VISVESVARAYA TECHNOLOGICAL UNIVERSITY

JNANA SANGAMA, BELAGAVI – 590 018

A Report on

ART GALLERY

Submitted in partial fulfillment of the requirements as a part of the DBMS Laboratory for the award of degree of

Bachelor of Engineering
in
Information Science and Engineering

Submitted by

AKSHATHA N T
1RN16IS013

HARSHITHA N
1RN16IS037

Faculty Incharge
Kusuma R
Assistant Professor
Dept. of ISE, RNSIT

Lab Incharge
R Rajkumar
Assistant Professor
Dept. of ISE, RNSIT

Department of Information Science and Engineering
RNS Institute of Technology
Channasandra, Dr. Vishnuvardhan Road, RR Nagar Post,
Bengaluru – 560 098

2018 – 2019

RNS Institute of Technology

Channasandra, Dr.Vishnuvardhan Road, RR Nagar Post,
Bengaluru – 560 098

DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that the mini project report entitled ***ART GALLERY*** has been successfully completed by **AKSHATHA N T** and **HARSHITHA N** bearing USN **1RN16IS013** and **1RN16IS037**, presently V semester student of **RNS Institute of Technology** in partial fulfillment of the requirements as a part of the DBMS Laboratory for the award of the degree ***Bachelor of Engineering in Information Science and Engineering*** under **Visvesvaraya Technological University, Belagavi** during academic year 2018 – 2019. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in the departmental library. The mini project report has been approved as it satisfies the academic requirements as a part of DBMS Laboratory for the said degree.

Mrs. Kusuma R	**Mr. R Rajkumar**	**Dr. M V Sudhamani**
Faculty Incharge	Lab Incharge	Professor and HOD

External Viva

Name of the Examiners	**Signature with date**
1. _____	_____
2. _____	_____

ABSTRACT

Art Gallery is an application used to sell art works of artists irrespective of their nationality, gender and other narrow considerations. The art and decor items are bought by the art gallery where visitors can view and buy them. The art works are either hosted for auction or sold for a fixed price. The art gallery displays a wide range of products from paintings to artifacts. Each customer has the right to purchase an art work and participate in auctions by submitting their bids. After each valid payment, the art work will be shipped within 15 days if the buyer chooses for shipment.

The aim is to automate its existing manual system by the help of computerized equipments and full-fledged computer software, fulfilling their requirements, so that their valuable information can be stored for longer period with easy accessing and manipulation of the same. Basically the project describes how to manage the gallery for good performance and better services.

ACKNOWLEDGMENT

The fulfillment and rapture that go with the fruitful finishing of any assignment would be inadequate without the specifying the people who made it conceivable, whose steady direction and support delegated the endeavors with success.

We would like to profoundly thank **Management** of **RNS Institute of Technology** for providing such a healthy environment to carry out this Project work.

We would like to thank our beloved Director, **Dr. H N Shivashankar** for his confidence feeling words and support for providing facilities throughout the course.

We would like to express my thanks to our Principal, **Dr. M K Venkatesha** for his support and inspired me towards the attainment of knowledge.

We wish to place on record my words of gratitude to, **Dr. M V Sudhamani,** Professor and Head of the Department, Information Science and Engineering, for being the enzyme and master mind behind my Project work.

We would like to express my profound and cordial gratitude to my Faculty incharge, **Mr. R Rajkumar**, Assistant Professor, Department of Information Science and Engineering for their valuable guidance, constructive comments and continuous encouragement throughout the Project work.

We would like to express my profound and cordial gratitude to my guide, **Mrs. Kusuma R**, Assistant Professor, Department of Information Science and Engineering for their valuable guidance in preparing Project report.

We would like to thank all other teaching and non-teaching staff of Information Science & Engineering who have directly or indirectly helped me to carry out the project work.

And lastly, we would hereby acknowledge and thank our parents who have been a source of inspiration and also instrumental in carrying out this Project work.

AKSHATHA.N.T **HARSHITHA.N**

1RN16IS013 **1RN16IS037**

TABLE OF CONTENTS

LIST OF FIGURES

ABBREVIATIONS

BOOTP	-	Bootstrap Protocol
BGP	-	Border Gateway Protocol
CMC	-	C Model Checker
DNS	-	Domain Name Service
DHCP	-	Dynamic Host Control Protocol
DART	-	Directed Automated Random Testing
D3S	-	Debugging Deployed Distributed Systems
DNSSD	-	DNS Service Discovery
D-ITG	-	Distributed Internet Traffic Generator
DNV	-	Declarative Network Verifier
IETF	-	Internet Engineering Task Force
IOT	-	Interoperability Testing
LLVM	-	Low Level Virtual Machine
MPE-SE	-	Multiple Packet Exchange – Symbolic Execution
PPP	-	Pont-to-Point Protocol
PC	-	Path Condition
RFC	-	Request for Comments
SAGE	-	Scalable, Automated Guided Execution
SM	-	Symbolic Map
SPE-SE	-	Single Packet Exchange – Symbolic Execution
TRAM	-	Tree Based Reliable Multicast
mDNS	-	MulicastDN

Chapter 1

INTRODUCTION

1.1 Background

A **database** is an organized collection of data, generally stored and accessed electronically from a computer system. Where databases are more complex they are often developed using formal design and modeling techniques.

The database management system (DBMS) is the software that interacts with end users, applications, the database itself to capture and analyze the data and provides facilities to administer the database. The sum total of the database, the DBMS and the associated applications can be referred to as a "database system". Often the term "database" is also used to loosely refer to any of the DBMS, the database system or an application associated with the database. The DBMS manages three important things: the data, the database engine that allows data to be accessed, locked and modified and the database schema, which defines the database's logical structure. These three foundational elements help provide concurrency, security, data integrity and uniform administration procedures. Typical database administration tasks supported by the DBMS include change management, performance monitoring/tuning and backup and recovery. Many database management systems are also responsible for automated rollbacks, restarts and recovery as well as the logging and auditing of activity.

1.2 Introduction to Art Gallery

Art gallery is the place where the arts and artifacts can be displayed for the visitor's view. People also comment on the arts that artists exhibit so that there will be some scope of improvement of the arts that are displayed. Sometimes it is difficult to maintain the details of the artists and their art details through the pen paper method. Art gallery management is an application that allows the owners to maintain the details of the artists and their works with great ease. The artifacts bought from the artists and vendors are displayed at the gallery. These products can be bought by the interested visitors. The gallery also hosts various events like puppet shows, auctions and many more. At the

auctions, the highest bidders get the auctioned item whose details will also be stored in the Art Gallery application for future references.

The artists can have their paintings, artifacts, decors and many other forms of art that can be displayed in the art gallery. Some of the art galleries would have artifacts, paintings of some years ago. But at some point of time if the artist's details are required, then this application would come in handy. Details of any artist and their products, sales made by the gallery, events hosted, all of it can be obtained with ease in just one mouse click.

Chapter 2

ER DIAGRAM AND RELATIONAL SCHEMA

2.1 E-R Diagram

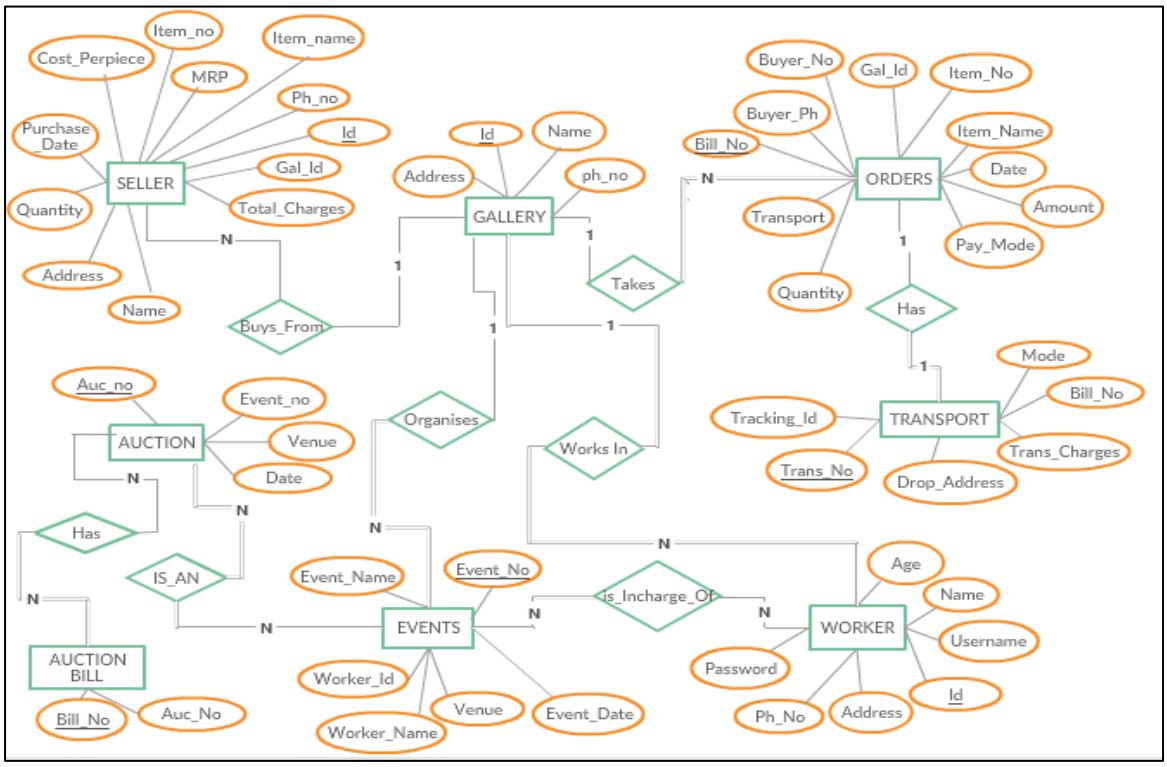


Fig .2.1: ER diagram of Art Gallery

The E-R Diagram in Fig. 2.1 describes entities, attributes and relationships.

- Entity types like GALLERY and ORDERS are in rectangular boxes.
- Relationships like HAS and WORKS_IN are in diamond boxes, attached to entity types with straight lines.
- Attributes are shown in ovals, each attached by a straight line to entity or relationship type.
- Key attributes (like ID) are underlined.
- Component attributes of a composite attribute are attached to oval representing it.

2.1.1 E-R Diagram Relationships Description

1. **SELLER: GALLERY** is of cardinality N: 1 as one gallery can have more than one seller and therefore connected via 'Buys_From' relationship. There is no total participation of both entities as:
    - A gallery can exist without belonging to a seller.
    - A seller can exist without gallery that belongs to him/her.
2. **GALLERY: ORDERS** is of cardinality 1: N as 1 gallery can have N number of orders that are taken by it. They are connected via the relationship 'Takes'. There is total participation of ORDERS and GALLERY has partial participation as:
    - An order cannot exist without having a particular gallery to take it.
    - A gallery can exist without orders being taken.
3. **ORDERS: TRANSPORT** is of cardinality 1: 1 as 1 order can be transported only once. Also, each transportation detail belongs to specific product. They are connected via the relationship 'Has'. There is total participation of TRANSPORT and partial participation of ORDERS as:
    - An order may or may not require transportation.
    - A transportation cannot exist without an order being placed.
4. **EVENTS: AUCTION** is of cardinality N: N as events will have N auctions and gallery has N events. Also each auction is an event. They are connected vis the relationship 'IS_AN'. There is total participation of AUCTION and partial participation of EVENTS as:
    - An event can exist without having an auction being conducted.
    - An auction is an event.
5. **GALLERY: WORKER** is of cardinality 1: N as 1 gallery can have only N workers working. Also, each worker's information is distinct by their id. They are connected via relationship 'Works In'. There is total participation of WORKER and partial participation of GALLERY as:
    - A gallery can exist without workers working for the gallery.
    - A worker does not exist without the existence of a gallery to work for.
6. **WORKER: EVENTS** is of cardinality N: N as N number of workers work for N no of events that take place in gallery. They are connected by relationship 'Is_Incharge_of'. There is total participation of both entities as:
    - A worker cannot exist without working for any of the events.

• An event cannot take place without any workers working for it.

7. **GALLERY: EVENTS** is of cardinality 1: N as 1 gallery conducts N number of events. They are connected via relationship 'Organises'. There is total participation of EVENTS and partial participation of GALERY as:
   • A gallery can exist without organising events.
   • An event cannot take place without it being organized by a gallery.
8. **AUCTION: AUCTION BILL** is of cardinality N: N as N auctions an have N number of bills of the items that are purchased by the bidders. The relationship is 'Has'. There is total participation of AUCTION BILL and partial participation of AUCTION entity as:
   • An auction bill cannot exist without an auction being conducted.
   • No item might be sold at a particular auction conducted.

2.2 Relational schema diagram

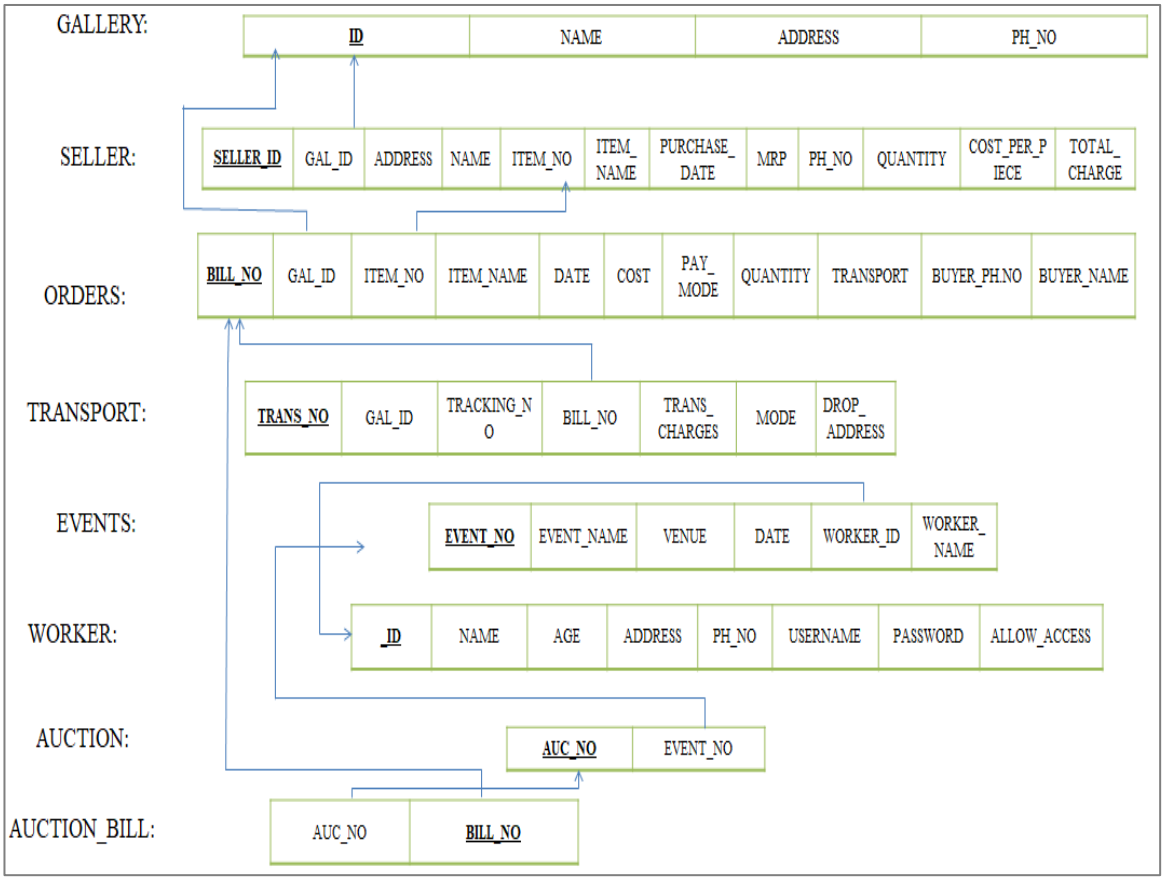


Fig.2.2: Relational Schema of Art Gallery

2.2.1 General Constraints

1) **NULL Constraint:** Attributes that are under NOT NULL constraints have to be filled compulsorily. Almost all the attributes in the project are under NOT NULL constraint.
2) **Entity Integrity Constraint:** This constraint makes sure that no primary key can have a NULL value assigned to it. The primary keys involved in the project include:
    - ID
    - NAME
    - GAL ID
3) **Referential Integrity Constraints:** A table in the back end of the project may have references pointing to an attribute in another table. For example: GAL_ID in the SELLER table refers to ID in GALLERY table. The various tables are also linked with multiple foreign keys which are all set to cascade any update or delete operation on the attribute in the main table. The various Foreign Key attributes are:
    - GAL_ID
    - WORKER_ID
    - BILL_NO

2.2.2 Schema Description

The above Fig.2.2 shows the relational schema of Art Gallery. It has the following entities.

1) **GALLERY:** This is a master table that consists of details like id, name, address and phone_no .Here only one gallery is taken into consideration with no branches.
2) **SELLER:** It is a tale of items that are bought by the store from various sellers and artists. It stores details like seller details, MRP, quantity ,cost per piece, item_no etc.
3) **ORDERS:** It is a table that consists of details of the products bought by the customer at the Art Gallery It consists of bill no, gal id, item name, item_no,cost etc.

4) **TRANSPORT:** The gallery facilitates the function of transportation for customers who opt for delivery. It stores details like tracking_id, drop address, bill_no etc.
5) **WORKER:** This table consists of details of the workers working in the gallery. It also has login credentials (username, password) with which the workers can log into the application. It also consists attributes like name, address, phone_no etc.
6) **EVENTS:** It is a table that consists information regarding the events that are conducted by the gallery. It has the following attributes, event_no ,venue ,workers incharge of the event etc.
7) **AUCTION:** Auction is an event which has a different table but links to the events. It consists of event number along with auction no.
8) **AUCTION BILL:** A separate database is maintained for details of the items that are purchased at the auctions.

STORED PROCEDURES: Various stored procedures are used to insert values into the tables, print the tuples of various tables, delete tuples etc.

VIEW: the view VIEWALL takes all the information of all the tables holding the gallery attributes and stores them conveniently in one place for the front-end queries to access easily.

Chapter 3

SYSTEM DESIGN

3.1 Tables

Gallery

Field	Type	Null	Key	Default
Id	numeric(6)	No	primary	Null
Name	Varchar(20)	No		Null
Address	Varchar(50)	No		Null
Ph_no	numeric(10)	No		Null

Seller

Field	Type	Null	Key	Default
Seller_id	numeric(5)	No	primary	Null
Gal_id	numeric(6)	No		Null
Address	Varchar(50)	No		Null
Name	Varchar(20)	No		Null
Item_no	numeric(3)	No		Null
Item_name	Varchar(15)	No		Null
Purchase_Date	Date	No		Null
Phone	numeric(10)	No		Null
Mrp	Int	No		Null
Total_Charge	Int	No		Null
Quantity	Int	No		Null
Cost_Per_Piece	Int	No		Null

Orders

Field	Type	Null	Key	Default
Bill_no	Int	No	primary	Null
Gal_id	numeric(6)	No		Null
Item_no	numeric(3)	No		Null
Item_name	Varchar(15)	No		Null
Date	Date	No		Null
Cost	Int	No		Null
Pay_mode	Varchar(10)	No		Null
Quantity	Int	No		Null
Transport	Bit	No		Null
Buyer_ph	numeric(10)	No		Null
Buyer_name	Varchar(20)	No		Null

Transport

Field	Type	Null	Key	Default
Trans_no	numeric(10)	No	primary	Null
Tracking_no	numeric(18)	No		Null
Bill_no	Int	No		Null
Trans_charges	Int	No		Null
Mode	Char	No		Null
Drop_address	Varchar(50)	No		Null
Gal_Id	numeric(6)	No		Null

Event

Field	Type	Null	Key	Default
Event_no	numeric(2)	No	primary	Null
Venue	Varchar(50)	No		Null
Event_date	Date	No		Null
Worker_id	numeric(3)	No		Null
Worker_name	Varchar(20)	No		null

Worker

Field	Type	Null	Key	Default
Id	Int	No	primary	Null
Name	Varchar(20)	No		Null
Age	Int	No		Null
Address	Varchar(50)	No		Null
Ph_no	numeric(10)	No		Null
Username	numeric(6)	Yes		Null
Password	numeric(8)	Yes		Null
Allow_Access	numeric(1)	Yes		Null

Auction

Field	Type	Null	Key	Default
Auc_no	numeric(1)	No	primary	Null
Event_no	numeric(2)	No		Null

Auction bill

Field	Type	Null	Key	Default
Bill_no	Int	No	primary	Null
Auc_no	numeric(1)	No		Null

Chapter 4

IMPLEMENTATION

4.1 Software Requirements

4.1.1 Front End Software

C# is a simple, modern, general-purpose, object-oriented programming language developed by Microsoft within its .NET initiative led by Anders Hejlsberg. This tutorial will teach you basic C# programming and will also take you through various advanced concepts related to C# programming language.

Let us look at a simple code that prints the word “hello world”

```
Using System;
namespaces HelloWorldApplication{
class HelloWorld{
Static Void Main (String[] args){
/*my first program in c#*/
Console.WriteLine (“Hello World”);
Console.ReadKey ();
}
}
}
```

- The using keyword is used to include the System namespace in the program.
- The HelloWorldApplication namespace contains the class HelloWorld.
- The class HelloWorld contains the data and method definitions that your program uses. However, the HelloWorld class has only one method Main.
- The Main method specifies the statement Console.WriteLine("Hello World");

  WriteLine is a method of the Console class defined in the System namespace. This statement causes the message "Hello, World!" to be displayed on the screen.
- The last line Console.ReadKey **();** is for the VS.NET Users. This makes the program wait for a key press and it prevents the screen from running and closing quickly when the program is launched from Visual Studio .NET.

4.1.2 Back End Software

SQL is a small, fast and embeddable database where the database engine and the interface are combined into a single library. It also has the ability to store all the data in a single file. It is a software library that implements a self-contained, serverless, zero-configuration, transactional SQL database engine. SQL is the most widely deployed SQL database engine in the world. All the SQL statements start with any of the keywords like select, create, update, insert, delete, alter, drop etc., and all the statements end with a semicolon (;).

4.2 Code Segment

4.2.2 Connection

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Drawing;
using System.Data;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.SqlClient;

namespace WindowsFormsApp4
{
      public partial class takeorder_uc : UserControl
   {
    private static takeorder_uc _instance;

      public static takeorder_uc Instance
      {
         get
         {
            if (_instance == null)
            {
               _instance = new takeorder_uc();
            }
            return _instance;
              }
      }
```

```
        public takeorder_uc()
        {
            InitializeComponent();
    }
```

SqlConnection con = new SqlConnection(@"Data Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=|DataDirectory|\Database1.mdf;Integrated Security=True;Connect Timeout=30");

```
private void takeorder_uc_Load(object sender, EventArgs e)
{
        panel1.Hide();
}
//submit orders
 private void SUBMIT_Click(object sender, EventArgs e)
 {
        orders_uc o = new orders_uc();

        SqlCommand cmd = new SqlCommand("take_orders_sp", con);
        cmd.CommandType = CommandType.StoredProcedure;

        cmd.Parameters.AddWithValue("@bill_no",bill_txt.Text);
        cmd.Parameters.AddWithValue("@date", date_txt.Text);
        cmd.Parameters.AddWithValue("@gal_id", gal_txt.Text);
        cmd.Parameters.AddWithValue("@item_no", itemno_txt.Text);
        cmd.Parameters.AddWithValue("@item_name", itemnm_txt.Text);
        cmd.Parameters.AddWithValue("@quantity", quan_txt.Text);
        cmd.Parameters.AddWithValue("@cost", cost_txt.Text);
        cmd.Parameters.AddWithValue("@pay_mode", pay_txt.Text);
        if (yes.Checked)
        {
          cmd.Parameters.AddWithValue("@transport","y");
        }
        else if(no.Checked)
        {
          cmd.Parameters.AddWithValue("@transport", "n");
        }
        cmd.Parameters.AddWithValue("@buyer_name", bn_txt.Text);
        cmd.Parameters.AddWithValue("@buyer_contact", bp_txt.Text);

        con.Open();
         try
        {
```

```
            cmd.ExecuteNonQuery();
          }
          catch(Exception ex)
          {
            MessageBox.Show("              <<INVALID SQL OPERATION>>:  \n" + ex);
          }

          con.Close();
          o.refresh_data();

          int amount = int.Parse(cost_txt.Text);
         MessageBox.Show("BILL AMOUNT IS:" + amount);

          bill_txt.Clear();
          date_txt.Clear();
          itemno_txt.Clear();
          itemnm_txt.Clear();
          quan_txt.Clear();
          cost_txt.Clear();
          pay_txt.Clear();
          bn_txt.Clear();
          bp_txt.Clear();
}

//enter transport
private void no_CheckedChanged(object sender, EventArgs e)
{
   if (no.Checked)
   {
      panel1.Hide();
    }
 }

 private void yes_CheckedChanged(object sender, EventArgs e)
{
   if (yes.Checked)
       panel1.Show();
 }
      private void enter_Click(object sender, EventArgs e)
      {
          SqlCommand cmd = new SqlCommand("transport_sp", con);
          cmd.CommandType = CommandType.StoredProcedure;
```

```
            cmd.Parameters.AddWithValue("@trans_no", tano_txt.Text);
            cmd.Parameters.AddWithValue("@gal_id", gal_txt.Text);
            cmd.Parameters.AddWithValue("@tracking_no", track_txt.Text);
            cmd.Parameters.AddWithValue("@bill_no", bill_txt.Text);
            cmd.Parameters.AddWithValue("@trans_charges", tc_txt.Text);
            cmd.Parameters.AddWithValue("@trans_mode", mode_txt.Text);
            cmd.Parameters.AddWithValue("@drop_address", drop_txt.Text);

            con.Open();

             try
            {
               cmd.ExecuteNonQuery();
            }
            catch (Exception ex)
            {
               MessageBox.Show("              <<INVALID SQL OPERATION>>:  \n" + ex);
            }

            con.Close();

            int trans=int.Parse(tc_txt.Text);
            MessageBox.Show("TRANSPORT CHARGES ARE:" + trans);

            tano_txt.Clear();
            textBox4.Clear();
            track_txt.Clear();
            bill_txt.Clear();
            tc_txt.Clear();
            mode_txt.Clear();
            drop_txt.Clear();
        }
    }
}
```

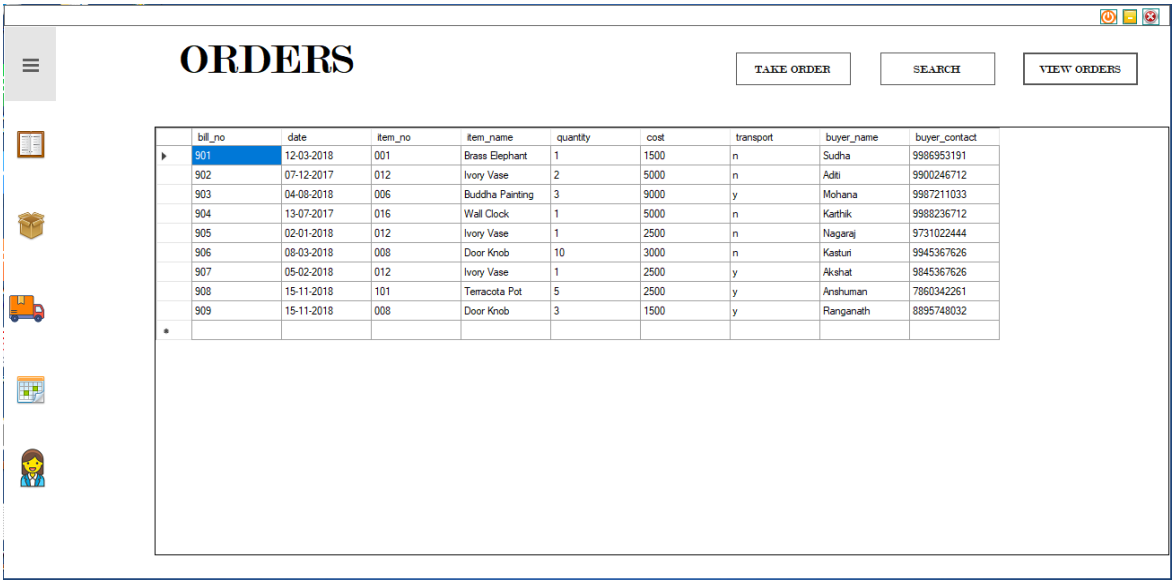
4.3 Snapshots

Fig.4.1: Login form

Fig.4.1 shows the login form of the Art Gallery application.

Fig.4.2: Main Menu

Fig.4.2 shows the main menu of the application from where the logged in worker can access various options like taking orders, viewing the products, updating an event, adding a auction, etc.



bill_no	date	item_no	item_name	quantity	cost	transport	buyer_name	buyer_contact
901	12-03-2018	001	Brass Elephant	1	1500	n	Sudha	9986953191
902	07-12-2017	012	Ivory Vase	2	5000	n	Aditi	9900246712
903	04-08-2018	006	Buddha Painting	3	9000	y	Mohana	9987211033
904	13-07-2017	016	Wall Clock	1	5000	n	Karthik	9988236712
905	02-01-2018	012	Ivory Vase	1	2500	n	Nagaraj	9731022444
906	08-03-2018	008	Door Knob	10	3000	n	Kasturi	9945367626
907	05-02-2018	012	Ivory Vase	1	2500	y	Akshat	9845367626
908	15-11-2018	101	Terracota Pot	5	2500	y	Anshuman	7860342261
909	15-11-2018	008	Door Knob	3	1500	y	Ranganath	8895748032

Fig.4.3: View Orders

Fig.4.3 shows the detailed list of orders billed at the gallery. This list does not include the bills issued at the auctions.

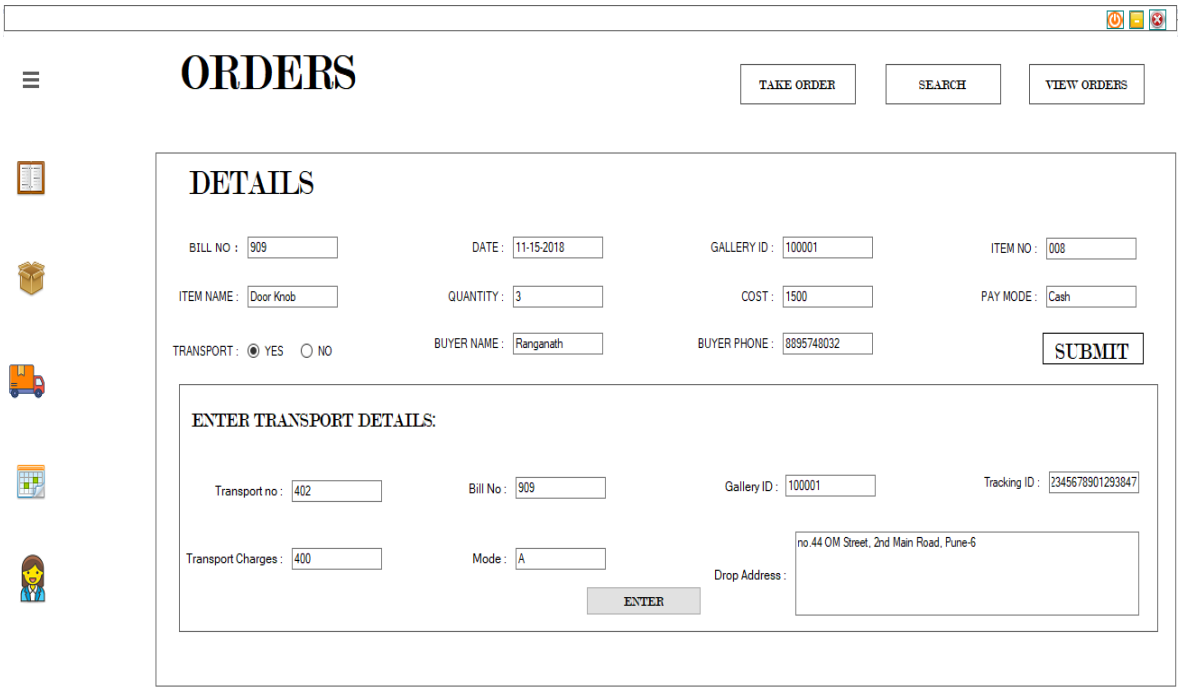


Fig.4.4: Take Orders

Fig.4.4 shows the form to take the orders of products bought at the gallery along with the transport requirements form.

Fig.4.5: Search Orders

Fig4.5 shows the page to enter bill number in order to search for a particular bill. Once the "OKAY" button is clicked, the bill details are displayed.

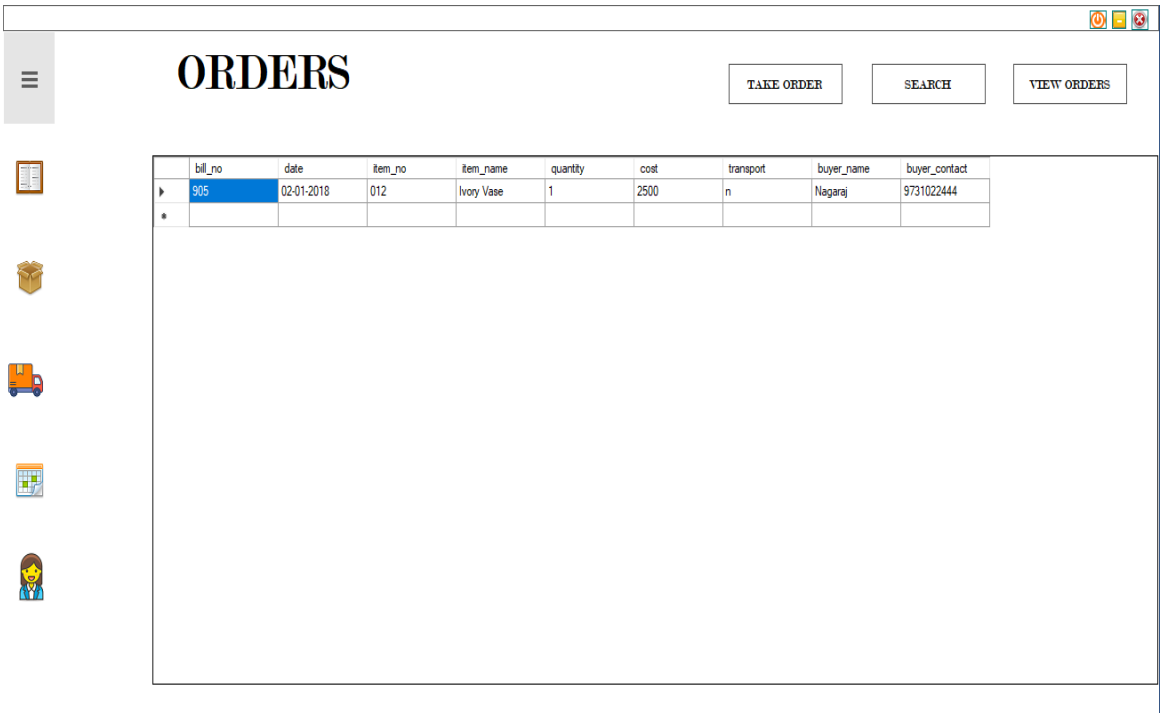


Fig.4.6: Show Search Orders

Fig.4.6 shows the particulars of the searched bill.

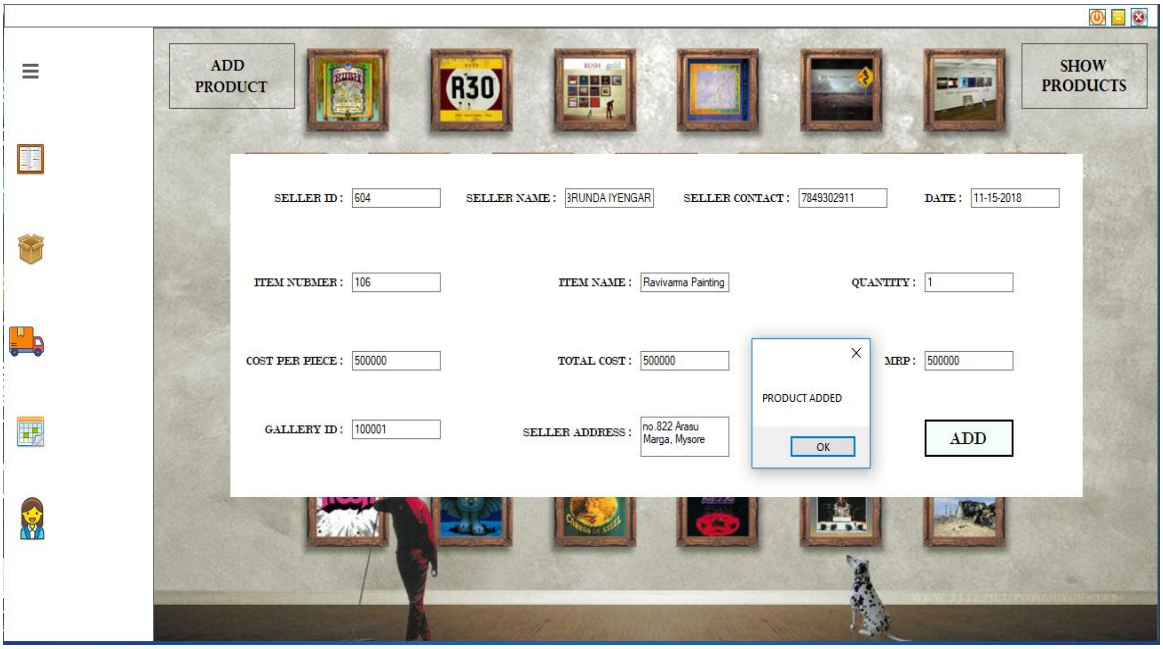


Fig.4.7: Add Product

Fig.4.7 show the form to add new products bought by the gallery.

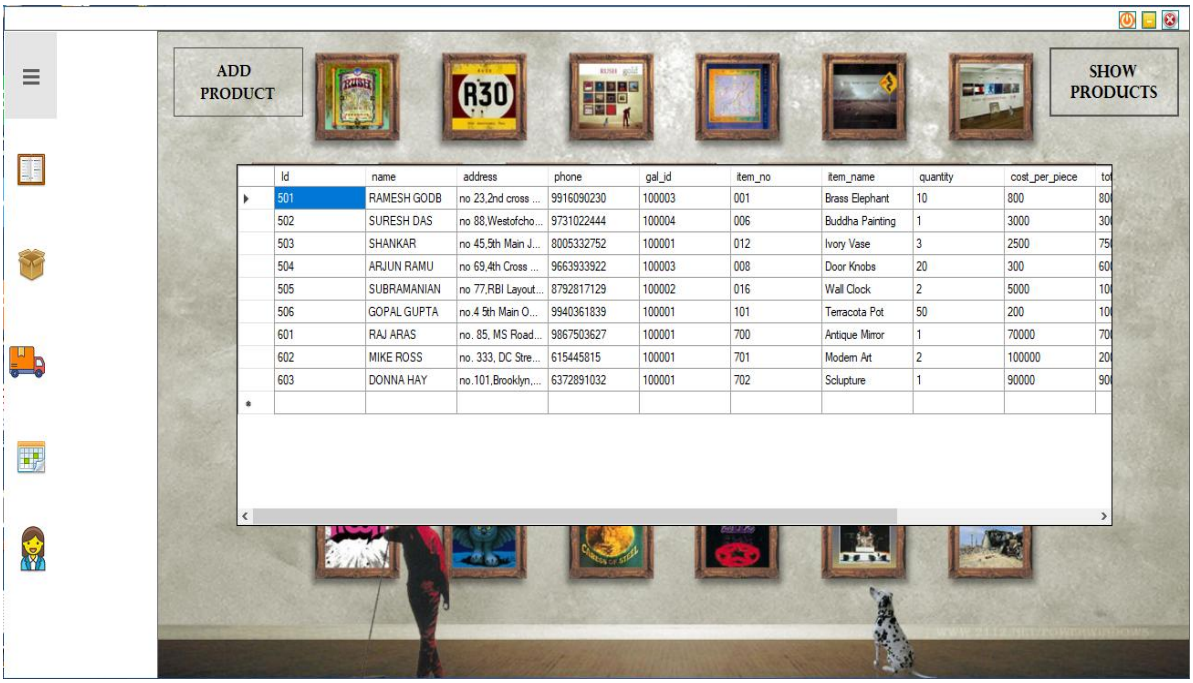


Fig.4.8: List Of Products

Fig.4.8 show the list of products bought by the gallery along with their seller's details.

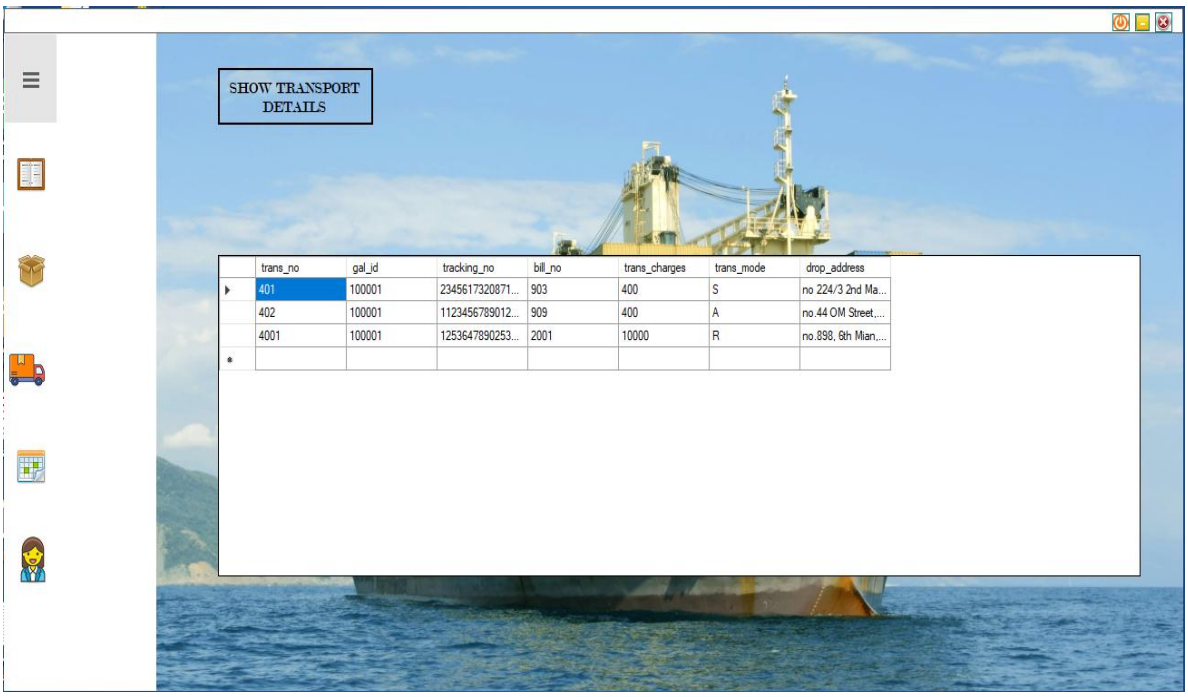


Fig.4.9: Transport Details

Fig.4.9 show the list of shipments. These are the details filled while taking the orders, if the customer opted for transport.

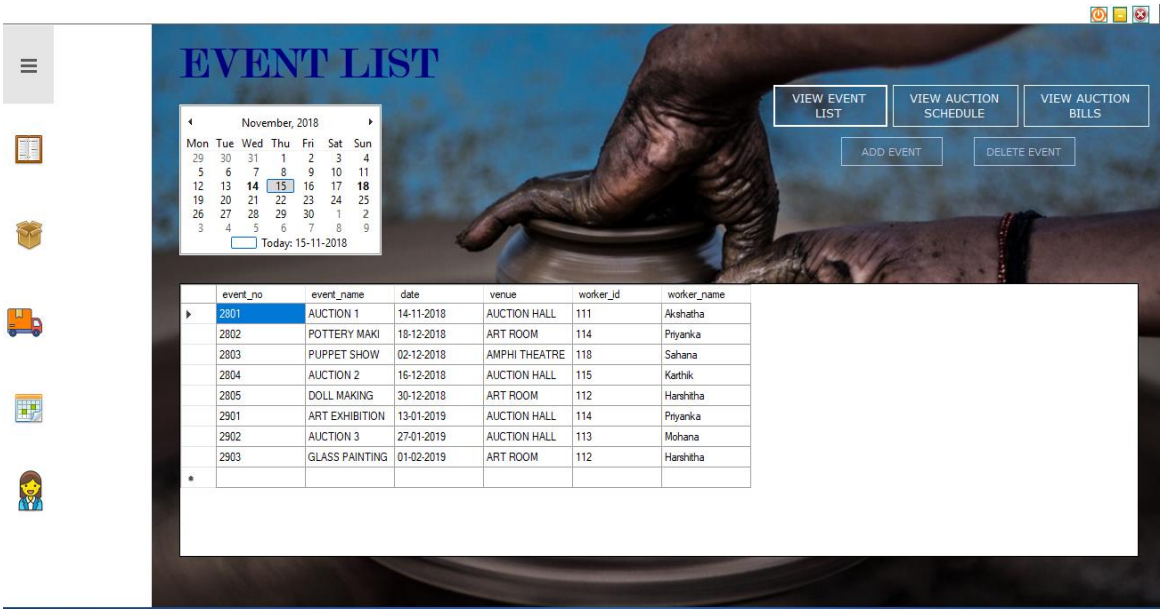


Fig.4.10: Event List

Fig.4.10 show the list of events conducted by the Art Gallery.

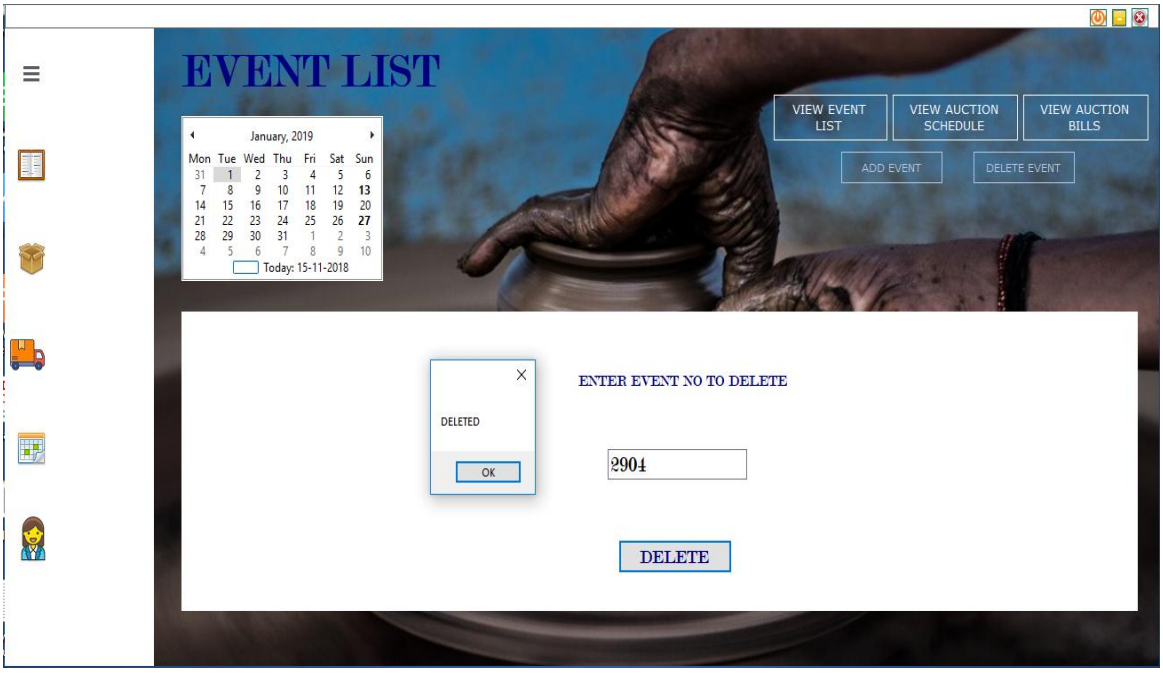


Fig.4.11: Delete Event

Fig.4.11 show the deletion of an event.

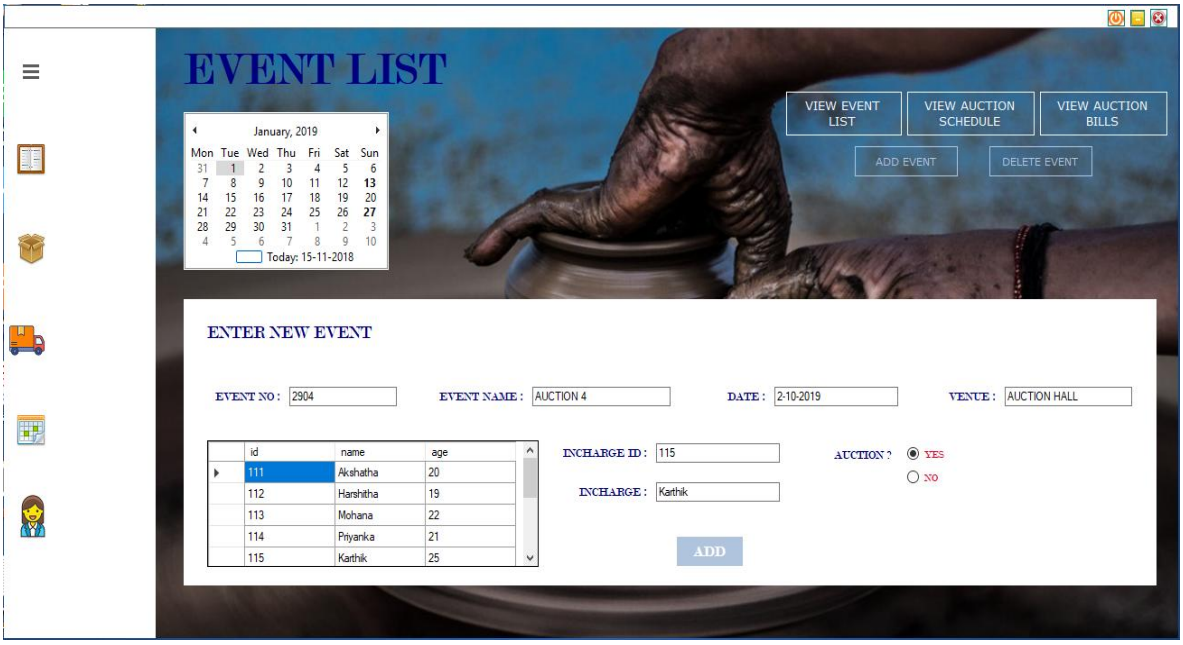


Fig.4.12: New Event

Fig.4.12 shows the addition of a new event.

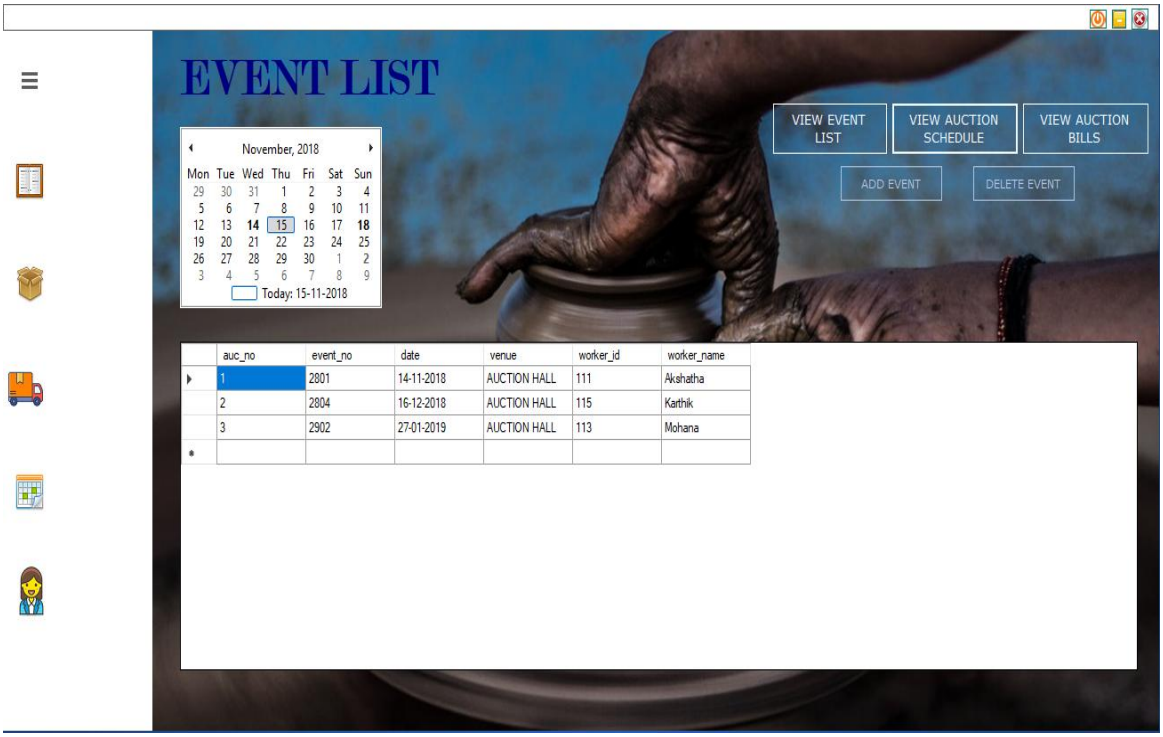


Fig.4.13: Auction List

Fig.4.13 shows the list of auctions held/to be held.

Fig.4.14: Worker data login

Fig.4.14 show the login page to access workers' data. This can be accessed only by the admin as shown.

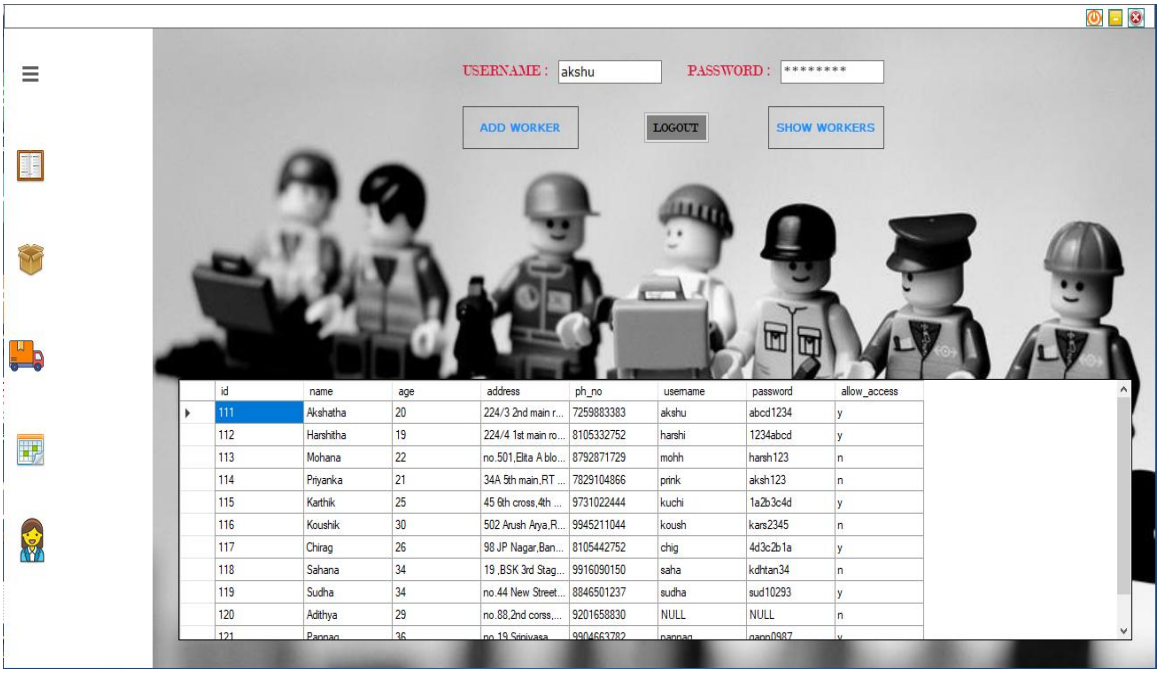


Fig.4.15: Worker List

Fig.4.15 shows details of the workers. Along with login username and password. It also tells if the worker is given access to the application or not.

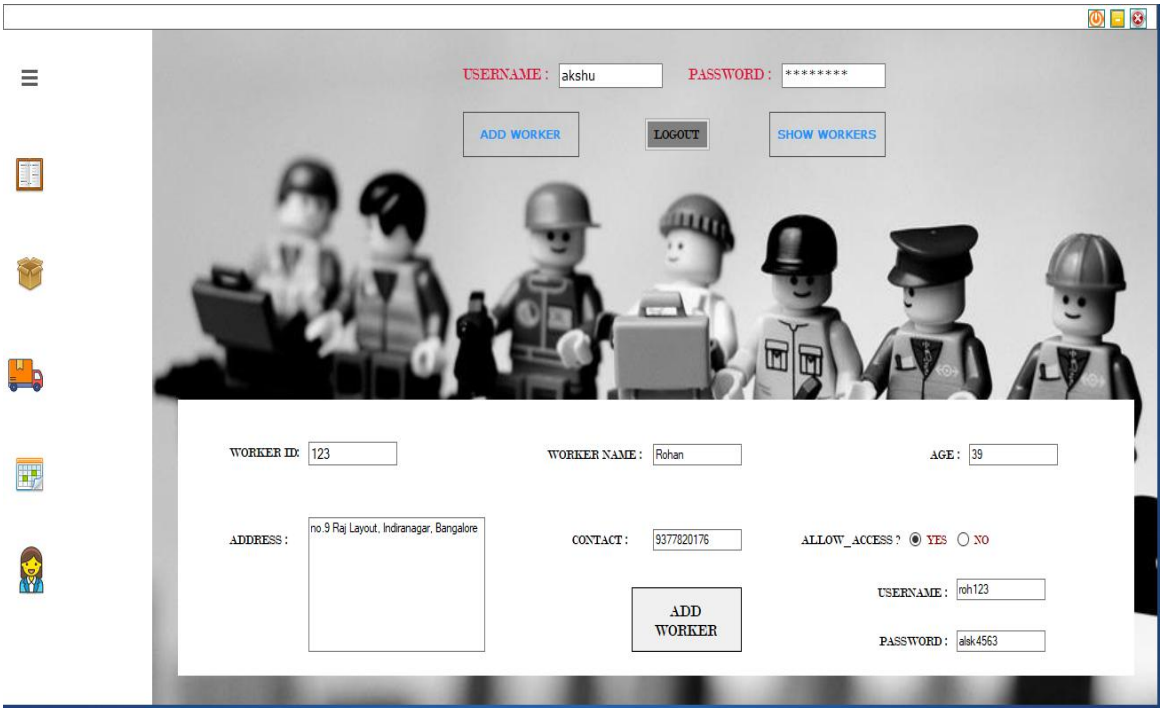


Fig.4.16: Add Worker

Fig.4.16 shows the form to add a new worker.

Chapter 5

APPLICATIONS

Following are some of the applications of Art Gallery:

- Managing sales in any retail store.
- Keeping a track of workers at the store.
- Database of events conducted throughout the year.
- Easy access to list of products and their availability at the store along with their details.

All the above are available under a single platform. Hence, allowing the running of a store smooth and efficient. The idea is to expand the application to as many genres of store as possible from a small grocery store to a large museum.

Chapter 6

CONCLUSION AND FUTURE ENHANCEMENTS

Our project is a an attempt to make the structure and working of any retail store simpler and user-friendly. In this scenario, all the undertakings of an Art Gallery are achieved in a constructive manner.

Give the right guidance and support; we hope to enhance its applications and availability.

Future Enhancements:

- Printing generated bills.
- Advanced software for art gallery including more facilities like market values of products, connections to artists and vendors across the world.
- Hosting the platform on online servers.
- Integrate multiple load balances to distribute the loads of the system.
- Create the master and slave database structure to reduce the overload of the database queries.
- Implement the backup mechanism for taking backup of codebase and database on a regular basis on different servers.
- E-Commerce website.

REFERENCES

[1] Raghu Ramakrishnan and Johannes Gehrke, Database Management Systems, McGREW HILL, 3rd Edition.

[2] Ramez Elmasri and Shamkant B. Navathe, Fundamentals of Database Systems, Pearson, 7th Edition.

[3] Herbert Schildt, Java: The Complete Reference, McGRAW HILL, 7th Edition.

[4] www.stackoverflow.com

[5] www.mysql.com/doc

[6] https://docs.microsoft.com

[7] https://www.geeksforgeeks.org