task to be performed, the next step is to analyze the problem and understand its context. The first activity in the phase is studying the existing system and other is to understand the requirements and domain of the new system. Both the activities are equally important, but the first activity serves as a basis of giving the functional specifications and then successful design of the proposed system. Understanding the properties and requirements of a new system is more difficult and requires creative thinking and understanding of existing running system is also difficult, improper understanding of present system can lead diversion from solution.

Analysis study is presented in the form of Software Requirement Specification. Review of SRS is conducted to determine the suitability and the adequacy of the software requirement. The review addresses the following questions/issues:

* Are the requirements appropriate to the user needs or project objectives?
* Are the requirements complete?
* Are the requirements defined unambiguously?
* Are the requirements self-consistent?
* Is every requirements

One must know what the problem is before it can be solved. General approaches for determining user requirements are:

* **Preliminary investigation –** asking general questions
* **Analysis of existing system –** getting information from existing system.

**4.2 Feasibility of Study** is a test of system proposal according to its workability

Feasibility is a test of system proposal according to its workability impact on the organization, ability to meet user needs & effective use of resources. Three key considerations are involved in the feasibility analysis are as under

* Operational Feasibility

To find out whether the system will be functional after its development and installation?

#### Technical Feasibility

To examine out whether the current technology is sufficient for the development of the system. An estimate is made of whether the identified user needs may be satisfied using current software and hardware technologies.

* Economic feasibility

This analysis is the most frequently used method for comparing the cost with benefit or income that is expected from the developed system.

**4.3 User Requirements:**

Here, the focus is on specifying what has been found giving analysis such as representation, specification languages and tools, and checking the specifications are addressed during this activity.

The requirement phase terminates with the production of the validate SRS document. Producing the SRS document is the basic of this phase.

**4.3.1 Problem/Requirement Analysis:**

The process is order and more nebulous of the two, deals with understand the problem, the goal and constraints.

**4.3.2 Software Requirements Specification**

**What is SRS ?**

Software Requirements Specification (SRS) is the starting point of the software developing activity. As system grows more complex it becomes evident that the goal of the entire system cannot be easily comprehended. Hence the need for the requirement phase arose. The software project is initiated by the client needs. The SRS is the means of translating the ideas of the minds of clients (the input) into a formal document (the output of the requirement phase).

The SRS phase consist of two basic activities:

**Problem/Requirement Analysis:**

The process is order and more nebulous of the two, deals with understand the problem, the goal and constraints.

**Requirement Specification:**

Here, the focus is on specifying what has been found giving analysis such as representation, specification languages and tools, and checking the specifications are addressed during this activity.

The requirement phase terminates with the production of the validate SRS document. Producing the SRS document is the basic of this phase.

**Characteristics of a Good SRS Document**

Some of the identified desirable qualities of the SRS documents are following:-

**Concise**: The SRS document should be concise and at the same time unambiguous.

**Structured:** The SRS document should be well structured.

**Black-box view**: It should only specify what the system should do and refrain from

stating how to do.

**Conceptual integrity**: The SRS document should exhibit conceptual integrity so that the

reader can easily understand the contents.

**Response to undesired events**: The document should characterize acceptable responses

to undesired events.

**Verifiable:** All requirements of the system as documented in SRS document should be

verifiable. This means that it should be possible to determine whether or not requirementshave been met in an implementation.

**CHAPTER – 05**

**SYSTEM DESIGN**

**5.1 DATAFLOW DIAGRAMS**

Data flow diagram is used to decrease analysis the movement of data through a system store of data in the system. Data flow diagrams are the central tool basing on which components are developed.

The transformation of data from input to output, through process may be describe logically and independently of physically components associated with the system. They are called logical data flow diagrams. In contrast physical data flow diagrams show the actual implementation and movement of data between people, Department, and work station.

The data flow diagram show functional composition of the system. The first level of conceptual level in context diagram is flowed by the description of input and output for each of entities the next level of DFD is level 0, which shows the main functions in the system. Level 0 is followed by the description of the main functions. The main function further broken into functions and sub functions.

**5.1 Purpose/Objective**:

The purpose of data flow diagrams is to provide a semantic bridge between users and system developers.

The diagrams are:

* Graphical, eliminating thousands of words;
* Logical representations, modeling WHAT a system does, rather than physical models showing HOW it does it;
* Hierarchical, showing systems at any level of detail and Jargon less, allowing user understanding and reviewing.

The goal of data flow diagramming is to have a commonly understood model of a system.

The diagrams are the basis of structured systems analysis. Data flow diagrams are supported by other techniques of structured systems analysis such as data structure diagrams, data dictionaries, and procedure-representing techniques such as decision tables, decision trees, and structured English.

The objective of Data flow diagrams is avoiding the cost of user/developer

misunderstanding of a system, resulting in a need to redo systems or in not using the system.

**5.2 DFD Symbols:** In the DFD, there are four symbols shown in figure below:

* **A Square** defines a source (originator) or destination of data pipeline through which information flows.

* **An Arrow** identifies data flow data in motion. It is a pipeline through which information flows.
* **A Circleor a "bubble"** (some people use an oval bubble) represents a process that transforms incoming data flow into outgoing data flow.
* **An Open Rectangle** is a data store – data at rest, or a temporary repository of data.

**DFD Level 0** :

****

**4.2. E-R Diagrams:**

**ENTITY RELATIONSHIP DIAGRAM**

An entity relationship diagram is a graphical representation of an organization’s data storage

requirements. Entity relationship diagrams are abstractions of the real world, which simplify

the problem to be solved while retaining its essential features. Entity relationship diagrams

are used to identify the data that must be captured, stored and retrieved in order to support

the business activities performed by an organization; and identify the data required to derive

and report on the performance measures measures that an organization should be

monitoring Entity relationship diagrams have three different components:

**Entities**

An entity is an object that exists and is distinguishable from other objects. An entity may be

concrete (a person or a book, for example) or abstract (like a holiday or a concept). In short,

anything, which an organization needs to store data about Entities, represents collections of

things

**Attributes**

Entities are further described by their attributes (sometimes called data elements). These are

the smallest units of data that can be described in a meaningful manner.

**Relationships**

A relationship is an association between several entities. There are

potentially three types of relationship, which can exist between two different entities:

**One-to-One Relationships**

This type of relationship takes place when a single occurrence of an entity is related to just

one occurrence of a second entity.

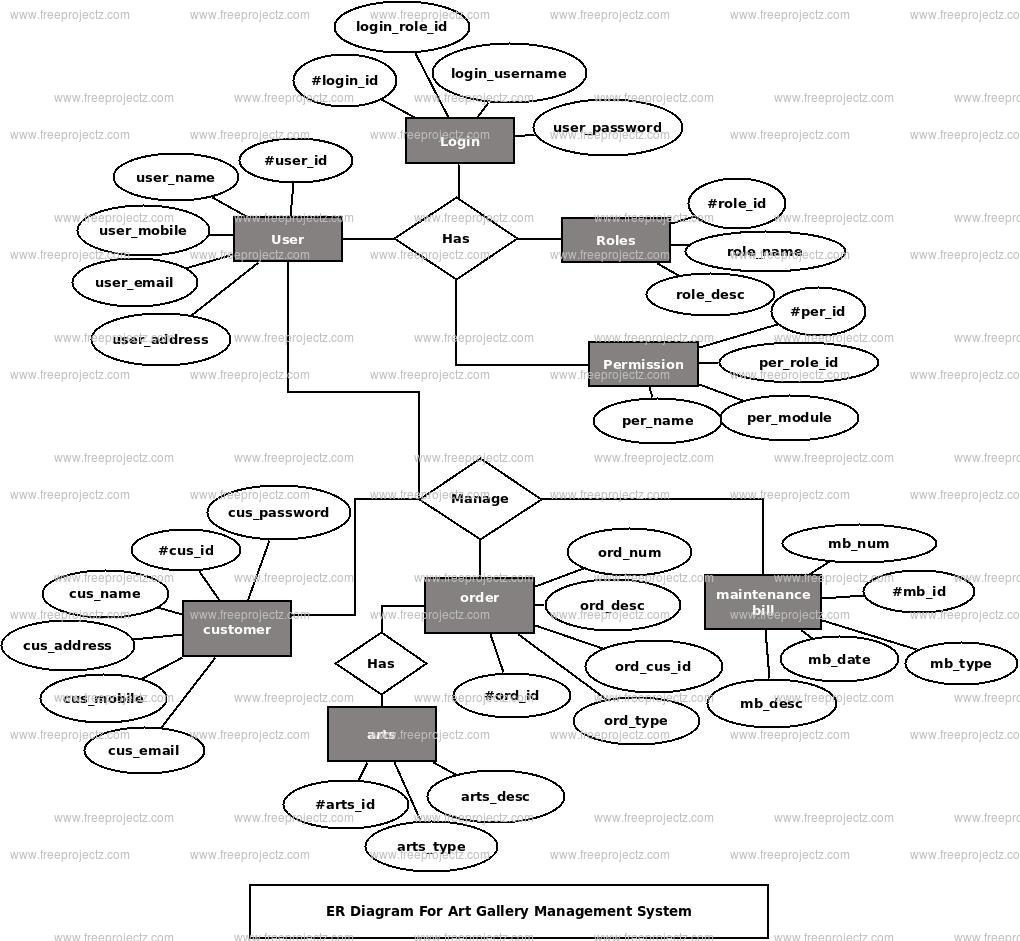
**One-to-Many Relationships**

This type of relationship takes place when a single occurrence of an entity is related to many

occurrences of a second entity.

**Many-to-Many Relationships**

This type of relationship takes place when many occurrences of an entity are related to many occurrences of a second entity.



**SQL COMMAND METHODS**

**SqlCommand.BeginExecuteNonQuery Method:**

Initiates the asynchronous execution of the Transact-SQL statement or procedure that is described by this SQL Command. This member is overloaded. For complete information about this member, including syntax, usages, and examples, click a name in the overload list.

**SqlCommand.BeginExecuteReader Method:**

Initiates the asynchronous execution of the transact SQL statement or stored procedure that is described by this SQL Command, and retrieves one or more results sets from the server .This member is overloaded. For complete information about this member,including syntax, usages, and examples, click a name in the overload list.

**SqlCommand.Clone Method:**

Creates a new SqlCommand object that is a copy of the current instance.

**SqlCommand. DisposeMethod**:

This member is overloaded. For complete information about this member, including syntax, usage and. examples, click a name in the overload list.

**SqlCommand.EndExecuteNonQueryMethod:**

Finishes asynchronous execution of transact-SQL statement.

**SqlCommand.EndExecuteReaderMethod :**

Finishes asynchronous execution of transact-SQL statement,returning the requested SqlDataReader.

**SqlCommand.ExecuteNonQuery Method**:

Executes a transact- SQL statement against the connection and returns the number of rows affected.

**SqlCommand.Executereader Method:**

This member is overloaded. For complete information about this member,including syntax, usages, and examples, click a name in the overload list.

**SqlCommand.ExecuteScalar Method**:

Executes the query, and returns the first columns of the first row in the result set returned by the query, additional columns or rows are ignored.

**Data AdapterClass:**

Represents the set of data commands and a database connection that are used to fill the data set and update the data source.

**Retrieving Data Using the Data Reader:**

You can use the ado.net data reader to retrieve a read-only,forward-only stream of data from a database. Results arereturned as the query executes, and are stored in the networkbuffer on the client until you request them using the readmethod of the data reader. Using the data reader can increaseapplication performance both by retrieving data as soon as it isavailable, rather than waiting for the entire results of the queryto be returned, and (by default) storing only one row at a time inmemory reducing system overhead.

What is Dataset?

A Dataset is an in memory representation of data loaded from any data source. Even though the most common data source is database, we can use dataset to load data from other data

sources including XML files etc. in this article, we will talk about the role of data set in manipulating data from database. In .NET, a dataset is a class provided by the .NET Framework. The dataset class exposes several properties and methods that can be used to retrieve, manipulate and save data from various data sources.

**CHAPTER-06**

**IMPLEMENTATION**

**Software Interfaces uses in the software are:**

* Operating System: Windows 10 pro
* VB.NET Version 8.0
* Microsoft SQL server 2000

**6.1 OPERATING SYSTEM**

**WINDOWS 10 PRO:**

Windows 10 is the most recent version of the Microsoft Windows operating system. There have been many different versions of Windows over the years, including **Windows 8** (released in 2012), **Windows 7** (2009), **Windows Vista** (2006), and **Windows XP** (2001). While older versions of Windows mainly ran on desktop and laptop computers, Windows 10 is also designed to run equally well on **tablets**. An **operating system** manages all of the hardware and software on a computer. Without it, the computer would be useless. To learn more about how operating systems work

**6.2 LANGUAGES**

* Vb.net
* Features of SQL 2000
* Normalization

**6.2.1 Introduction to Vb.net**

Visual Basic .NET (VB.NET) is a re-engineering of this venerable language, which departs in

significant ways from earlier versions of Visual Basic. VB.NET has a number of features that

help it retain backwards compatibility with Visual Basic 6 (VB6). Other features have been

added specifically to adapt Visual Basic to object-oriented programming and to the .NET

platform.

VB.NET provides support in the language to find bugs early in the development process. This makes for code that is easier to maintain and programs that are more reliable. VB.NET does not support many features available in other languages (e.g., pointers) that make for unsafe code. VB.NET is considered safe because it provides support in the language to find bugs early in the development process. This makes for code that is easier to maintain and programs that are more reliable.

VB.NET provides full support for object-oriented programming. This book will explain not only how to write object-oriented programs, but will explain why object-oriented programming has become so popular. The short answer is this: programs are becoming increasingly complex, and object-oriented programming techniques help you manage that complexity.

**6.2.2 Features of VB.NET**

Visual Basic.NET has many new and improved language features such as inheritance,

interfaces, and overloading that make it a powerful object-oriented programming language.

As a Visual Basic developer, we can now create multithreaded scalable applications using

explicit multithreading. Other new language features in Visual Basic .NET include structured

exception handling, custom attributes, and common language specification (CLS) compliance.

**Common Language Specification**

The CLS is a set of rules that standardizes such things as data types and how objects are exposed and interoperate. Visual Basic .NET adds several features that take advantage of the CLS. Any CLS-compliant language can use the classes, objects, and components you create in Visual Basic .NET. as a Visual Basic user, can access classes, components, and objects from other CLS-compliant programming languages without worrying about language-specific differences such as data types. CLS features used by Visual Basic .NET programs include assemblies, namespaces, and attributes. These are the new features to be stated briefly:

**Inheritance**

Visual Basic .NET supports inheritance by allowing you to define classes that serve asthe basis for derived classes. Derived classes inherit and can extend the properties andmethods of the base class. They can also override inherited methods with newimplementations.

**Exception Handling**

Visual Basic .NET supports structured exception handling, using an enhanced version of the Try…Catch…Finally syntax supported by other languages such as C++.

Structured exception handling combines a modern control structure (similar to Select Case or While) with exceptions, protected blocks of code, and filters.

**Overloading**

Overloading is the ability to define properties, methods, or procedures that have the same name but use different data types. Overloaded procedures allow you to provide as many implementations as necessary to handle different kinds of data, while givingthe appearance of a single, versatile procedure.

**Constructors and Destructors**

Constructors are procedures that control initialization of new instances of a class. Conversely, destructors are methods that free system resources when a class leaves

scope or is set to Nothing. Visual Basic .NET supports constructors and destructors

using the Sub New and Sub Finalize procedures.

* **Data Types**

Visual Basic .NET introduces three new data types. The Char data type is an unsigned 16-bit quantity used to store Unicode characters. It is equivalent to the .NET Framework System. Char data type. The Short data type, a signed 16-bit integer, was named Integer in earlier versions of Visual Basic. The Decimal data type is a 96-bit signed integer scaled by a variable power of 10.

• **Interfaces**

Interfaces describe the properties and methods of classes, but unlike classes, do not provide implementations. The Interface statement allows you to declare interfaces, while the Implements statement lets you write code that puts the items described in the interface into practice.

**6.3 Introduction to Microsoft SQL Server 2000**

Microsoft SQL Server is a Structured Query Language (SQL) based, client/server relational database. Relational databases are the most effective among the different way to organize data in database. Relational database systems are am application of mathematical set theory to the problem of effectively organizing data. In a relational database data is collected into tables (called relations in relational theory).

Microsoft SQL Server (version 7.0) offers broad availability of solution tailored for business operations, data warehousing, electronic commerce and mobile computing. It provides a comprehensive platform that makes it easy to design, build, manage and use data warehousing solutions, which enable organizations to make effective business decisions based on timely and accurate information.

**6.3.1Features of SQL Server 2000**

Microsoft® SQL Server™ 2000 features include:

• **Internet Integration.**

The SQL Server 2000 database engine includes integrated XML support. It also has the scalability, availability, and security features required to operate as the data storage component of the largest Web sites. The SQL Server 2000 programming model is integrated with the Windows DNA architecture for developing Web applications, and SQL Server 2000 supports features such as English Query and the Microsoft Search Service to incorporate user-friendly queries and powerful search capabilities in Web applications.

* **Scalability and Availability**.

The same database engine can be used across platforms ranging from laptop computers running Microsoft Windows® 98 through large, multiprocessor servers running Microsoft Windows 2000 Data Center Edition. SQL Server 2000 Enterprise Edition supports features such as federated servers, indexed views, and large memory support that allow it to scale to the performance levels required by the largest Websites.

* **Enterprise-Level Database Features**.

The SQL Server 2000 relational database engine supports the features required to support demanding data processing environments. The database engine protects data integrity while minimizing the overhead of managing thousands of users concurrently modifying the database. SQL Server 2000 distributed queries allow you to reference data from multiple sources as if it were a part of a SQL Server 2000 database, while at the same time, the distributed transaction support protects the integrity of any updates of the distributed data. Replication allows you to also maintain multiple copies of data, while ensuring that the separate copies remain synchronized.

* **Ease of installation, deployment, and use**.

SQL Server 2000 includes a set of administrative and development tools that improve upon the process of installing, deploying, managing, and using SQL Server across several sites. SQL Server 2000 also supports a standards-based programming model integrated with the Windows DNA, making the use of SQL Server databases and data warehouses a seamless part of building powerful and scalable systems

* **Data warehousing**.

SQL Server 2000 includes tools for extracting and analyzing summary data for online analytical processing. SQL Server also includes tools for visually designing databases and analyzing data using English-based questions.

* **Online Restore:**

With SQL Server 2000, database administrators are able to perform a Restore operation while an instance of SQL server is running. Online restore improves the availability of SQL Server, because only the data being restored is unavailable. The rest of the database remains online and available.

* **Fast Recovery:**

A new fast recovery option improves availability of SQL server database. Administrator can reconnect to a recovering database after the transaction Log has been rolled forward.

* **SQL Server Enterprise Manager**

Microsoft® Management Console (MMC) is a tool that presents a common interface for managing different server applications in a Microsoft Windows® network. Server applications provide a component called an MMC snap-in that presents MMC users with a user interface for managing the server application. SQL Server Enterprise Manager is the Microsoft SQL Server™ MMC snap-in.

* **Overview of the SQL Server Tools**
* Microsoft® SQL Server™ 2000 includes many graphical and command prompt utilities
* that allow users, programmers, and administrators to:
* Administer and configure SQL Server.
* Determine the catalog information in a copy of SQL Server.
* Design and test queries for retrieving data.
* Copy, import, export, and transform data.
* Provide diagnostic information.
* Start and stop SQL Server.

***6.4 Normalization***

Normalization is one of the powerful methods used in the design process. It ensures that redundancy of the database is reduced to reasonable level. It ensures that the database is integrated by the way of reducing the redundancy. Hence it is the best method to design effective database. The normalized database is later converted to a physical database. Relations are normalized so that when relations in a database are to be altered during the life time of the database, we do not lose information or introduce inconsistency. The type of alterations normally needed for relations are:

Insertion of new data values to a relation. This should be possible without being forced to leave blank fields for some attributes.

* Deletion of a tuple – a row in a relation. This should be possible without losing vital information unknowingly.
* Updating or changing a value of an attribute in a tuple. This should be possible without exhaustingly searching all the tuples in the relation.

**FIRST NORMAL FORM:**

* This form is also called as flat file. A relation is said to be in the first normal form if there are no composite attributes, and every attribute is single and describes one property.

**SECOND FORM NORMAL:**

* A relation is said to be in the second normal form if it is in the first normal form and non-key attributes are functionally dependent on the key attribute(s).

**THIRD NORMAL FORM:**

* A third normal form will be needed where all attributes in a relation tuple are not functionally dependent only on the key attribute.
* Third normal form (3NF) requires that there are no functional dependencies of non-key attributes on something other than a candidate key.
* A table is in 3NF if all of the non-primary key attributes are mutually independent.
* There should not be transitive dependencies.

**CODING**

**APP.CONFIGURATION**

<?xml version="1.0" encoding="utf-8" ?>

<configuration>

<configSections>

</configSections>

<connectionStrings>

<add name="ArtGallery.My.MySettings.artConnectionString" connectionString="Data Source=.\SQLEXPRESS;AttachDbFilename=|DataDirectory|\art.mdf;Integrated Security=True;Connect Timeout=30;User Instance=True"

providerName="System.Data.SqlClient" />

</connectionStrings>

<system.diagnostics>

<sources>

<!-- This section defines the logging configuration for My.Application.Log -->

<source name="DefaultSource" switchName="DefaultSwitch">

<listeners>

<add name="FileLog"/>

<!-- Uncomment the below section to write to the Application Event Log -->

<!--<add name="EventLog"/>-->

</listeners>

</source>

</sources>

<switches>

<add name="DefaultSwitch" value="Information" />

</switches>

<sharedListeners>

<add name="FileLog"

type="Microsoft.VisualBasic.Logging.FileLogTraceListener, Microsoft.VisualBasic, Version=8.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a, processorArchitecture=MSIL"

initializeData="FileLogWriter"/>

<!-- Uncomment the below section and replace APPLICATION\_NAME with the name of your application to write to the Application Event Log -->

<!--<add name="EventLog" type="System.Diagnostics.EventLogTraceListener" initializeData="APPLICATION\_NAME"/> -->

</sharedListeners>

</system.diagnostics>

</configuration>

**MODULE**

Imports System.Data.SqlClient

Module Module1

Public Conn As New SqlConnection("Data Source=.\SQLEXPRESS;AttachDbFilename=C:\Users\Intel\Documents\Visual Studio 2010\Projects\ArtGallery\ArtGallery\art.mdf;Integrated Security=True;Connect Timeout=30;User Instance=True")

Public SqlStr, q1Var, q2Var As String

Public colVar, rowVar, I, J, inVar, outVar, L1 As Long

Public vTypeVar As Byte

Public con1 As Boolean

Public sinChar As String

Enum CtrlType

TextBox = 1

ComboBox = 2

End Enum

Public Sub ClearTxtControls(ByRef frm As Object, ByRef ControlType As CtrlType, Optional ByRef Tagstr As Object = Nothing)

Dim Contrl As Object

For Each Contrl In frm.Controls

If Not (IsNothing(Tagstr)) Then

If Trim(UCase(Contrl.Tag)) = Trim(UCase(Tagstr)) Then

Contrl.Text = ""

Exit For

End If

Else

Select Case ControlType

Case CtrlType.ComboBox

If TypeOf Contrl Is System.Windows.Forms.ComboBox Then Contrl.Text = ""

Case CtrlType.TextBox

If TypeOf Contrl Is System.Windows.Forms.TextBox Then Contrl.Text = ""

End Select

End If

Next Contrl

Contrl = Nothing

End Sub

Public Function CheckChar(ByRef CharString As Object) As String

Dim I

L1 = Len(CharString)

CharString = UCase(Left(CharString, 1)) & Right(CharString, L1 - 1)

For I = 1 To L1

If con1 = True Then CharString = Mid(CharString, 1, I - 1) & UCase(Mid(CharString, I, 1)) & Mid(CharString, I + 1, L1)

sinChar = Mid(CharString, I, 1)

If sinChar = " " Then

con1 = True

Else

con1 = False

End If

Next I

CheckChar = CharString

End Function

Public Function CheckNum(ByRef KeyVar As String) As String

If Asc(KeyVar) = 8 Then

CheckNum = KeyVar : Exit Function

End If

If Asc(KeyVar) < 46 Or Asc(KeyVar) > 57 Then

CheckNum = Nothing

MsgBox("Please Enter Numbers Only")

Else

CheckNum = KeyVar

End If

If Asc(KeyVar) = 47 Then CheckNum = Nothing

End Function

Public Function forCur(ByRef CurAmt As Object) As Object

forCur = Format(CurAmt, "##,##,##0.00")

End Function

Public Function forDate(ByRef DateVar As Object) As Object

forDate = Format(DateVar, "dd/MMM/yyyy")

End Function

End Module

**START PAGE**

---------------------------------

Public Class StartPage

Dim I As Integer

Private Sub Timer1\_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Timer1.Tick

I = I + 1

If I > 20 Then

Timer1.Enabled = False

Me.Hide()

LoginForm1.Show()

End If

End Sub

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

Me.WindowState = FormWindowState.Maximized

End Sub

End Class

**LOGIN FORM**

Imports System.Data.SqlClient

Public Class LoginForm1

Private Sub OK\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles OK.Click

If Conn.State = ConnectionState.Open Then Conn.Close()

Conn.Open()

Dim Cmd0 As New SqlCommand("select \* from SignUpTab where username='" & UCase(UsernameTextBox.Text) & "' and password='" & PasswordTextBox.Text & "'", Conn)

Dim D1 As SqlDataReader = Cmd0.ExecuteReader()

If D1.HasRows Then

adminMain.Show()

Me.Hide()

If Conn.State = ConnectionState.Open Then Conn.Close()

Else

MsgBox("Username or Password is not correct please check")

End If

UsernameTextBox.Clear()

PasswordTextBox.Clear()

End Sub

Private Sub Cancel\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Cancel.Click

UsernameTextBox.Text = ""

PasswordTextBox.Text = ""

End Sub

Private Sub LoginForm1\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

PasswordTextBox.UseSystemPasswordChar = True

' Me.WindowState = FormWindowState.Maximized

End Sub

--------------------------------------

Private Sub CheckBox1\_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles CheckBox1.CheckedChanged

If CheckBox1.CheckState = CheckState.Checked Then

PasswordTextBox.UseSystemPasswordChar = False

Else

PasswordTextBox.UseSystemPasswordChar = True

End If

End Sub

Private Sub BunifuImageButton1\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles BunifuImageButton1.Click

Me.Hide()

Form1.Show()

End Sub

Private Sub BunifuImageButton2\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles BunifuImageButton2.Click

Me.Hide()

Form2.Show()

End Sub

Private Sub BunifuImageButton3\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles BunifuImageButton3.Click

Me.Hide()

ArtGallery.Show()

End Sub

Private Sub PictureBox2\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles PictureBox2.Click

Me.Dispose()

End Sub

End Class

**ART GALLERY**

Public Class ArtGallery

Private Sub RegistrationToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles RegistrationToolStripMenuItem.Click

Panel1.Controls.Clear()

With ArtistReg

.TopLevel = False

Panel1.Controls.Add(ArtistReg)

.BringToFront()

.Show()

End With

End Sub

Private Sub SignInToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles SignInToolStripMenuItem.Click

Panel1.Controls.Clear()

With ArtistSignin

.TopLevel = False

Panel1.Controls.Add(ArtistSignin)

.BringToFront()

.Show()

End With

End Sub

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

Me.WindowState = FormWindowState.Maximized

End Sub

Private Sub BillingToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)

Billing.Show()

End Sub

Private Sub ExitToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ExitToolStripMenuItem.Click

MsgBox("Are you sure you want to exit")

Me.Hide()

LoginForm1.Show()

End Sub

Private Sub SignInToolStripMenuItem2\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)

Me.Close()

custsignin.Show()

End Sub

Private Sub SignInToolStripMenuItem1\_Click\_1(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles SignInToolStripMenuItem1.Click

Panel1.Controls.Clear()

With custsignin

.TopLevel = False

Panel1.Controls.Add(custsignin)

.BringToFront()

.Show()

End With

End Sub

Private Sub SIGNUPToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles SIGNUPToolStripMenuItem.Click

Panel1.Controls.Clear()

With custsignup

.TopLevel = False

Panel1.Controls.Add(custsignup)

.BringToFront()

.Show()

End With

End Sub

Private Sub PictureBox1\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles PictureBox1.Click

Me.Dispose()

LoginForm1.Show()

End Sub

Private Sub Panel1\_Paint(ByVal sender As System.Object, ByVal e As System.Windows.Forms.PaintEventArgs) Handles Panel1.Paint

End Sub

End Class

**ARTIST REGISTRATION**

Imports System.Data.SqlClient

Imports System.Text.RegularExpressions

Public Class ArtistReg

Private Sub ButNew\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ButNew.Click

ClearTxtControls(Me, 1)

TextBox1.Focus()

ButNew.Enabled = False

ButSave.Enabled = True

End Sub

Private Sub ButSave\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ButSave.Click

SaveRecord()

ButNew.Enabled = True

ButSave.Enabled = False

End Sub

Sub SaveRecord()

If (TextBox1.Text = "" Or TextBox2.Text = "" Or TextBox3.Text = "" Or TextBox5.Text = "" Or TextBox6.Text = "" Or TextBox7.Text = "") Then

MsgBox("Please Enter The Necessary Details")

Exit Sub

End If

If Conn.State = ConnectionState.Open Then Conn.Close()

Conn.Open()

Dim Cmd0 As New SqlCommand("select artistname from artistreg where artistname='" & UCase(TextBox1.Text) & "'", Conn)

Dim D1 As SqlDataReader = Cmd0.ExecuteReader()

If D1.HasRows Then

MsgBox("This record already exists")

If Conn.State = ConnectionState.Open Then Conn.Close()

Exit Sub

End If

Dim command As New SqlCommand("insert into artistreg(artistname,pword,addrs,dob,mob,email,qualif) values(@artistname,@pword,@addrs,@dob,@mob,@email,@qualif)", Conn)

' Dim ms As New MemoryStream

' PictureBox1.Image.Save(ms, PictureBox1.Image.RawFormat)

command.Parameters.Add("@artistname", SqlDbType.NVarChar).Value = TextBox1.Text

command.Parameters.Add("@pword", SqlDbType.NVarChar).Value = TextBox2.Text

command.Parameters.Add("@addrs", SqlDbType.NVarChar).Value = TextBox3.Text

command.Parameters.Add("@dob", SqlDbType.NVarChar).Value = DateTimePicker2.Text

command.Parameters.Add("@mob", SqlDbType.NVarChar).Value = TextBox5.Text

command.Parameters.Add("@email", SqlDbType.NVarChar).Value = TextBox6.Text

command.Parameters.Add("@qualif", SqlDbType.NVarChar).Value = TextBox7.Text

If Conn.State = ConnectionState.Open Then Conn.Close()

Conn.Open()

If command.ExecuteNonQuery() = 1 Then

MsgBox("Saved Successfully!!!")

Else

MsgBox("Error!!!")

End If

End Sub

Private Sub Butclose\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Butclose.Click

Me.Close()

End Sub

Private Sub TextBox1\_LostFocus(ByVal sender As Object, ByVal e As System.EventArgs) Handles TextBox1.LostFocus

If Not Regex.Match(TextBox1.Text, "^[a-z. ]\*$", RegexOptions.IgnoreCase).Success Then

MsgBox("please enter alpha text only!")

TextBox1.Focus()

TextBox1.Clear()

End If

End Sub

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

Me.WindowState = FormWindowState.Maximized

End Sub

Private Sub TextBox6\_Validating(ByVal sender As Object, ByVal e As System.ComponentModel.CancelEventArgs) Handles TextBox6.Validating

Dim tbox As TextBox = DirectCast(sender, TextBox)

If Regex.IsMatch(tbox.Text, "^([a-zA-Z0-9\_\-\.]+)@((\[[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.)|(([a-zA-Z0-9\-]+\.)+))([a-zA-Z]{2,4}|[0-9]{1,3})(\]?)$") = True Then

TextBox7.Focus()

ErrorProvider1.SetError(Me.TextBox6, "")

e.Cancel = False

Else

ErrorProvider1.SetError(Me.TextBox6, "Not a valid email")

e.Cancel = True

End If

End Sub

Private Sub DateTimePicker2\_Validating(ByVal sender As Object, ByVal e As System.ComponentModel.CancelEventArgs) Handles DateTimePicker2.Validating

Dim d As Integer

d = Year(DateTimePicker2.Value) - Year(DateTimePicker1.Value)

' MsgBox(d)

If d < 19 Then

MsgBox("Sorry you should be 18 years for applying for license")

Else

TextBox5.Focus()

End If

End Sub

Private Sub TextBox5\_Validating(ByVal sender As Object, ByVal e As System.ComponentModel.CancelEventArgs) Handles TextBox5.Validating

'Dim phoneNumber As New Regex("\d{3}\d{3}\d{4}")

Dim phoneNumber As New Regex("^([6-9]{1})([0-9]{9})")

TextBox5.MaxLength = 10

If phoneNumber.IsMatch(TextBox5.Text) Then

TextBox6.Focus()

Else

MsgBox("Not Valid Phone Number")

TextBox5.Text = ""

TextBox5.Focus()

End If

End Sub

Private Sub TextBox1\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox1.TextChanged

End Sub

Private Sub PictureBox2\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles PictureBox2.Click

Me.Dispose()

LoginForm1.Show()

End Sub

Private Sub Label8\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)

End Sub

End Class

**ARTIST SIGNIN**

Imports System.Data.SqlClient

Public Class ArtistSignin

Private Sub OK\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles OK.Click

If Conn.State = ConnectionState.Open Then Conn.Close()

Conn.Open()

Dim Cmd0 As New SqlCommand("select \* from ArtistReg where artistname='" & UCase(UsernameTextBox.Text) & "' and pword='" & PasswordTextBox.Text & "'", Conn)

Dim D1 As SqlDataReader = Cmd0.ExecuteReader()

If D1.HasRows Then

Artists.Show()

Me.Hide()

If Conn.State = ConnectionState.Open Then Conn.Close()

Else

MsgBox("Username or Password is not correct please check")

End If

End Sub

Private Sub Cancel\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Cancel.Click

Me.Close()

End Sub

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

PasswordTextBox.UseSystemPasswordChar = True

Me.WindowState = FormWindowState.Maximized

End Sub

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

PasswordTextBox.Text = ""

UsernameTextBox.Text = ""

End Sub

Private Sub CheckBox1\_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles CheckBox1.CheckedChanged

If CheckBox1.CheckState = CheckState.Checked Then

PasswordTextBox.UseSystemPasswordChar = False

Else

PasswordTextBox.UseSystemPasswordChar = True

End If

End Sub

Private Sub PictureBox2\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles PictureBox2.Click

Me.Dispose()

End Sub

End Class

**ARTIST PAGE**

Public Class Artists

Private Sub VIEWALLCATEGORIESToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles VIEWALLCATEGORIESToolStripMenuItem.Click

Panel1.Controls.Clear()

With ViewCategory

.TopLevel = False

Panel1.Controls.Add(ViewCategory)

.BringToFront()

.Show()

End With

End Sub

Private Sub ADDPAINTINGSToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ADDPAINTINGSToolStripMenuItem.Click

' AddPaintings.Show()

Panel1.Controls.Clear()

With AddPaintings

.TopLevel = False

Panel1.Controls.Add(AddPaintings)

.BringToFront()

.Show()

End With

End Sub

Private Sub ExitToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)

MsgBox("are you sure you want to exit")

Me.Close()

ArtGallery.Show()

End Sub

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

Me.WindowState = FormWindowState.Maximized

End Sub

Private Sub PictureBox1\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles PictureBox1.Click

Me.Dispose()

End Sub

Private Sub Panel1\_Paint(ByVal sender As System.Object, ByVal e As System.Windows.Forms.PaintEventArgs) Handles Panel1.Paint

End Sub

End Class

**VIEW PAINTING CATEGORY**

Imports System.Data.SqlClient

Public Class ViewCategory

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

Me.Close()

End Sub

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

Me.WindowState = FormWindowState.Maximized

If Conn.State = ConnectionState.Open Then Conn.Close()

Conn.Open()

Dim DS1 As New DataSet

Dim adp As New SqlDataAdapter("Select catname,details From category order by catname", Conn)

adp.Fill(DS1)

DG1.DataSource = DS1.Tables(0)

If Conn.State = ConnectionState.Open Then Conn.Close()

End Sub

Private Sub DG1\_CellContentClick(ByVal sender As System.Object, ByVal e As System.Windows.Forms.DataGridViewCellEventArgs) Handles DG1.CellContentClick

End Sub

Private Sub PictureBox1\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles PictureBox1.Click

Me.Dispose()

End Sub

End Class

**ADD PAINTINGS**

Imports System.Data.SqlClient

Imports System.IO

Imports System.Text.RegularExpressions

Public Class AddPaintings

Dim pkVar As String

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

ClearTxtControls(Me, 1)

PictureBox1.Image = Nothing

' ComboBox1.Text = String.Empty

ComboBox1.SelectedIndex = -1

TextBox1.Focus()

ButNew.Enabled = True

ButSave.Enabled = True

End Sub

Private Sub Button3\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ButSave.Click

SaveRecord()

ButNew.Enabled = True

ButSave.Enabled = False

ButModify.Enabled = False

ButDelete.Enabled = False

End Sub

Sub saverecord()

If (TextBox1.Text = "" Or TextBox2.Text = "" Or ComboBox1.Text = "" Or TextBox3.Text = "" Or TextBox4.Text = "" Or TextBox5.Text = "") Then

MsgBox("Please Enter The Necessary Details")

Exit Sub

End If

If Conn.State = ConnectionState.Open Then Conn.Close()

Conn.Open()

Dim Cmd0 As New SqlCommand("select refno from AddPaintings where refno='" & UCase(TextBox2.Text) & "'", Conn)

Dim D1 As SqlDataReader = Cmd0.ExecuteReader()

If D1.HasRows Then

MsgBox("This reference number already exist in the database")

If Conn.State = ConnectionState.Open Then Conn.Close()

Exit Sub

End If

Dim command As New SqlCommand("insert into AddPaintings(artistname,refno,catname,paintingname,size,photo,price) values(@artistname,@refno,@catname,@paintingname,@size,@photo,@price)", Conn)

Dim ms As New MemoryStream

PictureBox1.Image.Save(ms, PictureBox1.Image.RawFormat)

command.Parameters.Add("@artistname", SqlDbType.NVarChar).Value = TextBox1.Text

command.Parameters.Add("@refno", SqlDbType.NVarChar).Value = TextBox2.Text

command.Parameters.Add("@catname", SqlDbType.NVarChar).Value = ComboBox1.Text

command.Parameters.Add("@paintingname", SqlDbType.NVarChar).Value = TextBox3.Text

command.Parameters.Add("@size", SqlDbType.NVarChar).Value = TextBox4.Text

command.Parameters.Add("@photo", SqlDbType.Image).Value = ms.ToArray

command.Parameters.Add("@price", SqlDbType.NVarChar).Value = TextBox5.Text

'command.ExecuteNonQuery()

If Conn.State = ConnectionState.Open Then Conn.Close()

Conn.Open()

If command.ExecuteNonQuery() = 1 Then

MsgBox("Painting added Successfully!!!")

Else

MsgBox("Error!!!")

End If

disRecords()

ButNew.Enabled = True

End Sub

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

TextBox1.Focus()

Me.WindowState = FormWindowState.Maximized

End Sub

Private Sub Button4\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button4.Click

Me.Close()

Artists.Show()

End Sub

Private Sub ItemForm\_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Load

Me.WindowState = FormWindowState.Maximized

If Conn.State = ConnectionState.Open Then Conn.Close()

Conn.Open()

Dim Cmd1 As New SqlCommand("select catname from category order by catname", Conn)

Dim D1 As SqlDataReader = Cmd1.ExecuteReader()

While D1.Read

ComboBox1.Items.Add(D1(0).ToString)

End While

End Sub

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

Try

With OpenFileDialog1

.Filter = ("Images |\*.png; \*.bmp; \*.jpg; \*.jpeg; \*.gif; \*.ico;")

.FilterIndex = 4

End With

OpenFileDialog1.FileName = ""

If OpenFileDialog1.ShowDialog() = DialogResult.OK Then

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

End If

Catch ex As Exception

MsgBox(ex.ToString

End Try

End Sub

Private Sub TextBox1\_LostFocus(ByVal sender As Object, ByVal e As System.EventArgs) Handles TextBox1.LostFocus

If Not Regex.Match(TextBox1.Text, "^[a-zA-Z ]\*$", RegexOptions.IgnoreCase).Success Then

MsgBox("please enter alpha text only!")

TextBox3.Focus()

TextBox1.Clear()

End If

End Sub

Private Sub TextBox3\_LostFocus(ByVal sender As Object, ByVal e As System.EventArgs) Handles TextBox3.LostFocus

If Not Regex.Match(TextBox3.Text, "^[a-zA-Z ]\*$", RegexOptions.IgnoreCase).Success Then

MsgBox("please enter alpha text only!")

TextBox2.Focus()

TextBox3.Clear()

End If

End Sub

Private Sub TextBox4\_KeyPress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles TextBox4.KeyPress

' e.Handled = Not (e.KeyChar = "." Or Char.IsDigit(e.KeyChar) Or e.KeyChar = vbBack)

End Sub

Private Sub TextBox5\_KeyPress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles TextBox5.KeyPress

e.Handled = Not (e.KeyChar = "." Or Char.IsDigit(e.KeyChar) Or e.KeyChar = vbBack)

End Sub

Private Sub TextBox1\_KeyPress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles TextBox1.KeyPress

e.Handled = Not (e.KeyChar = "." Or Char.IsLetter(e.KeyChar) Or e.KeyChar = vbBack)

End Sub

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

End Sub

Private Sub DG1\_CellContentClick(ByVal sender As System.Object, ByVal e As System.Windows.Forms.DataGridViewCellEventArgs) Handles DG1.CellContentClick

End Sub

Sub disRecords()

If Conn.State = ConnectionState.Open Then Conn.Close()

Conn.Open()

Dim DS1 As New DataSet

Dim adp As New SqlDataAdapter("Select \* From AddPaintings order by artistname", Conn)

adp.Fill(DS1)

DG1.DataSource = DS1.Tables(0)

If Conn.State = ConnectionState.Open Then Conn.Close()

End Sub

Private Sub Button5\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ButModify.Click

If vbNo = MsgBox("Are you sure you want modify this record", MsgBoxStyle.YesNo, "Delete") Then Exit Sub

If Conn.State = ConnectionState.Open Then Conn.Close()

Conn.Open()

Dim cmd1 As New SqlCommand("Delete from AddPaintings where artistname='" & pkVar & "'", Conn)

cmd1.ExecuteNonQuery()

If Conn.State = ConnectionState.Open Then Conn.Close()

SaveRecord()

ButNew.Enabled = True

ButSave.Enabled = False

ButModify.Enabled = False

ButDelete.Enabled = False

End Sub

Private Sub Button6\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ButDelete.Click

If vbNo = MsgBox("Are you sure you want delete this record", MsgBoxStyle.YesNo, "Delete") Then Exit Sub

If Conn.State = ConnectionState.Open Then Conn.Close()

Conn.Open()

Dim cmd1 As New SqlCommand("Delete \* from AddPaintings where artistname='" & pkVar & "'", Conn)

cmd1.ExecuteNonQuery()

If Conn.State = ConnectionState.Open Then Conn.Close()

disRecords()

ButNew.Enabled = True

ButSave.Enabled = False

ButModify.Enabled = False

ButDelete.Enabled = False

End Sub

Private Sub DG1\_CellMouseClick(ByVal sender As Object, ByVal e As System.Windows.Forms.DataGridViewCellMouseEventArgs) Handles DG1.CellMouseClick

pkVar = DG1.CurrentRow.Cells(0).Value

If Conn.State = ConnectionState.Open Then Conn.Close()

Conn.Open()

Dim Cmd0 As New SqlCommand("select \* from AddPaintings where artistname ='" & pkVar & "'", Conn)

Dim D1 As SqlDataReader = Cmd0.ExecuteReader()

If D1.HasRows Then

D1.Read()

TextBox1.Text = D1(0).ToString

ComboBox1.Text = D1(2).ToString

TextBox2.Text = D1(1).ToString

TextBox3.Text = D1(3).ToString

TextBox4.Text = D1(4).ToString

TextBox5.Text = D1(6).ToString

Dim img As Array

img = DG1.Item(5, e.RowIndex).Value

Dim ms As New MemoryStream(img)

PictureBox1.Image = Image.FromStream(ms)

ButNew.Enabled = True

ButSave.Enabled = False

ButModify.Enabled = True

ButDelete.Enabled = True

Else

TextBox1.Text = ""

TextBox2.Text = ""

ButNew.Enabled = True

ButSave.Enabled = False

ButModify.Enabled = False

ButDelete.Enabled = False

End If

If Conn.State = ConnectionState.Open Then Conn.Close()

End Sub

Private Sub butview\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles butview.Click

disRecords()

End Sub

Private Sub PictureBox2\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles PictureBox2.Click

Me.Dispose()

End Sub

Private Sub TextBox1\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox1.TextChanged

End Sub

Private Sub TextBox3\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox3.TextChanged

End Sub

Private Sub TextBox2\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox2.TextChanged

End Sub

End Class

**CUSTOMER SIGNUP**

Imports System.Data.SqlClient

Imports System.IO

Imports System.Text.RegularExpressions

Imports System.Text

Imports System.Configuration

Imports System.Net.Mail

Public Class custsignup

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

If (TextBox1.Text = "" Or TextBox2.Text = "" Or TextBox3.Text = "" Or TextBox4.Text = "" Or TextBox5.Text = "") Then

MsgBox("Please Enter The Necessary Details")

Exit Sub

End If

Dim command As New SqlCommand("insert into custsignup(loginname,password,address,mobileno,email) values(@loginname,@password,@address,@mobileno,@email)", Conn)

Dim ms As New MemoryStream

' PictureBox1.Image.Save(ms, PictureBox1.Image.RawFormat)

command.Parameters.Add("@loginname", SqlDbType.NVarChar).Value = TextBox1.Text

command.Parameters.Add("@password", SqlDbType.NVarChar).Value = TextBox2.Text

command.Parameters.Add("@address", SqlDbType.NVarChar).Value = TextBox3.Text

command.Parameters.Add("@mobileno", SqlDbType.NVarChar).Value = TextBox4.Text

command.Parameters.Add("@email", SqlDbType.NVarChar).Value = TextBox5.Text

If Conn.State = ConnectionState.Open Then Conn.Close()

Conn.Open()

If command.ExecuteNonQuery() = 1 Then

MsgBox("Saved Successfully!!!")

Sendmail()

Else

MsgBox("Error!!!")

End If

End Sub

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

Me.Hide()

ArtGallery.Show()

End Sub

Private Sub TextBox5\_Validating(ByVal sender As Object, ByVal e As System.ComponentModel.CancelEventArgs) Handles TextBox5.Validating

Dim tbox As TextBox = DirectCast(sender, TextBox)

If Regex.IsMatch(tbox.Text, "^([a-zA-Z0-9\_\-\.]+)@((\[[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.)|(([a-zA-Z0-9\-]+\.)+))([a-zA-Z]{2,4}|[0-9]{1,3})(\]?)$") = True Then

ErrorProvider1.SetError(Me.TextBox5, "")

e.Cancel = False

Else

ErrorProvider1.SetError(Me.TextBox5, "Not a valid email")

e.Cancel = True

End If

End Sub

Private Sub TextBox4\_Validating(ByVal sender As Object, ByVal e As System.ComponentModel.CancelEventArgs) Handles TextBox4.Validating

'Dim phoneNumber As New Regex("\d{3}\d{3}\d{4}")

Dim phoneNumber As New Regex("^([6-9]{1})([0-9]{9})")

TextBox4.MaxLength = 10

If phoneNumber.IsMatch(TextBox4.Text) Then

TextBox5.Focus()

Else

MsgBox("Not Valid Phone Number")

TextBox4.Text = ""

TextBox4.Focus()

End If

End Sub

Sub Sendmail()

Dim Mail As New MailMessage

Mail.Subject = "Your Account Has Been Approved!!!"

If TextBox5.Text = "" Then

MsgBox("Please Enter The E-Mail Address", MsgBoxStyle.Information + MsgBoxStyle.OkOnly, "Error!")

End If

Mail.To.Add(TextBox5.Text)

Mail.From = New MailAddress("pavithraparpattedar29@gmail.com")

Mail.Body = ("Your account name=" & TextBox1.Text & " and password=" & TextBox2.Text & " ")

Dim SMTP As New SmtpClient("smtp.gmail.com")

SMTP.UseDefaultCredentials = False

SMTP.EnableSsl = True

SMTP.Credentials = New System.Net.NetworkCredential("pavithraparpattedar29@gmail.com", "putgoovityxflqrf")

SMTP.Port = 587

SMTP.Send(Mail)

MsgBox("Mail sent Successfully")

End Sub

Private Sub PictureBox1\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles PictureBox1.Click

Me.Dispose()

End Sub

End Class

**CUSTOMER SIGN IN**

Imports System.Data.SqlClient

Public Class custsignin

Private Sub OK\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles OK.Click

If Conn.State = ConnectionState.Open Then Conn.Close()

Conn.Open()

Dim Cmd0 As New SqlCommand("select \* from custsignup where loginname='" & UCase(UsernameTextBox.Text) & "' and password='" & PasswordTextBox.Text & "'", Conn)

Dim D1 As SqlDataReader = Cmd0.ExecuteReader()

If D1.HasRows Then

Customer.Show()

Me.Hide()

If Conn.State = ConnectionState.Open Then Conn.Close()

Else

MsgBox("Username or Password is not correct please check")

End If

End Sub

Private Sub Cancel\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Cancel.Click

Me.Close()

ArtGallery.Show()

End Sub

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

UsernameTextBox.Text = ""

PasswordTextBox.Text = ""

End Sub

Private Sub CheckBox1\_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles CheckBox1.CheckedChanged

If CheckBox1.CheckState = CheckState.Checked Then

PasswordTextBox.UseSystemPasswordChar = False

Else

PasswordTextBox.UseSystemPasswordChar = True

End If

End Sub

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

PasswordTextBox.UseSystemPasswordChar = True

End Sub

Private Sub PictureBox2\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles PictureBox2.Click

Me.Dispose()

End Sub

End Class

**CUSTOMER**

Public Class Customer

Private Sub HomeToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles HomeToolStripMenuItem.Click

Me.Close()

custsignin.Close()

ArtGallery.Show()

End Sub

Private Sub ArtistsToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ArtistsToolStripMenuItem.Click

Panel1.Controls.Clear()

With ViewAllArtists

.TopLevel = False

Panel1.Controls.Add(ViewAllArtists)

.BringToFront()

.Show()

End With

End Sub

Private Sub ToolStripMenuItem1\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ToolStripMenuItem1.Click

Panel1.Controls.Clear()

With SearchPaintings

.TopLevel = False

Panel1.Controls.Add(SearchPaintings)

.BringToFront()

.Show()

End With

End Sub

Private Sub ExhibitionToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ExhibitionToolStripMenuItem.Click

Panel1.Controls.Clear()

With Exhibition

.TopLevel = False

Panel1.Controls.Add(Exhibition)

.BringToFront()

.Show()

End With

' Exhibition.Show()

End Sub

Private Sub ExitToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)

MsgBox("are you sure you want to exit")

Me.Close()

ArtGallery.Show()

End Sub

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

Me.WindowState = FormWindowState.Maximized

End Sub

Private Sub PictureBox1\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles PictureBox1.Click

Me.Dispose()

ArtGallery.Show()

End Sub

Private Sub Panel1\_Paint(ByVal sender As System.Object, ByVal e As System.Windows.Forms.PaintEventArgs) Handles Panel1.Paint

End Sub

End Class

**VIEW ALL ARTISTS**

Imports System.Data.SqlClient

Public Class ViewAllArtists

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

Me.WindowState = FormWindowState.Maximized

If Conn.State = ConnectionState.Open Then Conn.Close()

Conn.Open()

Dim DS1 As New DataSet

Dim adp As New SqlDataAdapter("Select artistname,addrs,dob,mob,email,qualif From ArtistReg order by artistname", Conn)

adp.Fill(DS1)

DG1.DataSource = DS1.Tables(0)

If Conn.State = ConnectionState.Open Then Conn.Close()

End Sub

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

Me.Close()

End Sub

Private Sub PictureBox1\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles PictureBox1.Click

Me.Dispose()

End Sub

End Class

**SEARCH PAINTINGS**

Imports System.Data.SqlClient

Public Class SearchPaintings

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

Me.WindowState = FormWindowState.Maximized

End Sub

Private Sub ItemForm\_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Load

Me.WindowState = FormWindowState.Maximized

If Conn.State = ConnectionState.Open Then Conn.Close()

Conn.Open()

Dim Cmd1 As New SqlCommand("select artistname from AddPaintings order by artistname", Conn)

Dim D1 As SqlDataReader = Cmd1.ExecuteReader()

While D1.Read

ComboBox1.Items.Add(D1(0).ToString)

End While

End Sub

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

If Conn.State = ConnectionState.Open Then Conn.Close()

Conn.Open()

Dim DS1 As New DataSet

Dim adp As New SqlDataAdapter("select artistname,refno,catname,paintingname,size,photo,price From AddPaintings where artistname='" & ComboBox1.Text & "' order by artistname", Conn)

adp.Fill(DS1)

DG1.AllowUserToAddRows = False

DG1.RowTemplate.Height = 50

Dim imgc As New DataGridViewImageColumn

DG1.DataSource = DS1.Tables(0)

imgc = DG1.Columns(5)

imgc.ImageLayout = DataGridViewImageCellLayout.Stretch

If Conn.State = ConnectionState.Open Then Conn.Close()

End Sub

Private Sub DG1\_CellMouseClick(ByVal sender As Object, ByVal e As System.Windows.Forms.DataGridViewCellMouseEventArgs) Handles DG1.CellMouseClick

Dim q1var, q2var

If Conn.State = ConnectionState.Open Then Conn.Close()

Conn.Open()

Dim cmd0 As New SqlCommand("Select refno from SelectedList where refno='" & DG1.CurrentRow.Cells(1).Value & "'", Conn)

Dim D1 As SqlDataReader = cmd0.ExecuteReader()

If D1.HasRows Then

MsgBox("This Item Is Already Present In The Cart")

If Conn.State = ConnectionState.Open Then Conn.Close()

Exit Sub

End If

If Conn.State = ConnectionState.Open Then Conn.Close()

Conn.Open()

If Conn.State = ConnectionState.Open Then Conn.Close()

Conn.Open()

q1var = "insert into SelectedList("

q2var = " values("

q1var = q1var & "artistname" & ","

q2var = q2var & "'" & DG1.CurrentRow.Cells(0).Value & "',"

q1var = q1var & "refno" & ","

q2var = q2var & "'" & DG1.CurrentRow.Cells(1).Value & "',"

q1var = q1var & "paintingname" & ","

q2var = q2var & "'" & DG1.CurrentRow.Cells(3).Value & "',"

q1var = q1var & "size" & ","

q2var = q2var & "'" & DG1.CurrentRow.Cells(4).Value & "',"

q1var = q1var & "price" & ")"

q2var = q2var & "'" & DG1.CurrentRow.Cells(6).Value & "')"

'MsgBox(q1var & q2var)

Dim cmd1 As New SqlCommand(q1var & q2var, Conn)

cmd1.ExecuteNonQuery()

If Conn.State = ConnectionState.Open Then Conn.Close()

'disRecords()

MsgBox("Added To selected list!")

End Sub

Private Sub DG1\_CellContentClick(ByVal sender As System.Object, ByVal e As System.Windows.Forms.DataGridViewCellEventArgs) Handles DG1.CellContentClick

End Sub

Private Sub Button3\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button3.Click

Me.Close()

searchbycategory.Show()

End Sub

Private Sub PictureBox1\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles PictureBox1.Click

Me.Dispose()

End Sub

Private Sub Label1\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Label1.Click

End Sub

End Class

**SEARCH CATEGORY**

Imports System.Data.SqlClient

Public Class searchbycategory

Private Sub Search\_By\_Category\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

Me.WindowState = FormWindowState.Maximized

End Sub

Private Sub ItemForm\_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Load

Me.WindowState = FormWindowState.Maximized

If Conn.State = ConnectionState.Open Then Conn.Close()

Conn.Open()

Dim Cmd1 As New SqlCommand("select catname from AddPaintings order by catname", Conn)

Dim D1 As SqlDataReader = Cmd1.ExecuteReader()

While D1.Read

ComboBox1.Items.Add(D1(0).ToString)

End While

End Sub

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

If Conn.State = ConnectionState.Open Then Conn.Close()

Conn.Open()

Dim DS1 As New DataSet

Dim adp As New SqlDataAdapter("select artistname,refno,catname,paintingname,size,photo,price From AddPaintings where catname='" & ComboBox1.Text & "' order by catname", Conn)

adp.Fill(DS1)

DG1.AllowUserToAddRows = False

DG1.RowTemplate.Height = 50

Dim imgc As New DataGridViewImageColumn

DG1.DataSource = DS1.Tables(0)

imgc = DG1.Columns(5)

imgc.ImageLayout = DataGridViewImageCellLayout.Stretch

If Conn.State = ConnectionState.Open Then Conn.Close()

End Sub

Private Sub DG1\_CellMouseClick(ByVal sender As Object, ByVal e As System.Windows.Forms.DataGridViewCellMouseEventArgs) Handles DG1.CellMouseClick

Dim q1var, q2var

If Conn.State = ConnectionState.Open Then Conn.Close()

Conn.Open()

Dim cmd0 As New SqlCommand("Select refno from SelectedList where refno='" & DG1.CurrentRow.Cells(1).Value & "'", Conn)

Dim D1 As SqlDataReader = cmd0.ExecuteReader()

If D1.HasRows Then

MsgBox("This Item Is Already Present In The Cart")

If Conn.State = ConnectionState.Open Then Conn.Close()

Exit Sub

End If

If Conn.State = ConnectionState.Open Then Conn.Close()

Conn.Open()

If Conn.State = ConnectionState.Open Then Conn.Close()

Conn.Open()

q1var = "insert into SelectedList("

q2var = " values("

q1var = q1var & "artistname" & ","

q2var = q2var & "'" & DG1.CurrentRow.Cells(0).Value & "',"

q1var = q1var & "refno" & ","

q2var = q2var & "'" & DG1.CurrentRow.Cells(1).Value & "',"

q1var = q1var & "paintingname" & ","

q2var = q2var & "'" & DG1.CurrentRow.Cells(3).Value & "',"

q1var = q1var & "size" & ","

q2var = q2var & "'" & DG1.CurrentRow.Cells(4).Value & "',"

q1var = q1var & "price" & ")"

q2var = q2var & "'" & DG1.CurrentRow.Cells(6).Value & "')"

'MsgBox(q1var & q2var)

Dim cmd1 As New SqlCommand(q1var & q2var, Conn)

cmd1.ExecuteNonQuery()

If Conn.State = ConnectionState.Open Then Conn.Close()

'disRecords()

MsgBox("Added To selected list!")

End Sub

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

Me.Close()

End Sub

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

End Sub

Private Sub PictureBox1\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles PictureBox1.Click

Me.Dispose()

End Sub

End Class

**ADMIN SIGNUP**

Imports System.Data.SqlClient

Public Class Form1

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

If Conn.State = ConnectionState.Open Then Conn.Close()

Conn.Open()

q1Var = "insert into SignUpTab("

q2Var = " values("

q1Var = q1Var & "name" & ","

q2Var = q2Var & "'" & TextBox1.Text & "',"

q1Var = q1Var & "userName" & ","

q2Var = q2Var & "'" & TextBox2.Text & "',"

q1Var = q1Var & "password" & ","

q2Var = q2Var & "'" & TextBox3.Text & "',"

q1Var = q1Var & "dob" & ","

q2Var = q2Var & "'" & DateTimePicker1.Value & "',"

q1Var = q1Var & "gender" & ","

q2Var = q2Var & "'" & ComboBox2.Text & "',"

q1Var = q1Var & "address" & ","

q2Var = q2Var & "'" & TextBox6.Text & "',"

q1Var = q1Var & "phoneno" & ","

q2Var = q2Var & "'" & TextBox7.Text & "',"

q1Var = q1Var & "state" & ","

q2Var = q2Var & "'" & TextBox4.Text & "',"

q1Var = q1Var & "country" & ","

q2Var = q2Var & "'" & ComboBox1.Text & "',"

q1Var = q1Var & "question" & ","

q2Var = q2Var & "'" & ComboBox3.Text & "',"

q1Var = q1Var & "answer" & ")"

q2Var = q2Var & "'" & TextBox5.Text & "')"

'MsgBox(q1var, q2var)

Dim cmd As New SqlCommand(q1Var & q2Var, Conn)

cmd.ExecuteNonQuery()

MsgBox("Record Saved successfully")

If Conn.State = ConnectionState.Open Then

End If

Conn.Close()

End Sub

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

Me.Hide()

LoginForm1.Show()

End Sub

Private Sub Label4\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Label4.Click

End Sub

Private Sub PictureBox1\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles PictureBox1.Click

Me.Dispose()

LoginForm1.Show()

End Sub

Private Sub ComboBox3\_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ComboBox3.SelectedIndexChanged

End Sub

Private Sub Form1\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

End Sub

End Class

**CHECK PASSWORD**

Imports System.Data.SqlClient

Public Class Form2

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

If Conn.State = ConnectionState.Open Then Conn.Close()

Conn.Open()

Dim Cmd0 As New SqlCommand("select \* from SignUpTab where Username='" & (Textusername.Text) & "'", Conn)

Dim D1 As SqlDataReader = Cmd0.ExecuteReader()

If D1.HasRows Then

Label2.Visible = True

Label3.Visible = True

ComboBox1.Visible = True

Textanswer.Visible = True

Button2.Visible = True

Button3.Visible = True

Else

MsgBox("INCORRECT USERNAME")

End If

End Sub

Private Sub Button3\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button3.Click

If Conn.State = ConnectionState.Open Then Conn.Close()

Conn.Open()

Dim Cmd0 As New SqlCommand("select \* from SignUpTab where Question='" & (ComboBox1.Text) & "' and Answer='" & Textanswer.Text & "'", Conn)

Dim D1 As SqlDataReader = Cmd0.ExecuteReader()

If D1.HasRows Then

D1.Read()

Label4.Text = D1(2).ToString

Label4.Visible = True

Else

MsgBox("INCORRECT ANSWER FOR USERNAME'S QUESTION")

End If

End Sub

Private Sub Form2\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

Label2.Visible = False

Label3.Visible = False

Label4.Visible = False

ComboBox1.Visible = False

Textanswer.Visible = False

Button2.Visible = False

Button3.Visible = False

End Sub

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

Me.Hide()

LoginForm1.Show()

End Sub

Private Sub PictureBox1\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles PictureBox1.Click

Me.Dispose()

LoginForm1.Show()

End Sub

End Class

**ADMIN LOGIN**

Imports System.Data.SqlClient

Public Class AdminLogin

Private Sub OK\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles OK.Click

If Conn.State = ConnectionState.Open Then Conn.Close()

Conn.Open()

Dim Cmd0 As New SqlCommand("select \* from AdminLog where uname='" & UCase(UsernameTextBox.Text) & "' and pword='" & PasswordTextBox.Text & "'", Conn)

Dim D1 As SqlDataReader = Cmd0.ExecuteReader()

If D1.HasRows Then

adminMain.Show()

Me.Hide()

If Conn.State = ConnectionState.Open Then Conn.Close()

Else

MsgBox("Username or Password is not correct please check")

End If

End Sub

Private Sub Cancel\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Cancel.Click

Me.Close()

End Sub

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

' Me.WindowState = FormWindowState.Maximized

End Sub

Private Sub PictureBox1\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles PictureBox1.Click

Me.Dispose()

End Sub

End Class

**ADMIN MAIN**

Public Class adminMain

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

Panel1.Controls.Clear()

With ArtistDatabase

.TopLevel = False

Panel1.Controls.Add(ArtistDatabase)

.BringToFront()

.Show()

End With

End Sub

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

Panel1.Controls.Clear()

With ApproveArtists

.TopLevel = False

Panel1.Controls.Add(ApproveArtists)

.BringToFront()

.Show()

End With

End Sub

Private Sub Button3\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button3.Click

Panel1.Controls.Clear()

With AddArtCategory

.TopLevel = False

Panel1.Controls.Add(AddArtCategory)

.BringToFront()

.Show()

End With

End Sub

Private Sub Button4\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button4.Click

Me.Hide()

LoginForm1.Show()

End Sub

Private Sub Panel1\_Paint(ByVal sender As System.Object, ByVal e As System.Windows.Forms.PaintEventArgs) Handles Panel1.Paint

End Sub

End Class

**APPROVE ARTISTS**

Imports System.Data.SqlClient

Imports System.Net.Mail

Imports System.Text.RegularExpressions

Public Class ApproveArtists

' Private Sub ButShow\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ButShow.Click

' If (TextBox1.Text = "" Or TextBox2.Text = "" Or TextBox3.Text = "" Or TextBox5.Text = "" Or TextBox6.Text = "") Then

' MsgBox("Please Enter The Necessary Details")

' Exit Sub

' End If

' If Conn.State = ConnectionState.Open Then Conn.Close()

'Conn.Open()

'Dim DS1 As New DataSet

'Dim adp As New SqlDataAdapter("Select artistname,pword,addrs,mob,email,qualif From ArtistReg order by artistname", Conn)

'adp.Fill(DS1)

'DG1.DataSource = DS1.Tables(0)

'If Conn.State = ConnectionState.Open Then Conn.Close()

'End Sub

Private Sub DG1\_CellMouseClick(ByVal sender As Object, ByVal e As System.Windows.Forms.DataGridViewCellMouseEventArgs) Handles DG1.CellMouseClick

On Error Resume Next

TextBox1.Text = DG1.Rows(e.RowIndex).Cells(0).Value

TextBox2.Text = DG1.Rows(e.RowIndex).Cells(1).Value

TextBox3.Text = DG1.Rows(e.RowIndex).Cells(2).Value

TextBox5.Text = DG1.Rows(e.RowIndex).Cells(3).Value

TextBox6.Text = DG1.Rows(e.RowIndex).Cells(4).Value

End Sub

Private Sub ButMail\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ButMail.Click

Dim Mail As New MailMessage

Mail.Subject = "Your Account Has Been Approved!!!"

If TextBox6.Text = "" Then

MsgBox("Please Enter The E-Mail Address", MsgBoxStyle.Information + MsgBoxStyle.OkOnly, "Error!")

End If

Mail.To.Add(TextBox6.Text)

Mail.From = New MailAddress("pavithraparpattedar29@gmail.com")

Mail.Body = ("Your account name=" & TextBox1.Text & " and password=" & TextBox2.Text & " ")

Dim SMTP As New SmtpClient("smtp.gmail.com")

SMTP.UseDefaultCredentials = False

SMTP.EnableSsl = True

SMTP.Credentials = New System.Net.NetworkCredential("pavithraparpattedar29@gmail.com", "putgoovityxflqrf")

SMTP.Port = 587

SMTP.Send(Mail)

MsgBox("Approved Successfully")

End Sub

Private Sub TextBox5\_LostFocus(ByVal sender As Object, ByVal e As System.EventArgs) Handles TextBox5.LostFocus

'Dim phoneNumber As New Regex("\d{3}\d{3}\d{4}")

Dim phoneNumber As New Regex("^([6-9]{1})([0-9]{9})")

TextBox5.MaxLength = 10

If phoneNumber.IsMatch(TextBox5.Text) Then

TextBox6.Focus()

Else

MsgBox("Not Valid Phone Number")

TextBox5.Text = ""

TextBox5.Focus()

End If

End Sub

Private Sub TextBox1\_LostFocus(ByVal sender As Object, ByVal e As System.EventArgs) Handles TextBox1.LostFocus

If Not Regex.Match(TextBox1.Text, "^[a-z. ]\*$", RegexOptions.IgnoreCase).Success Then

MsgBox("please enter alpha text only!")

TextBox2.Focus()

TextBox1.Clear()

End If

End Sub

Private Sub ButClose\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ButClose.Click

Me.Close()

adminMain.Show()

End Sub

Private Sub TextBox6\_Validating(ByVal sender As Object, ByVal e As System.ComponentModel.CancelEventArgs) Handles TextBox6.Validating

Dim tbox As TextBox = DirectCast(sender, TextBox)

If Regex.IsMatch(tbox.Text, "^([a-zA-Z0-9\_\-\.]+)@((\[[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.)|(([a-zA-Z0-9\-]+\.)+))([a-zA-Z]{2,4}|[0-9]{1,3})(\]?)$") = True Then

ErrorProvider1.SetError(Me.TextBox6, "")

e.Cancel = False

Else

ErrorProvider1.SetError(Me.TextBox6, "Not a valid email")

e.Cancel = True

End If

End Sub

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

TextBox1.Focus()

End Sub

Private Sub ButShow\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ButShow.Click

If Conn.State = ConnectionState.Open Then Conn.Close()

Conn.Open()

Dim DS1 As New DataSet

Dim adp As New SqlDataAdapter("Select artistname,pword,addrs,mob,email From ArtistReg order by artistname", Conn)

adp.Fill(DS1)

DG1.DataSource = DS1.Tables(0)

If Conn.State = ConnectionState.Open Then Conn.Close()

End Sub

Private Sub TextBox6\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox6.TextChanged

End Sub

Private Sub TextBox5\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox5.TextChanged

End Sub

Private Sub PictureBox1\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles PictureBox1.Click

Me.Dispose()

End Sub

End Class

**EXHIBITION**

Imports System.IO

Imports System.Data.SqlClient

Public Class Exhibition

Dim pkVar As String

Sub disRecords()

If Conn.State = ConnectionState.Open Then Conn.Close()

Conn.Open()

Dim DS1 As New DataSet

Dim adp As New SqlDataAdapter("Select \* From AddPaintings order by artistname", Conn)

adp.Fill(DS1)

DG1.DataSource = DS1.Tables(0)

If Conn.State = ConnectionState.Open Then Conn.Close()

End Sub

Private Sub Butview\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Butview.Click

disRecords()

End Sub

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

Me.Close()

End Sub

Private Sub DG1\_CellMouseClick(ByVal sender As Object, ByVal e As System.Windows.Forms.DataGridViewCellMouseEventArgs) Handles DG1.CellMouseClick

pkVar = DG1.CurrentRow.Cells(0).Value

If Conn.State = ConnectionState.Open Then Conn.Close()

Conn.Open()

Dim Cmd0 As New SqlCommand("select paintingname,refno,price,photo from AddPaintings where artistname ='" & pkVar & "'", Conn)

Dim D1 As SqlDataReader = Cmd0.ExecuteReader()

If D1.HasRows Then

D1.Read()

TextBox3.Text = D1(0).ToString

TextBox1.Text = D1(1).ToString

TextBox2.Text = D1(2).ToString

Dim img As Array

img = DG1.Item(5, e.RowIndex).Value

Dim ms As New MemoryStream(img)

PictureBox1.Image = Image.FromStream(ms)

End If

If Conn.State = ConnectionState.Open Then Conn.Close()

disRecords()

End Sub

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

Me.Hide()

Billing.Show()

End Sub

Private Sub PictureBox2\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles PictureBox2.Click

Me.Dispose()

End Sub

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

End Sub

End Class

**BILLING**

Imports System.Data.SqlClient

Imports System.IO

Public Class Billing

Dim GrandTotal

Dim pkVar As String

Dim BillNoVar As Long

Dim rowNo As Long

Dim totAmt, taxVar As Decimal

Dim DS1 As New DataSet

Dim Billing As DataTable

Private Sub Billing\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)

' TextBox3.Text = Exhibition.TextBox3.Text

'TextBox2.Text = Exhibition.TextBox1.Text

'Me.WindowState = FormWindowState.Maximized

'If Conn.State = ConnectionState.Open Then Conn.Close()

'Conn.Open()

'Dim D2 As SqlDataReader = Cmd0.ExecuteReader()

'While D2.Read

' TextBox3.Text.Add(D2(0).ToString)

'End While

End Sub

Sub NewRec()

ClearTxtControls(Me, 1)

If Conn.State = ConnectionState.Open Then Conn.Close()

Conn.Open()

Dim Cmd0 As New SqlCommand("select max(billno) from Billing", Conn)

Dim D1 As SqlDataReader = Cmd0.ExecuteReader()

If D1.HasRows Then

D1.Read()

TextBox1.Text = IIf(IsDBNull(D1(0)), 11100000, D1(0)) + 1

Else

TextBox1.Text = "1001"

End If

End Sub

Private Sub butNew\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)

End Sub

Private Sub ButClose\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)

Me.Close()

End Sub

Private Sub butSave\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)

SaveRecord()

butNew.Enabled = True

butSave.Enabled = False

butPrint.Enabled = True

End Sub

Sub SaveRecord()

Dim cmd3 As New SqlCommand

If TextBox1.Text = "" Then

MsgBox("Please enter the necessary details")

Exit Sub

End If

'TextBox1.Text = BillNoVar

If Conn.State = ConnectionState.Open Then Conn.Close()

Conn.Open()

q1Var = "insert into Billing("

q2Var = " values("

q1Var = q1Var & "billno" & ","

q2Var = q2Var & "'" & UCase(TextBox1.Text) & "',"

q1Var = q1Var & "billdate" & ","

q2Var = q2Var & "'" & UCase(bDate.Text) & "',"

q1Var = q1Var & "paintingname" & ","

q2Var = q2Var & "'" & UCase(TextBox3.Text) & "',"

q1Var = q1Var & "refno" & ","

q2Var = q2Var & "'" & UCase(TextBox2.Text) & "',"

q1Var = q1Var & "rate" & ","

q2Var = q2Var & "'" & UCase(TxtRate.Text) & "',"

q1Var = q1Var & "qty" & ","

q2Var = q2Var & "'" & UCase(TxtQty.Text) & "',"

q1Var = q1Var & "total" & ")"

q2Var = q2Var & "'" & TxtTot.Text & "')"

MsgBox(q1Var & q2Var)

Dim cmd1 As New SqlCommand(q1Var & q2Var, Conn)

cmd1.ExecuteNonQuery()

If Conn.State = ConnectionState.Open Then Conn.Close()

butPrint.Enabled = True

End Sub

Sub SaveRecord1()

Dim cmd3 As New SqlCommand

If TextBox1.Text = "" Then

MsgBox("Please enter the necessary details")

Exit Sub

End If

'TextBox1.Text = BillNoVar

If Conn.State = ConnectionState.Open Then Conn.Close()

Conn.Open()

q1Var = "insert into t1Tab("

q2Var = " values("

q1Var = q1Var & "billno" & ","

q2Var = q2Var & "'" & UCase(TextBox1.Text) & "',"

q1Var = q1Var & "billdate" & ","

q2Var = q2Var & "'" & UCase(bDate.Text) & "',"

q1Var = q1Var & "paintingname" & ","

q2Var = q2Var & "'" & UCase(TextBox3.Text) & "',"

q1Var = q1Var & "refno" & ","

q2Var = q2Var & "'" & UCase(TextBox2.Text) & "',"

q1Var = q1Var & "rate" & ","

q2Var = q2Var & "'" & UCase(TxtRate.Text) & "',"

q1Var = q1Var & "qty" & ","

q2Var = q2Var & "'" & UCase(TxtQty.Text) & "',"

q1Var = q1Var & "total" & ")"

q2Var = q2Var & "'" & TxtTot.Text & "')"

MsgBox(q1Var & q2Var)

Dim cmd1 As New SqlCommand(q1Var & q2Var, Conn)

cmd1.ExecuteNonQuery()

If Conn.State = ConnectionState.Open Then Conn.Close()

butPrint.Enabled = True

End Sub

Private Sub TxtTot\_TextChanged\_1(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TxtTot.TextChanged

TxtTot.Text = (Val(TxtQty.Text) \* Val(TxtRate.Text))

End Sub

'print code

Private Sub butPrint\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)

PP1.ShowDialog()

End Sub

Private Sub PP1\_Disposed(ByVal sender As Object, ByVal e As System.EventArgs)

PP1.Close()

PrintDocument1.Dispose()

End Sub

Private Sub PrintDocument1\_PrintPage(ByVal sender As System.Object, ByVal e As System.Drawing.Printing.PrintPageEventArgs)

End Sub

'not in use

' Private Sub ComboBox1\_TextChanged(ByVal sender As Object, ByVal e As System.EventArgs)

' If Conn.State = ConnectionState.Open Then Conn.Close()

' Conn.Open()

'Dim Cmd0 As New SqlCommand("select paintingname,refno,price from SelectedList where paintingname='" & TextBox3.Text & "'", Conn)

'Dim D1 As SqlDataReader = Cmd0.ExecuteReader()

'If D1.HasRows Then

' D1.Read()

' TextBox2.Text = D1(1).ToString

' TxtRate.Text = D1(4).ToString

' End If

' If Conn.State = ConnectionState.Open Then Conn.Close()

'End Sub

Sub disRecords1()

If Conn.State = ConnectionState.Open Then Conn.Close()

Conn.Open()

Dim DS1 As New DataSet

Dim adp As New SqlDataAdapter("Select \* From t1Tab order by billno", Conn)

adp.Fill(DS1)

DG1.DataSource = DS1.Tables(0)

If Conn.State = ConnectionState.Open Then Conn.Close()

End Sub

Sub disRecords()

If Conn.State = ConnectionState.Open Then Conn.Close()

Conn.Open()

Dim DS1 As New DataSet

Dim adp As New SqlDataAdapter("Select \* From AddPaintings order by artistname", Conn)

adp.Fill(DS1)

DG2.DataSource = DS1.Tables(0)

If Conn.State = ConnectionState.Open Then Conn.Close()

End Sub

Private Sub Butview\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles butview.Click

disRecords()

End Sub

Private Sub DG2\_CellMouseClick(ByVal sender As Object, ByVal e As System.Windows.Forms.DataGridViewCellMouseEventArgs) Handles DG2.CellMouseClick

pkVar = DG2.CurrentRow.Cells(1).Value

If Conn.State = ConnectionState.Open Then Conn.Close()

Conn.Open()

Dim Cmd0 As New SqlCommand("select paintingname,refno,price,photo from AddPaintings where refno ='" & pkVar & "'", Conn)

Dim D1 As SqlDataReader = Cmd0.ExecuteReader()

If D1.HasRows Then

D1.Read()

TextBox3.Text = D1(0).ToString

TextBox2.Text = D1(1).ToString

TxtRate.Text = D1(2).ToString

' Dim img As Array

'img = DG2.Item(5, e.RowIndex).Value

'Dim ms As New MemoryStream(img)

'PictureBox1.Image = Image.FromStream(ms)

End If

If Conn.State = ConnectionState.Open Then Conn.Close()

'disRecords()

End Sub

Private Sub Billing\_Load\_1(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

End Sub

Private Sub butAdd\_Click\_1(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles butAdd.Click

SaveRecord1()

'SaveRecord()

disRecords1()

butNew.Enabled = True

End Sub

Private Sub TxtQty\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TxtQty.TextChanged

TxtTot.Text = (Val(TxtQty.Text) \* Val(TxtRate.Text))

End Sub

Private Sub TxtRate\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TxtRate.TextChanged

TxtTot.Text = (Val(TxtQty.Text) \* Val(TxtRate.Text))

End Sub

Private Sub butNew\_Click\_1(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles butNew.Click

DG1Init()

'ClearTxtControls(Me, 1)

' TextBox1.Focus()

NewRec()

butNew.Enabled = False

butSave.Enabled = True

butPrint.Enabled = False

End Sub

Sub DG1Init()

rowNo = 0

totAmt = 0

DS1 = New DataSet

Billing = New DataTable

Billing = DS1.Tables.Add("t1Tab")

Billing.Columns.Add("billno", GetType(Long))

Billing.Columns.Add("billdate", GetType(String))

Billing.Columns.Add("paintingname", GetType(String))

Billing.Columns.Add("refno", GetType(String))

Billing.Columns.Add("rate", GetType(Long))

Billing.Columns.Add("qty", GetType(Long))

Billing.Columns.Add("total", GetType(Long))

DG1.DataSource = DS1.Tables(0)

End Sub

Private Sub DG2\_CellContentClick(ByVal sender As System.Object, ByVal e As System.Windows.Forms.DataGridViewCellEventArgs) Handles DG2.CellContentClick

End Sub

Private Sub DG1\_CellContentClick(ByVal sender As System.Object, ByVal e As System.Windows.Forms.DataGridViewCellEventArgs) Handles DG1.CellContentClick

End Sub

Private Sub butSave\_Click\_1(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles butSave.Click

For Each r As DataGridViewRow In DG1.Rows

GrandTotal = GrandTotal + r.Cells(6).Value

Next

TxtT3.Text = GrandTotal

End Sub

Private Sub butPrint\_Click\_1(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles butPrint.Click

PP1.ShowDialog()

' If vbNo = MsgBox("Are you sure you want delete this record", MsgBoxStyle.YesNo, "Delete") Then Exit Sub

If Conn.State = ConnectionState.Open Then Conn.Close()

Conn.Open()

Dim cmd1 As New SqlCommand("Delete from t1Tab", Conn)

cmd1.ExecuteNonQuery()

If Conn.State = ConnectionState.Open Then Conn.Close()

disRecords1()

End Sub

Private Sub PrintDocument1\_PrintPage\_1(ByVal sender As System.Object, ByVal e As System.Drawing.Printing.PrintPageEventArgs) Handles PrintDocument1.PrintPage

Dim XPos, YPos As Long

YPos += 50

XPos = 10

Dim MyFont As New Font("Arial", 20)

e.Graphics.DrawString("Art Gallery ", MyFont, Brushes.Black, XPos, YPos)

YPos += 50

MyFont = New Font("Arial", 14)

XPos = 10

e.Graphics.DrawString(" Purchase Bill ", MyFont, Brushes.Black, XPos, YPos)

XPos = 200

e.Graphics.DrawString("Date : ", MyFont, Brushes.Black, XPos, YPos)

XPos = 300

e.Graphics.DrawString(bDate.Value, MyFont, Brushes.Black, XPos, YPos)

YPos += 50

XPos = 200

e.Graphics.DrawString("BillNo : ", MyFont, Brushes.Black, XPos, YPos)

XPos = 300

e.Graphics.DrawString(TextBox1.Text, MyFont, Brushes.Black, XPos, YPos)

YPos += 50

XPos = 10

MyFont = New Font("Arial", 12)

e.Graphics.DrawString("Painting name", MyFont, Brushes.Black, XPos, YPos)

YPos += 25

XPos = 10

e.Graphics.DrawString(TextBox3.Text, MyFont, Brushes.Black, XPos, YPos)

YPos += 20

XPos = 10

e.Graphics.DrawString("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_", MyFont, Brushes.Black, XPos, YPos)

YPos += 30

XPos = 10

e.Graphics.DrawString("Reference Number", MyFont, Brushes.Black, XPos, YPos)

XPos = XPos + 200

e.Graphics.DrawString("Qty", MyFont, Brushes.Black, XPos, YPos)

XPos = XPos + 150

e.Graphics.DrawString("Rate", MyFont, Brushes.Black, XPos, YPos)

XPos = XPos + 150

e.Graphics.DrawString("Total Amt", MyFont, Brushes.Black, XPos, YPos)

XPos = XPos + 150

e.Graphics.DrawString("Grand Total Amt", MyFont, Brushes.Black, XPos, YPos)

XPos = XPos + 150

YPos += 10

XPos = 10

e.Graphics.DrawString("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_", MyFont, Brushes.Black, XPos, YPos)

YPos += 25

For Each r As DataGridViewRow In DG1.Rows

XPos = 10

e.Graphics.DrawString(r.Cells(0).Value, MyFont, Brushes.Black, XPos, YPos)

XPos = XPos + 200

e.Graphics.DrawString(r.Cells(4).Value, MyFont, Brushes.Black, XPos, YPos)

XPos = XPos + 150

e.Graphics.DrawString(r.Cells(5).Value, MyFont, Brushes.Black, XPos, YPos)

XPos = XPos + 150

e.Graphics.DrawString(r.Cells(6).Value, MyFont, Brushes.Black, XPos, YPos)

XPos = XPos + 15

YPos += 25

Next

XPos = 10

XPos = XPos + 250

e.Graphics.DrawString(TxtT3.Text, MyFont, Brushes.Black, XPos, YPos)

XPos = 10

e.Graphics.DrawString("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_", MyFont, Brushes.Black, XPos, YPos)

End Sub

Private Sub PictureBox1\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles PictureBox1.Click

Me.Dispose()

End Sub

Private Sub ButClose\_Click\_1(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ButClose.Click

Me.Close()

End Sub

End Class

**CHAPTER-07**

**TESTING**

**Testing**

Testing is the process of detecting errors. Testing performs a very critical role for quality assurance and for ensuring the reliability of software. The results of testing are used later on during maintenance also.

### 7.1. Objectives:

* To identify Testing Requirements (Scope):

- to identify the software to be tested

- to identify the testing objectives

- to identify the test phases (testing coverage) within the testing life cycle that is required

* To identify Testing Approach:

- to identify the methods and testing tools required

- to identify any client assumptions/dependencies/limitations

* To identify Testing Tasks and Deliverables:

- to identify the activities to perform within each testing phase.

- to identify the external (client) deliverable document.

- to identify the table of contents for each deliverable

- to identify the internal deliverable documents

- to identify document deliverable reviewer

* To compile Testing Estimates

- to identify the budgetary estimate for each identified phase of software testing.

- to identify the start and end date for each phase of software testing

- to identify all testing phase overlaps in the schedule

- to identify delivery dates for all document deliverables

* To determine Test Environments:

- to identify the software/hardware requirements for each test phase

- to identify the number of test environments

* To identify Test Team Roles and Responsibilities:

- to identify the overall testing management responsibility and for each test phase

- to identify client roles and responsibilities

**7.2 Levels of Testing**

In order to uncover the errors present in different phases we have the concept of levels of testing. The basic levels of testing are as shown below

**7.2.1 Code Testing:**

This strategy examines the logic of the program. To follow this method we developed some test data that resulted in executing every instruction in the program and module i.e. every path is tested. Systems are not designed as entire nor are they tested as single systems. To ensure that the coding is perfect two types of testing is performed or for that matter is performed or that matter is performed or for that matter is performed on all systems.

Here the Aptitude Quest source code is developed in java and servlets which is tested by JDK.

**7.2.2 Unit Testing:**

The goal of unit testing is to assure that all functions and features of a single compilable unit of code perform as specified in the Design Specification. A unit test covers the testing of a software unit, or a group of closely related units, as a single entity. Unit testing is performed in isolation, using test drivers to simulate higher level units, and/or stubs to simulate lower level units.

Unit Testing Procedures consist of:

* + Creating a Unit Test Plan
  + Creating test data
  + Conducting tests according to the Unit Test Plan
  + Reporting and reviewing the results of the test

These procedures are performed by the team member responsible for programming and testing of the unit.

A Unit Test Plan is a set of test cases arranged in the sequence of chronological execution. The Unit Test Plan is created before the programming of the unit is started, and the test cases should cover the functional, input, output, and function interaction of the unit.

The modules of **Aptitude Quest** i.e. User and Administration modules are tested whether it can interact with database and server.

**6.2.3 Integration Testing:**

The goal of integration testing is to ensure that all interacting subsystems in a system interface correctly with one another to produce the desired results. Furthermore, in trying to attain this goal, integration tests will ensure that the introduction of one or more subsystems into the system does not have an adverse affect on existing functionality.

An integration test covers the testing of interface points between subsystems. Integration testing is performed once unit testing has been completed for all units contained in the subsystems being tested.

Integration Testing Procedures consist of:

* + Creating and integration test plan
  + Creating test data
  + Conducting tests according to the integration test plan
  + Reporting and reviewing the results of the test

During this phase, the interaction between subsystems is tested. This includes interfaces through Inter Process Communications (IPC) and files. This phase is performed by an independent test team. This team prepares and executes integration tests, generates problem reports and is responsible for passing the integrated system on to the System Test Team for system testing. The Integration Test team then enters a support mode in which it will test problem reports generated by the System Test team before forwarding code fixes to the System Testing environment.

This phase is sometimes combined with the system test phase as per the client's request.

Each and every modules of **Aptitude Quest** are integrated and checked whether they interact with each other or not.

**7.2.4 System Testing:**

The goal of System Testing is to ensure that the system performs as per the functional requirements specified by client.

A system test covers the testing of functions within the system. System testing is performed once integration testing has been completed. System Testing procedures consist of:

* + Creating Test Plans
  + Creating test data
  + Conducting tests according to the System Test Plan
  + Reporting and reviewing the results of the test

Features to be tested during System Testing are:

* + Functional Requirements
  + Depending on the project, any regression tests deemed necessary

The developed **Aptitude Quest** is entirely tested by means of system testing.

**7.2.5 Acceptance Testing:**

Acceptance Test is performed with realistic data of the client to demonstrate that the software is working satisfactorily. Testing here is focused on external behavior of the system; the internal logic of program is not emphasized. In this project ‘VOIP’ I have collected some data and tested whether project is working correctly or not.

Test cases should be selected so that the largest number of attributes of an equivalence class is exercised at once. The testing phase is an important part of software development. It is the process of finding errors and missing operations and also a complete verification to determine whether the objectives are met and the user requirements are satisfied.

**CHAPTER-08**

**MAINTENANCE**

**8.1. Definition**

Maintenance is very important task & is poorly managed. Times spent and effort required in maintaining software and keeping it operational takes about 40 % to 70% of the total cost of the life cycle.

“Software maintenance is the activity that includes error corrections, enhancements of capabilities, deletion of obsolete capabilities and optimization.” Basically, any work done to change the software after it is in operation is considered to be maintenance. Its purpose is to preserve the value of the software.

**8.2. Categories**

**8.2.1. Corrective Maintenance**

It means modifications made to the software to correct the defects. Defects can result from design errors, logic errors, coding errors, data processing errors and system performance errors.

**8.2.2. Adaptive Maintenance**

It includes modifying the software to match changes in the ever-changing environment. Environment refers to the totality of all conditions and influences which act from outside upon the software.

**8.2.3. Perfective Maintenance**

It means improving processing efficiency or performances, or restructuring the software to improve changeability.

**8.3. Process**

The process of maintenance for given software can be divided into four stages as follows:

* **Program understanding:**It consists of analyzing the program in order to

understand it. The ease of understanding the program is primarily affected by complexity and documentation of the program.

* **Generate particular maintenance proposal**: The ease of generating the maintenance proposal is primarily affected by extensibility of the program.
* **Account for ripple effect:**If any change is made to any part of the system, it may affect the other parts also. Thus, there is a kind of ripple effect from the

location of modification to the other parts of the software. The primary feature affecting the ripple effect is stability.

* **Modified program testing*:*** The modified program is to be tested again and

again to check that the software has enhanced and reliability is validated.

**CHAPTER-09**

**SCREEN-SHOTS OF FORMS**

**CHAPTER-10**

**CONCLUSION AND SCOPE**

**CONCLUSION**

**ONLINE ART GALLERY** is a application software and it is very helpful for the art lovers and others who wants to know the addresses where this kind of arts will we sold.

This application helps the end-users to search their arts and paintings and they can place order for the selected pieces. The end-user can also get the information about the art exhibition and the respective address, so, that they can visit to those exhibitions.

Art Gallery brings you an opportunity to view online art exhibitions at our [Online Art Gallery](http://www.paletteartgallery.com/contact-palette-art-gallery-india.asp)  we bring you details of all art exhibitions held in the past and the forthcoming show. The Online Art Gallery is updated daily, so the user can view and buy the latest collection of contemporary art online from any where in the world. You can view and buy the latest Indian contemporary art collection available at their exhibitions and also at their online gallery.

**FUTURE SCOPE :**

* This project “[Online Art Gallery](http://www.paletteartgallery.com/contact-palette-art-gallery-india.asp) ” gives sopes for the further enhancement.
* The product can enter any user requirement.
* The application is currently stand alone and can be extended for the web.
* The product creation part is to done in a more elaborate manner.

**CHAPTER-11**

**BIBLIOGRAPHY**

**BIBLIOGRAPHY**

**References and Resources:**

**BOOKS:**

1 System Analysis & Design- Elias M.Awad (Galgotia Publication)

2 SQL SERVER 2000 -Gunderloy & Jorden (BPB Publication)

3 VB.Net Programming -Steven Holzner (Black Book)

**Web Sites**

* <https://www.lovelycode.com>
* <https://codeproject.com>
* <https://elearning.nic.in>
* <https://google.com>