**CHAPTER 1**

**1. INTRODUCTION**

**1.1 OBJECTIVE**

The problem with public auction is that the participation of the general public is very limited. The aim of the project is to socialize the auction so that people from far & wide and even across the continent can participate in it. The "E-Auction" site is developed with a vision to wipe out the inherent problems of "Conventional Auction House". The salient features of the site are as follows:

1. Paperless Auction System
2. It's accessible to everyone, at any time no matter where they are
3. Reliable user validation & checking.
4. Easy online settlement.

"E-Auction" is designed in such a way that it is as user friendly as possible. So any aspiring bidder or seller can visit the site and engage in bidding with least effort.

**1.2 SCOPE**

"E-Auction" is an online auction web site aimed at taking the auction to the fingertips of aspiring bidders there by opening up the doors of the "OPEN Auction House' to a wider cross section of Art Lovers and Antique Collectors. This site also acts as an open forum where buyers and sellers can come together and exchange their products. The site makes sure that the sellers get a fair deal and buyers get a genuine product.

.

**CHAPTER 2**

**2. GENERAL DESCRIPTION**

**2.1 PRODUCT PERSPECTIVE**

The proposed computerized "E-Auction" site has made auction process simple. The only pre-condition is that the user must register and authenticate before he/she can take part in the bidding process. The system uses HTTP forms authentication which creates a session cookie for any signed in user. Throughout the span of the session the cookie remains valid until the user logs out.

An auction house needs to have products to auction, so in the proposed system this is done using product registration module. The module is open to user who is registered sellers and they need to authenticate before they register any product. The user can put a product for sales or they can buy a product through bidding.

Admin if this page can login into the page and view the bidding status of all the products and they can also view all the user details.

**2.2 USER CHARACTERISTICS**

The product should be user friendly; hence the user should not need any software and hardware knowledge. They are assumed to have basic knowledge of computers. Administrators of the system should have more knowledge of internal modules of the system and are able to rectify small problems that may arise due to disk crashes, power failures and other catastrophes. Friendly user interface, user guide must be sufficient to educate the users on how to use this product without any problems or difficulties.

**2.3 DESIGN AND IMPLEMENTATION CONSTRAINTS**

* The information of users who have registered are stored in database.
* Apache Tomcat 7.0 Web Server.
* It can be operated at any time.
* User can bid for the product under auction before the end date.

**CHAPTER 3**

**3. REQUIREMENTS**

**3.1 FUNCTIONAL REQUIREMENTS**

**3.1.1 User Module**

**Prerequisite (**user signed in) for all requirements below

**Requirement ID** R1.01.01

**Title** Login in module

**Description** This action is done to login to the site

**Priority** 1

**Requirement ID** R1.01.02

**Title** New User Register

**Description** This action is done to enter the details of new user.

**Priority** 1

**Requirement ID** R1.01.03

**Title** Main Module (Put for Sale)

**Description** This action is done to put products for sales in auction

**Priority** 1

**Requirement ID** R1.01.04

**Title** Main Module (Search Product)

**Description** This action is done to search products for bidding

**Priority** 1

**Requirement ID** R1.01.05

**Title** Main Module (Bidding Center)

**Description** This action is done to bid the products at highest price.

**Priority** 1

**Requirement ID** R1.01.05

**Title** Main Module (My Purchase)

**Description** This action is done to display the products bought in bidding

**Priority** 1

**Requirement ID** R1.02.03

**Title** Main Module (My Profile)

**Description** This action is done to view and edit user profile

**Priority** 2

**3.2 NON-FUNCTIONAL REQUIREMENTS**

**3.2.1 Error Handling**

* E-Auction product shall handle expected and non-expected errors in ways that prevent loss in information.

**3.2.2 Performance Requirements**

* The system accommodates high number of products and their prices.
* Responses to view information shall take no longer than 5 seconds to appear on the screen.

**3.2.3 Safety Requirements**

* Maintainability
* Reliability.

**3.2.4 Security Requirements**

* System will use secured database.
* Product prices are updated only with user’s knowledge

**3.3 USER INTERFACE**

**3.3.1 LOGIN MODULE**

User can login with his user id and password and check all the functionality from the menu. Refer screen print below for the User login.

**3.4.1 Hardware Requirements**

Hard disk: 250 GB and above

RAM: 2 GB and above

Processor: i3 and above

**3.4.2 Software Requirements**

Operating system: windows 7 and above

Web server: APACHE TOMCAT-8

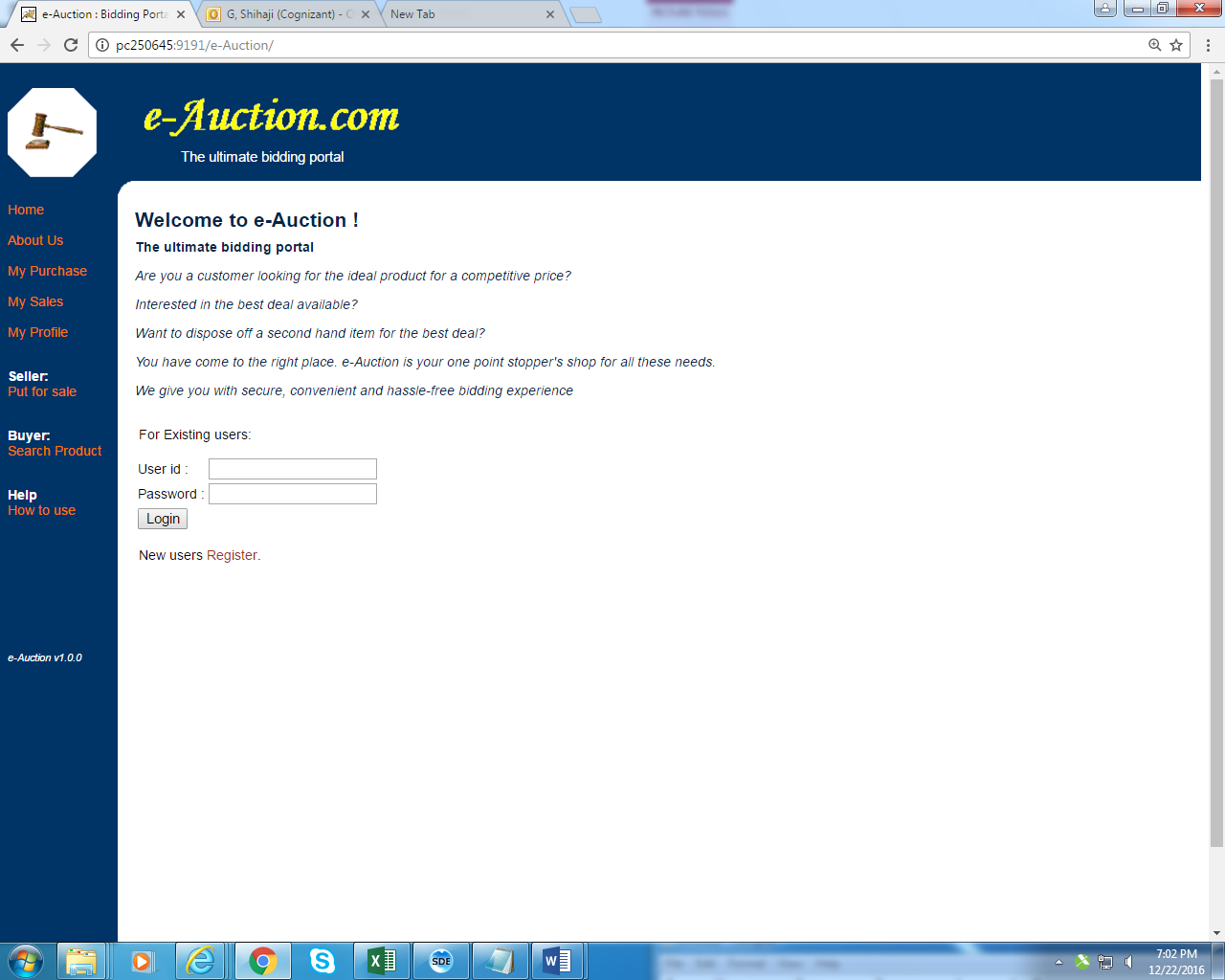
RDBMS: MySQL

### Login Module:

|  |  |  |
| --- | --- | --- |
| **Field** | **Validation** | **Error Message When Validation Fails** |
| User Id | Cannot be Null | Please enter name |
| Length less than 20 characters | Value cannot be more than 20 characters |
| Only contain alphanumeric characters | Invalid user id |
| Password | Cannot be Null | Please enter Password |
| Length less than 20 characters | Value cannot be more than 20 characters |
| Only contain alphanumeric characters | Invalid Password |

### Sample Data:

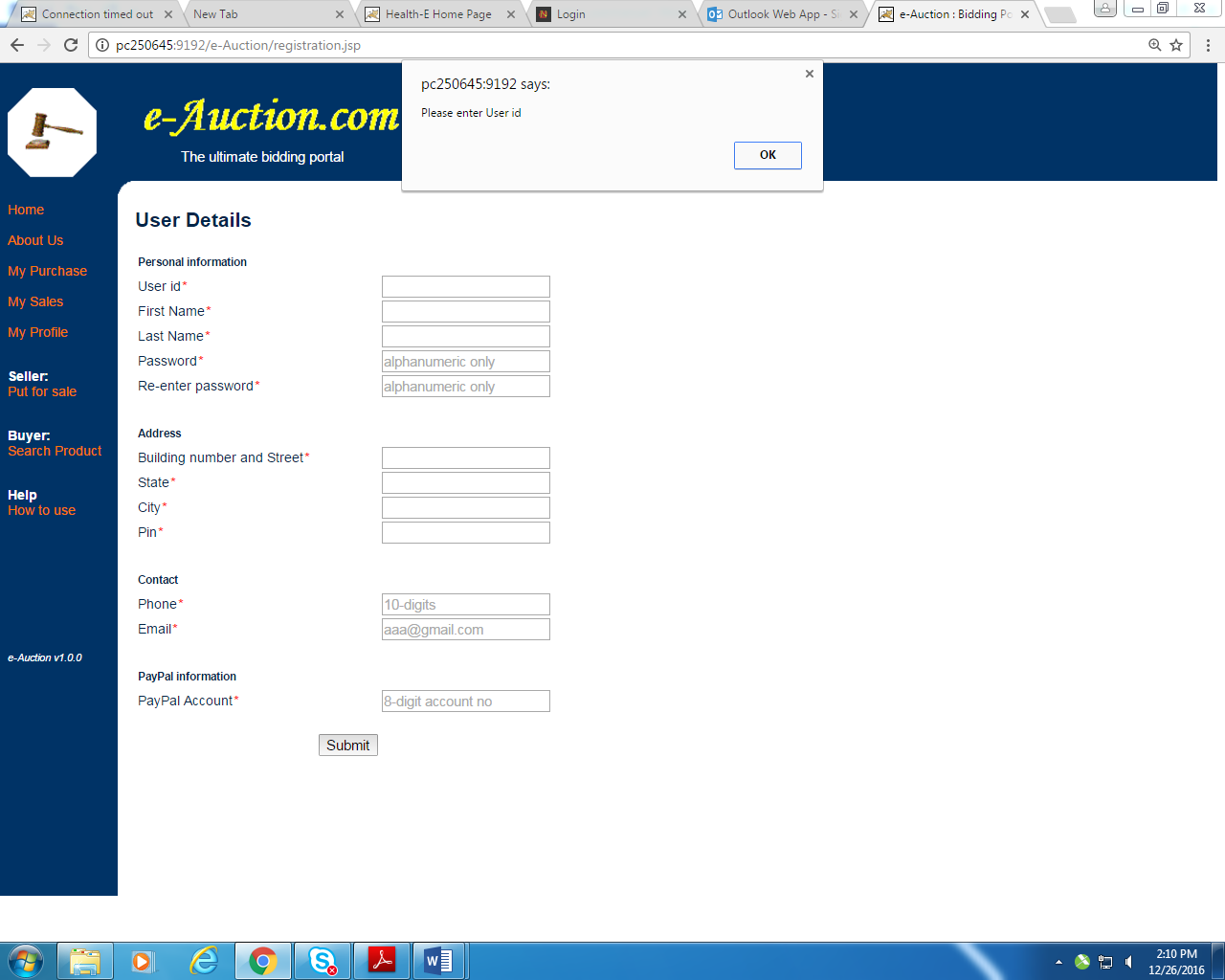
|  |  |
| --- | --- |
| User Id | Password |
| manikanta | manikanta |
| sanjay | Pass1234 |
| Bargavi | pass321 |
| Karthi12 | Welcome |



**Figure 3.1 : Login Module**

**3.3.2 NEW USER REGISTRATION:**

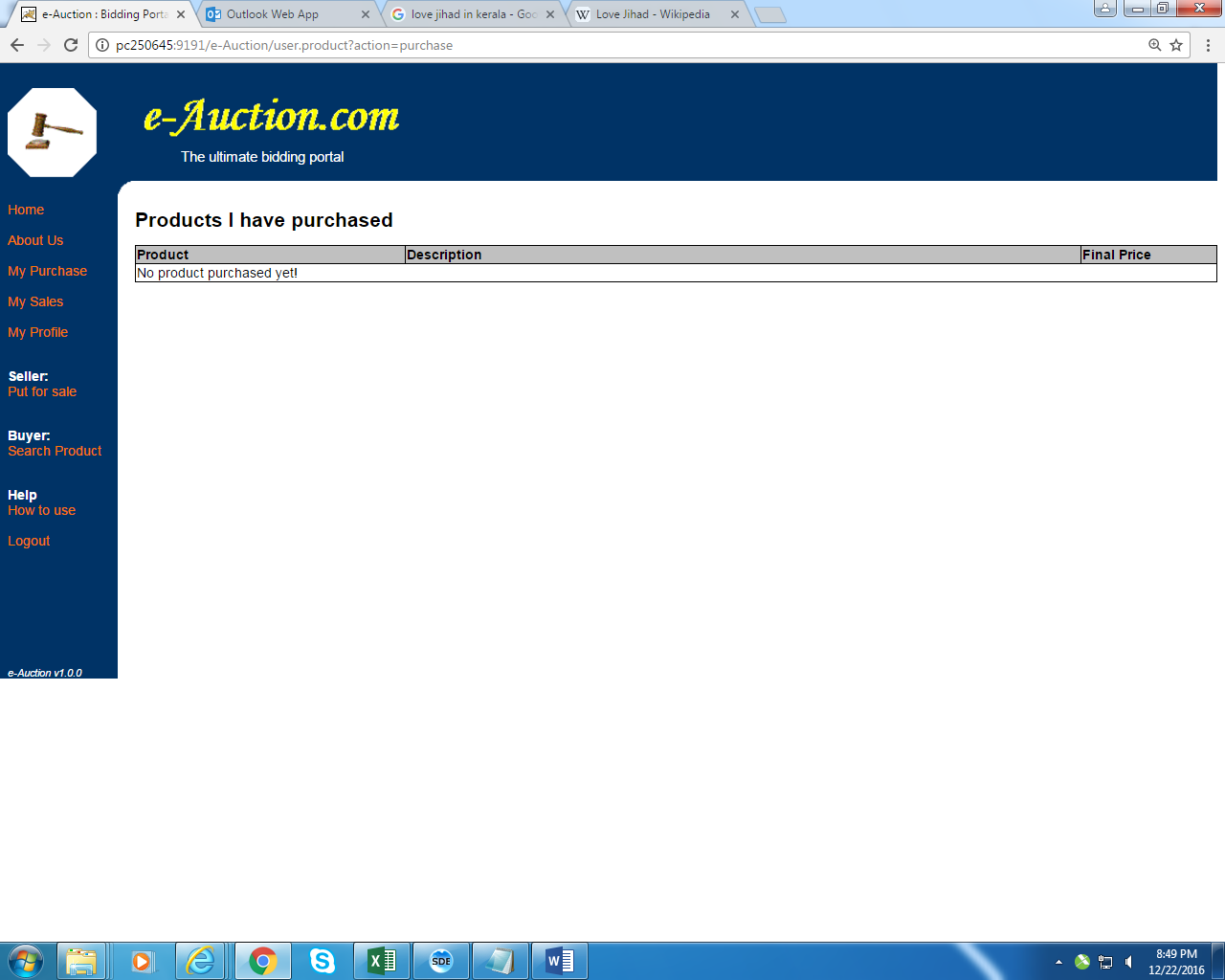
New user can be enrolled to the application by clicking **Register** link .User will be guided to fill registration form. User has to fill in all the mandatory field, not adhering to which lead to error message dialog to pop up. User can cancel the pop up by clicking **ok** in the dialog box and fill in the required field. When user leaves User name field blank, error dialog pops up. If the user enters all the correct credentials and click submit button user will redirected to welcome page.



**Figure 3.2 : New User Registration**

**3.3.3 MY PURCHASE AND MY SALE:**

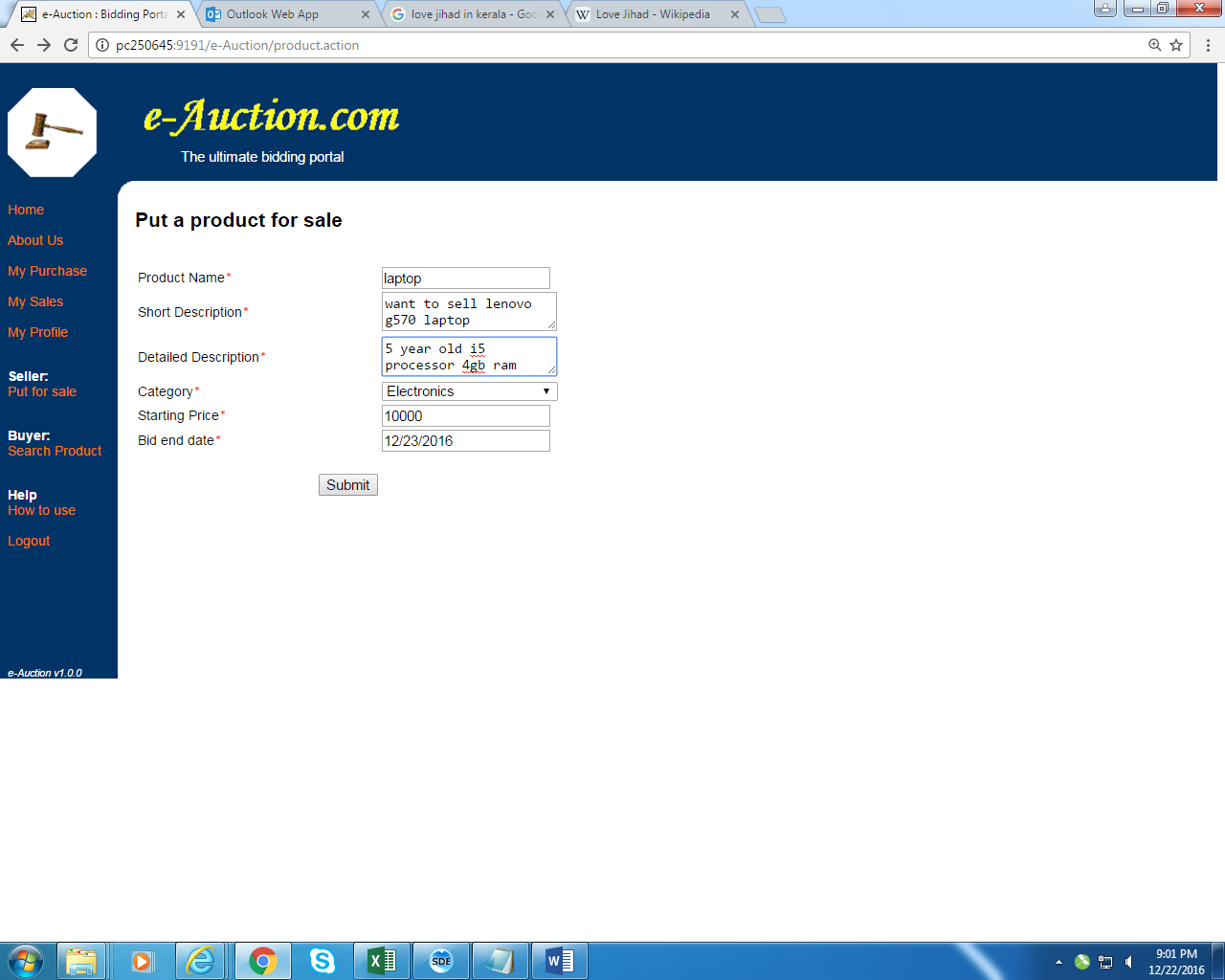
This module helps the user in viewing their products that are sold to other bidders and the products that they have bought in auction.



**Fig:3.3**

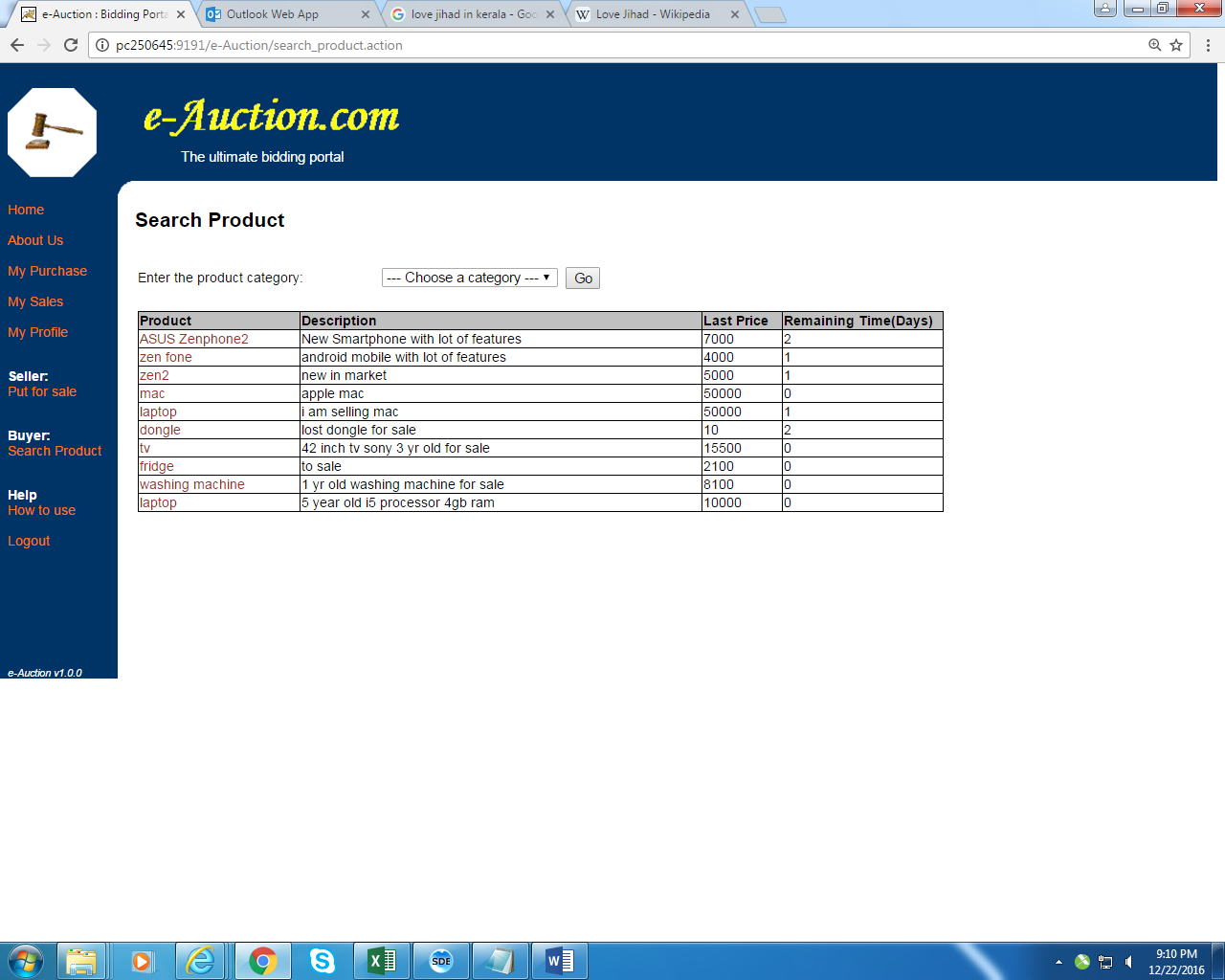
**3.3.4 PUT FOR SALE:**

By clicking put for sale link user can put his item to be sold by maximum bid. After user clicks submit button after filling all the mandatory fields like category from drop box ,choose a starting price for his product , when he want to close the bid etc, he will be redirected to welcome page with confirmatory message

**Fig: 3.4**

**3.3.5 SEARCH PRODUCT:**

By clicking search product he can buy any product by searching the product in choose a category drop box. User will be provided with the list of all the products available in that category, he/she can click hyper link of any product he/she likes . User will be redirected to Bidding Center to bid for the product which should be greater than the starting price. His bid will be posted in the bidding table with bid closing date



**Fig: 3.5**

**3.3.6 BIDDING CENTER:**

This is the module where multiple users bid for a product and the product is sold for the person who quotes for the highest rate within the end date.



**Fig: 3.6**

**CHAPTER 4**

**4. DETAILED DESIGN**

**4.1 ARCHITECTURAL DESIGN**

The architecture for any application is broken into three separate logical layers, each with a well - defined set of interfaces. The first tier is referred to as the presentation layer and typically consists of graphical user interface of some kind. The middle tier, or business layer, consists of application or business layer and the third layer- the data layer contains the data that is needed for the application. The middle tier is basically the code that the user calls upon to retrieve the desired data. The presentation layer then receives the data and formats it for display. This separation of application logic from the user interface adds enormous flexibility to the design of application. The third tier contains the data that is needed for the application.

**4.2 DATABASE DESIGN**

The overall objective in the development of database technology has been to treat data as an organizational resource and as an integrated whole. DBMS allow data to be protected and organized separately from other resources. Database is an integrated collection of data. The most significant form of data as seen by the programmers is data as stored on the direct access storage devices. This is the difference between logical and physical data.

Database files are the key source of information into the system. It is the process of designing database files, which are the key source of information to the system. The files should be properly designed and planned for collection, accumulation, editing and retrieving the required information.

The organization of data in database aims to achieve three major objectives: -

* Data integration.
* Data integrity.
* Data independence

The proposed system stores the information relevant for processing in the MS SQL SERVER 2000 database. This database contains tables, where each table corresponds to one particular type of information. Each piece of information in table is called a field or column. A table also contains records, which is a set of fields. All records in a table have the same set of fields with different information. There are primary key fields that uniquely identify a record in a table. There are also fields that contain primary key from another table called foreign keys.

**CHAPTER 5**

**5. TESTING**

**5.1 UNIT TESTING**

A Unit corresponds to a screen /form in the package. Unit testing focuses on verification of the corresponding class or Screen. This testing includes testing of control paths, interfaces, local data structures, logical decisions, boundary conditions, and error handling. Unit testing may use Test Drivers, which are control programs to co-ordinate test case inputs and outputs, and Test stubs, which replace low-level modules. A stub is a dummy subprogram.

**5.2 REGRESSION TESTING**

Each modification in software impacts unmodified areas, which results serious injuries to that software. So the process of re-testing for rectification of errors due to modification is known as regression testing.

**Installation and Delivery:**

Installation and Delivery is the process of delivering the developed and tested software to the customer. Refer the support procedures

**Acceptance and Project Closure:**

Acceptance is the part of the project by which the customer accepts the product. This will be done as per the Project Closure, once the customer accepts the product; closure of the project is started. This includes metrics collection, PCD, etc.

**5.3 VALIDATION**

Validation refers to the process of using the new software for the developed system in a live environment i.e., new software inside the organization, in order to find out the errors. The validation phase reveals the failures and the bugs in the developed system. It will be come to know about the practical difficulties the system faces when operated in the true environment.

By testing the code of the implemented software, the logic of the program can be examined. A specification test is conducted to check whether the specifications stating the program are performing under various conditions. Apart from these tests, there are some special tests conducted which are given below:

Peak Load Tests: This determines whether the new system will handle the volume of activities when the system is at the peak of its processing demand. The test has revealed that the new software for the agency is capable of handling the demands at the peak time.

Storage Testing: This determines the capacity of the new system to store transaction data on a disk or on other files. The proposed software has the required storage space available, because of the use of a number of hard disks.

Performance Time Testing: This test determines the length of the time used by the system to process transaction data.

**CHAPTER 6**

**6. RESULTS AND DISCUSSIONS**

In business, there are situations were one get cheated by the people they trust. One way to safeguard their properties is to rely on technology. This project enables a person to have accurate results and to maintain each and every details securely. By this each penny that goes out and comes in are recorded and enables a vendor to have a trust worthy environment. The reports generated gives an outlook of a shop’s status and allows them to improve at tragic times. By automating by technology a lot of man power is reduced and allows them to concentrate on enhancing their economic growth of the company. Improving technology and implementing them can result in concise and prosperity in work environment.

**CHAPTER 7**

**7. CONCLUSION AND FUTURE WORK**

In this project, we addressed the problem of erroneous calculations in purchase and sales account of a retail store. One problem is the existing work was, human errors and accurate results cannot be computed due to lack of knowledge at some unintentional cases. Thus our work provided an optimal solution that can provide a platform to proceed calculations accurately which was integrated in a desktop based application. When this system proves to be flexible to the users and achieves to be user friendly then it can go through a series of advance steps that might be handy to use for all the users. The future work may comprise of paying bill through online transactions and paying taxes being automated. If more numbers are benefited, then the details can be stored in cloud and lead to peer-to-peer interactions. This system can be modified in a further for audit computation and online transactions can be added if it is to be implemented as web application. Many different adaptations, test and experiments have been left for the future due to lack of time. Future work concerns deeper analysis of particular mechanism, new proposals to try different methods or simply curiosity. For further better improvements it relies on user’s needs and technical advisor’s optimized level of thinking.

**CHAPTER 8**

**8. APPENDICES**

**8.1 APPENDIX -1 CODING**

**8.1.1 Sales**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

using System.Data.OleDb;

using System.IO;

using iTextSharp.text;

using iTextSharp.text.pdf;

namespace AmuruthAgencies

{

public partial class Sales : Form

{

OleDbConnection conn;

decimal amt, tot;

public Sales()

{

try

{

InitializeComponent();

conn = new OleDbConnection();

conn.ConnectionString = @"Provider=Microsoft.ACE.OLEDB.12.0;DataSource=AmuruthDB.accdb";

conn.Open();

conn.Close();

}

catch (Exception ex)

{

MessageBox.Show("Error : " + ex);

}

}

private void Sales\_Load(object sender, EventArgs e)

{

try

{

this.dataGridView1.Hide();

button\_View.Hide();

dataGridView2.Hide();

}

catch (Exception ex)

{

MessageBox.Show("Error : " + ex);

}

}

private void radioButton1\_Salesman\_CheckedChanged(object sender, EventArgs e)

{

try

{

this.dataGridView1.Hide();

groupBox\_salesman.Show();

groupBox\_Dept.Hide();

textBox2\_Paid.Text = String.Empty;

textBox\_Today\_Credit.Text = String.Empty;

textBox\_Old\_Credit.Text = String.Empty;

textBox\_Total\_Credit.Text = String.Empty;

button\_View.Hide();

dataGridView2.Hide();

comboBox1.DataSource = null;

comboBox1.Items.Clear();

comboBox1.DropDownHeight = comboBox1.ItemHeight;

if (radioButton1\_Salesman.Checked)

{

conn.Open();

OleDbCommand cmd = new OleDbCommand();

cmd.Connection = conn;

string qry = "select \* from SALESMAN\_DETAILS ";

cmd.CommandText = qry;

OleDbDataReader reader = cmd.ExecuteReader();

while (reader.Read())

{

comboBox1.Items.Add(reader["SALESMAN\_NAME"].ToString());

comboBox1.DropDownHeight = comboBox1.ItemHeight \* (comboBox1.Items.Count + 1);

}

comboBox1.Focus();

conn.Close();

}

}

catch (Exception ex)

{

MessageBox.Show("Error : " + ex);

}

}

private void radioButton2\_Dept\_CheckedChanged(object sender, EventArgs e)

{

try

{

this.dataGridView1.Hide();

groupBox\_Dept.Show();

groupBox\_salesman.Hide();

textBox2\_Paid.Text = String.Empty;

textBox\_Today\_Credit.Text = String.Empty;

textBox\_Old\_Credit.Text = String.Empty;

textBox\_Total\_Credit.Text = String.Empty;

button\_View.Hide();

dataGridView2.Hide();

comboBox1.DataSource = null;

comboBox1.Items.Clear();

comboBox1.DropDownHeight = comboBox1.ItemHeight;

textBox\_add.Multiline = true;

textBox\_add.Height = 50;

textBox\_add1.Multiline = true;

textBox\_add1.Height = 50;

string query;

OleDbDataReader reader;

OleDbCommand cmd;

conn.Open();

query = "select \* from DEALERS\_DETAILS where DEALER\_NAME = 'AMRUTH AGENCIES'";

cmd = new OleDbCommand(query, conn);

reader = cmd.ExecuteReader();

if (reader.HasRows)

{

reader.Read();

textBox\_from.Text = reader.GetString(0);

textBox\_GST.Text = reader.GetString(2);

textBox\_add.Text = reader.GetString(1);

}

if (radioButton2\_Dept.Checked)

{

cmd = new OleDbCommand();

cmd.Connection = conn;

string qry = "select \* from DEPARTMENT\_DETAILS ";

cmd.CommandText = qry;

reader = cmd.ExecuteReader();

while (reader.Read())

{

comboBox\_to.Items.Add(reader["DEPARTMENT\_NAME"].ToString());

comboBox\_to.DropDownHeight = comboBox\_to.ItemHeight \* (comboBox\_to.Items.Count + 1);

}

comboBox1.Focus();

}

conn.Close();

}

catch (Exception ex)

{

MessageBox.Show("Error : " + ex);

}

}

private void comboBox1\_SelectedIndexChanged(object sender, EventArgs e)

{

try

{

tot = 0;

textBox1\_total.Text = Convert.ToString(tot);

textBox2\_Paid.Text = String.Empty;

textBox\_Today\_Credit.Text = String.Empty;

textBox\_Old\_Credit.Text = String.Empty;

textBox\_Total\_Credit.Text = String.Empty;

button\_View.Hide();

dataGridView2.Hide();

OleDbDataAdapter da;

DataSet ds;

string query = "";

if (radioButton1\_Salesman.Checked)

{

this.dataGridView1.Show();

string selected\_name = comboBox1.SelectedItem.ToString();

dataGridView1.AllowUserToAddRows = false;

dataGridView1.AutoGenerateColumns = false;

conn.Open();

query = @"SELECT PRODUCT\_DETAILS.ID, PRODUCT\_DETAILS.Product\_Description, PRODUCT\_DETAILS.HSN, PRODUCT\_DETAILS.GST, PRODUCT\_DETAILS.MRP, SALESMAN\_PRODUCTS.RATE,SALESMAN\_DETAILS.SALESMAN\_NAME FROM PRODUCT\_DETAILS INNER JOIN (SALESMAN\_DETAILS INNER JOIN SALESMAN\_PRODUCTS ON SALESMAN\_DETAILS.SALESMAN\_ID = SALESMAN\_PRODUCTS.SALESMAN\_ID) ON PRODUCT\_DETAILS.Product\_Description = SALESMAN\_PRODUCTS.PRODUCT\_DESCRIPTION WHERE (((SALESMAN\_DETAILS.SALESMAN\_NAME) In ('" + selected\_name + "')))";

}

da = new OleDbDataAdapter(query, conn);

ds = new DataSet();

dataGridView1.DataSource = null;

ds.Clear();

da.Fill(ds);

DataGridViewCheckBoxColumn chk = new DataGridViewCheckBoxColumn();

dataGridView1.Columns.Add(chk);

chk.HeaderText = "Select The Item";

chk.Name = "chk";

dataGridView1.ColumnCount = 10;

dataGridView1.Columns[1].Name = "SNo";

dataGridView1.Columns[1].HeaderText = "SNO";

dataGridView1.Columns[1].ReadOnly = true;

dataGridView1.Columns[2].Name = "Product\_ID";

dataGridView1.Columns[2].HeaderText = "Product ID";

dataGridView1.Columns[2].DataPropertyName = "ID";

dataGridView1.Columns[2].ReadOnly = true;

dataGridView1.Columns[3].HeaderText = "Product Description";

dataGridView1.Columns[3].Name = "Name";

dataGridView1.Columns[3].Width = 150;

dataGridView1.Columns[3].DataPropertyName = "Product\_Description";

dataGridView1.Columns[3].ReadOnly = true;

dataGridView1.Columns[4].Name = "GST";

dataGridView1.Columns[4].HeaderText = "GST";

dataGridView1.Columns[4].DataPropertyName = "GST";

dataGridView1.Columns[4].ReadOnly = true;

dataGridView1.Columns[5].Name = "HSN";

dataGridView1.Columns[5].HeaderText = "HSN";

dataGridView1.Columns[5].DataPropertyName = "HSN";

dataGridView1.Columns[5].ReadOnly = true;

dataGridView1.Columns[6].Name = "MRP";

dataGridView1.Columns[6].HeaderText = "MRP";

dataGridView1.Columns[6].DataPropertyName = "MRP";

dataGridView1.Columns[6].ReadOnly = true;

dataGridView1.Columns[7].Name = "Quantity";

dataGridView1.Columns[7].HeaderText = "Quantity";

dataGridView1.Columns[7].ReadOnly = true;

dataGridView1.Columns[8].Name = "Rate";

dataGridView1.Columns[8].HeaderText = "Rate";

dataGridView1.Columns[8].DataPropertyName = "RATE";

dataGridView1.Columns[8].ReadOnly = true;

dataGridView1.Columns[9].Name = "Amount";

dataGridView1.Columns[9].HeaderText = "Amount";

dataGridView1.Columns[9].ReadOnly = true;

dataGridView1.DataSource = ds.Tables[0];//Set AutoGenerateColumns False

conn.Close();

}

catch (Exception ex)

{

MessageBox.Show("Error : " + ex);

}

}

private void dataGridView1\_RowPostPaint\_1(object sender, DataGridViewRowPostPaintEventArgs e)

{

this.dataGridView1.Rows[e.RowIndex].Cells["SNo"].Value = (e.RowIndex + 1).ToString();

}

private void dataGridView1\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

try

{

if (radioButton2\_Dept.Checked == true)

{

if (e.ColumnIndex == dataGridView1.Columns["chk"].Index)

{

if (Convert.ToBoolean(dataGridView1.Rows[e.RowIndex].Cells["chk"].Value) == false)

{

dataGridView1.Rows[e.RowIndex].Cells["Quantity"].ReadOnly = false;

}

else

{

dataGridView1.Rows[e.RowIndex].Cells["Quantity"].ReadOnly = true;

}

decimal GST;

int GSTcount = 0;

GST = Convert.ToDecimal(dataGridView1.Rows[e.RowIndex].Cells["GST"].Value) / 2;

foreach (DataGridViewRow row3 in dataGridView3.Rows)

{

if (Convert.ToDecimal(row3.Cells["GST\_CGST"].Value) == GST

}

}

catch (Exception ex)

{

MessageBox.Show("Error : " + ex);

}

}

private void dataGridView1\_CellEnter(object sender, DataGridViewCellEventArgs e)

{

try

{

bool chked = Convert.ToBoolean(dataGridView1.Rows[e.RowIndex].Cells["chk"].Value);

int amt\_index = dataGridView1.Columns["Amount"].Index;

int qty\_index = dataGridView1.Columns["Quantity"].Index;

int rate\_index = dataGridView1.Columns["Rate"].Index;

if (e.ColumnIndex == amt\_index && chked == true)

{

if (dataGridView1.Rows[e.RowIndex].Cells[qty\_index].EditedFormattedValue.ToString() == "" || dataGridView1.Rows[e.RowIndex].Cells[rate\_index].EditedFormattedValue.ToString() == "")

{

MessageBox.Show("Enter the value");

}

else

{

decimal qty = Convert.ToDecimal(dataGridView1.Rows[e.RowIndex].Cells[qty\_index].Value.ToString());

decimal rate = Convert.ToDecimal(dataGridView1.Rows[e.RowIndex].Cells[rate\_index].Value.ToString());

amt = qty \* rate;

tot = tot + amt;

dataGridView1.Rows[e.RowIndex].Cells[amt\_index].Value = amt;

}

}

}

catch (Exception ex)

{

MessageBox. Show("Error : " + ex);

}

}

private void button1\_save\_Click(object sender, EventArgs e)

{

try

{

if (radioButton1\_Salesman.Checked)

{

string qry;

OleDbCommand cmd;

qry = "select SALESMAN\_ID from SALESMAN\_DETAILS where SALESMAN\_NAME = '" + comboBox1.SelectedItem.ToString() + "'";

cmd = new OleDbCommand();

conn.Open();

cmd.Connection = conn;

cmd.CommandText = qry;

Int32 salesman\_id = (Int32)cmd.ExecuteScalar();

conn.Close();

int count, count1;

foreach (DataGridViewRow row in dataGridView1.Rows)

{

if (Convert.ToBoolean(row.Cells["chk"].Value) == true)

{

var dte = DateTime.Today.ToShortDateString();

string pro\_desc = row.Cells["Name"].Value.ToString();

Int32 qnty = Convert.ToInt32(row.Cells["Quantity"].Value);

decimal Rate = Convert.ToDecimal(row.Cells["Rate"].Value);

decimal amount = Convert.ToDecimal(row.Cells["Amount"].Value);

conn.Open();

string qry\_chk = @"select COUNT(\*) from SALESMAN\_PRODUCT\_ENTRY where DATE\_DAILY = #" + dte + "# and SALESMAN\_ID = " + salesman\_id + " and PRODUCT\_DESCRIPTION = '" + pro\_desc + "'";

OleDbCommand cmd\_chk = conn.CreateCommand();

cmd\_chk.CommandText = qry\_chk;

count = (int)cmd\_chk.ExecuteScalar();

conn.Close();

string qry1;

if (count >= 1)

{

conn.Open();

qry1 = @"update SALESMAN\_PRODUCT\_ENTRY set QUANTITY = '" + qnty + "', RATE = '" + Rate + "', AMOUNT = '" + amount + "' where DATE\_DAILY = #" + dte + "# and SALESMAN\_ID = " + salesman\_id + " and PRODUCT\_DESCRIPTION = '" + pro\_desc + "'";

OleDbCommand cmd1 = new OleDbCommand(qry1, conn);

cmd1.ExecuteNonQuery();

conn.Close();

}

else

{

conn.Open();

qry1 = @"insert into SALESMAN\_PRODUCT\_ENTRY values ('" + dte + "','" + salesman\_id + "','" + pro\_desc + "','" + qnty + "','" + Rate + "','" + amount + "')";

OleDbCommand cmd1 = new OleDbCommand(qry1, conn);

cmd1.ExecuteNonQuery();

conn.Close();

}

conn.Open();

qry = "select STOCK from STOCK\_DETAILS where PRODUCT\_DESCRIPTION = '" + pro\_desc + "'";

cmd = new OleDbCommand(qry, conn);

int stock = (int)cmd.ExecuteScalar();

conn.Close();

conn.Open();

qry = "update STOCK\_DETAILS set STOCK = " + (stock - qnty) + " where PRODUCT\_DESCRIPTION = '" + pro\_desc + "'";

cmd = new OleDbCommand(qry, conn);

cmd.ExecuteNonQuery();

conn.Close();

conn.Open();

decimal credits = Convert.ToDecimal(textBox\_Total\_Credit.Text);

qry1 = @"update SALESMAN\_DETAILS set CREDITS = '" + credits + "' where SALESMAN\_ID = " + salesman\_id;

OleDbCommand cmd\_cre = new OleDbCommand(qry1, conn);

cmd\_cre.ExecuteNonQuery();

conn.Close();

}

}

foreach (DataGridViewRow row in dataGridView2.Rows)

{

if (Convert.ToBoolean(row.Cells["Paid\_or\_Not"].Value) == true)

{

conn.Open();

qry = @"update SALESMAN\_CREDITS set PAID\_OR\_NOT = " + true + ", PAID\_DATE = #" + row.Cells["Paid\_Date"].Value + "# where ID = " + Convert.ToInt32(row.Cells["ID"].Value) + " and STOCK\_DATE = #" + row.Cells["Stock\_Date"].Value + "#";

MessageBox.Show(qry);

cmd = new OleDbCommand(qry, conn);

cmd.ExecuteNonQuery();

conn.Close();

}

}

conn.Open();

qry = @"select COUNT(\*) from SALESMAN\_CREDITS where STOCK\_DATE = #" + dateTimePicker1.Value.Date.ToShortDateString() + "# and ID = " + salesman\_id ;

cmd=new OleDbCommand(qry,conn);

count1 = (int)cmd.ExecuteScalar();

if (count1 == 0)

{

qry = @"insert into SALESMAN\_CREDITS (ID, STOCK\_DATE, CREDITS) values (" + salesman\_id + ", #" + dateTimePicker1.Value.Date.ToShortDateString() + "# , " + Convert.ToDecimal(textBox\_Today\_Credit.Text) + ")";

cmd = new OleDbCommand(qry, conn);

cmd.ExecuteNonQuery();

}

doc.Open();

Paragraph p11 = new Paragraph(dateTimePicker1.Value.Date.ToShortDateString(),font1);

doc.Add(p11);

doc.Add(new Phrase("Salesman Name : "+this.comboBox1.SelectedItem.ToString(),font2));

PdfPTable table;

table = new PdfPTable(dataGridView1.Columns.Count);

table.HorizontalAlignment = 1;

table.TotalWidth = 700f;

table.LockedWidth = true;

float[] widths = new float[] { 50f, 30f, 70f, 130f, 40f, 100f, 70f, 70f, 70f, 70f };

table.SetWidths(widths);

for (int i = 0; i < dataGridView1.Columns.Count; i++)

{

table.AddCell(new Phrase(dataGridView1.Columns[i].HeaderText, font2));

}

table.HeaderRows = 1;

string s = "";

for (int i = 0; i < dataGridView1.Rows.Count; i++)

{

for (int j = 0; j < dataGridView1.Columns.Count; j++)

{

if (dataGridView1[j, i].Value != null)

{

s = dataGridView1[j, i].Value.ToString();

table.AddCell(new Phrase(dataGridView1[j, i].Value.ToString(), font3));

}

else

{

table.AddCell("");

}

}

}

doc.Add(table);

doc.Close();

MessageBox.Show("FILE SAVED");

}

else if (radioButton2\_Dept.Checked == true)

{

string qry;

OleDbCommand cmd;

qry = "select DEPARTMENT\_ID from DEPARTMENT\_DETAILS where DEPARTMENT\_NAME = '" + comboBox\_to.SelectedItem.ToString() + "'";

conn.Open();

Convert.ToDecimal(textBox\_Today\_Credit.Text) + " where STOCK\_DATE = #" + dateTimePicker1.Value.Date.ToShortDateString() + "# and ID = " + dept\_id;

cmd = new OleDbCommand(qry, conn);

cmd.ExecuteNonQuery();

}

conn.Close();

iTextSharp.text.Font font1 = FontFactory.GetFont("Arial", 20, iTextSharp.text.Font.NORMAL, BaseColor.RED);

iTextSharp.text.Font heading = FontFactory.GetFont("Times New Roman", 12, iTextSharp.text.Font.BOLD, BaseColor.BLACK);

iTextSharp.text.Font content = FontFactory.GetFont("Times New Roman", 10, iTextSharp.text.Font.NORMAL, BaseColor.BLACK);

Paragraph newline = new Paragraph("\n");

Document doc = new Document(iTextSharp.text.PageSize.A4, 30, 30, 30, 30);

string pdf\_File = @"d:\Users\admin1\Desktop\INVOICE DETAILS\Invoice " + textBox\_Invoice.Text + ".pdf";

PdfWriter wri = PdfWriter.GetInstance(doc, new FileStream(pdf\_File, FileMode.Create));

doc.Open();

Paragraph p11 = new Paragraph("Invoice No : " + textBox\_Invoice.Text, font1);

doc.Add(p11);

doc.Add(newline); doc.Add(newline);

float[] width = { 1, 5, 1, 5 };

PdfPTable top\_table = new PdfPTable(width);

top\_table.WidthPercentage = 100F;

top\_table.HorizontalAlignment = 1;

top\_table.DefaultCell.Padding = 4;

PdfPCell pcell;

pcell = new PdfPCell(new Phrase("INVOICE DATE : " + dateTimePicker1.Value.Date.ToShortDateString(), heading));

pcell.Colspan = 4;

pcell.HorizontalAlignment = 1;

top\_table.AddCell(pcell);

top\_table.AddCell(new Phrase("FROM", heading));

top\_table.AddCell(new Phrase(textBox\_from.Text + "\nGST NO : " + textBox\_GST.Text + "\n" + textBox\_add.Text, content));

top\_table.AddCell(new Phrase("BILL TO", heading));

top\_table.AddCell(new Phrase(comboBox\_to.SelectedItem.ToString() + "\nGST NO : " + textBox\_GST1.Text + "\n" + textBox\_add1.Text, content));

doc.Add(top\_table);

doc.Add(newline);

PdfPTable table;

float[] wid = { 1, 4, 1, 2, 2, 2, 2, 2 };

table = new PdfPTable(wid);

table.WidthPercentage = 100F;

table.HorizontalAlignment = 1;

table.DefaultCell.Padding = 4;

for (int i = 1; i < dataGridView1.Columns.Count; i++)

{

if (i != 2)

{

table.AddCell(new Phrase(dataGridView1.Columns[i].HeaderText, heading));

}

}

table.HeaderRows = 1;

string s = "";

int sno = 1;

for (int i = 0; i < dataGridView1.Rows.Count; i++)

{

if (Convert.ToBoolean(dataGridView1.Rows[i].Cells["chk"].Value) == false)

{

}

else

{

table.AddCell(Convert.ToString(sno++));

for (int j = 3; j < dataGridView1.Columns.Count; j++)

{

if (dataGridView1[j, i].Value != null)

{

s = dataGridView1[j, i].Value.ToString();

table.AddCell(new Phrase(dataGridView1[j, i].Value.ToString(), content));

}

else

{

table.AddCell("");

}

}

}

}

doc.Add(table);

doc.Add(newline); doc.Add(newline);

Paragraph gst = new Paragraph(new Phrase("GST CALCULATION", heading));

gst.Alignment = 2;

doc.Add(gst);

doc.Add(newline);

float[] gstWidth = { 2, 2, 2, 2, 2 };

table = new PdfPTable(gstWidth);

table.WidthPercentage = 100F;

table.HorizontalAlignment = 2;

table.DefaultCell.Padding = 4;

for (int i = 0; i < dataGridView3.Columns.Count; i++)

{

if (i != 2)

{

table.AddCell(new Phrase(dataGridView3.Columns[i].HeaderText,heading));

}

}

table.HeaderRows = 1;

for (int i = 0; i < dataGridView3.Rows.Count; i++)

{

for (int j = 0; j < dataGridView3.Columns.Count; j++)

{

if (j != 2)

{

if (dataGridView3[j, i].Value != null)

{

table.AddCell(new Phrase(dataGridView3[j, i].Value.ToString(), content));

}

else

{

table.AddCell("");

}

textBox\_Today\_Credit.Text = String.Empty;

textBox\_Old\_Credit.Text = String.Empty;

textBox\_Total\_Credit.Text = String.Empty;

button\_View.Hide();

dataGridView2.Hide();

OleDbDataAdapter da;

OleDbCommand cmd;

OleDbDataReader reader;

DataSet ds;

string query;

conn.Open();

this.dataGridView1.Show();

string selected\_name = comboBox\_to.SelectedItem.ToString();

query = "select \* from DEALERS\_DETAILS where DEALER\_NAME = '" + selected\_name + "'";

cmd = new OleDbCommand(query, conn);

reader = cmd.ExecuteReader();

if (reader.HasRows)

{

reader.Read();

textBox\_GST1.Text = reader.GetString(2);

textBox\_add1.Text = reader.GetString(1);

}

query = @"SELECT IIf(Max(INVOICE\_DETAILS.INV\_NO) Is Null,0+1,Max(INVOICE\_DETAILS.INV\_NO)+1) FROM INVOICE\_DETAILS";

cmd = new OleDbCommand(query, conn);

int invoice = (int)cmd.ExecuteScalar();

textBox\_Invoice.Text = Convert.ToString(invoice);

dataGridView1.AllowUserToAddRows = false;

dataGridView1.AutoGenerateColumns = false;

query = @"SELECT PRODUCT\_DETAILS.ID, PRODUCT\_DETAILS.Product\_Description, PRODUCT\_DETAILS.HSN, PRODUCT\_DETAILS.MRP, PRODUCT\_DETAILS.GST, DEPARTMENT\_DETAILS.DEPARTMENT\_NAME,DEPARTMENT\_RPODUCTS.RATE FROM PRODUCT\_DETAILS INNER JOIN (DEPARTMENT\_DETAILS INNER JOIN DEPARTMENT\_RPODUCTS ON DEPARTMENT\_DETAILS.DEPARTMENT\_ID = DEPARTMENT\_RPODUCTS.DEPARTMENT\_ID) ON PRODUCT\_DETAILS.Product\_Description = DEPARTMENT\_RPODUCTS.PRODUCT\_DESCRIPTION WHERE (((DEPARTMENT\_DETAILS.DEPARTMENT\_NAME) In ('" + selected\_name + "')))";

da = new OleDbDataAdapter(query, conn);

ds = new DataSet();

dataGridView1.DataSource = null;

ds.Clear();

da.Fill(ds);

DataGridViewCheckBoxColumn chk = new DataGridViewCheckBoxColumn();

dataGridView1.Columns.Add(chk);

chk.HeaderText = "Select The Item";

chk.Name = "chk";

dataGridView1.ColumnCount = 10;

dataGridView1.Columns[1].Name = "SNo";

dataGridView1.Columns[1].HeaderText = "SNO";

dataGridView1.Columns[1].ReadOnly = true;

dataGridView1.Columns[2].Name = "Product\_ID";

dataGridView1.Columns[2].HeaderText = "Product ID";

dataGridView1.Columns[2].DataPropertyName = "ID";

dataGridView1.Columns[2].ReadOnly = true;

dataGridView1.Columns[3].HeaderText = "Product Description";

dataGridView1.Columns[3].Name = "Name";

if (radioButton2\_Dept.Checked)

{

today\_cre = Convert.ToDecimal(textBox1\_total.Text) - Convert.ToDecimal(textBox2\_Paid.Text);

textBox\_Today\_Credit.Text = Convert.ToString(today\_cre);

qry = @"select CREDITS from DEPARTMENT\_DETAILS where DEPARTMENT\_NAME = '" + comboBox\_to.SelectedItem.ToString() + "'";

cmd = new OleDbCommand(qry, conn);

old\_cre = (decimal)cmd.ExecuteScalar();

textBox\_Old\_Credit.Text = Convert.ToString(old\_cre);

conn.Close();

tot\_cre = today\_cre + old\_cre;

textBox\_Total\_Credit.Text = Convert.ToString(tot\_cre);

}

}

catch (Exception ex)

{

MessageBox.Show("Error : " + ex);

}

}

private void button\_View\_Click(object sender, EventArgs e)

{

try

{

string qry;

OleDbDataAdapter da;

DataSet ds;

dataGridView2.Show();

if (radioButton1\_Salesman.Checked)

{

conn.Open();

qry = "select \* from SALESMAN\_CREDITS where ID = (select SALESMAN\_ID from SALESMAN\_DETAILS where SALESMAN\_NAME = '" + comboBox1.SelectedItem.ToString() + "') and PAID\_OR\_NOT = " + false;

da = new OleDbDataAdapter(qry, conn);

ds = new DataSet();

dataGridView2.DataSource = null;

ds.Clear();

da.Fill(ds);

dataGridView2.AllowUserToAddRows = false;

dataGridView2.AutoGenerateColumns = false;

dataGridView2.ColumnCount = 5;

dataGridView2.Columns[0].Name = "ID";

dataGridView2.Columns[0].HeaderText = "ID";

dataGridView2.Columns[0].DataPropertyName = "ID";

dataGridView2.Columns[0].ReadOnly = true;

dataGridView2.Columns[1].Name = "Stock\_Date";

dataGridView2.Columns[1].HeaderText = "STOCK\_DATE";

dataGridView2.Columns[1].DataPropertyName = "STOCK\_DATE";

dataGridView2.Columns[1].ReadOnly = true;

dataGridView2.Columns[2].Name = "Credits";

dataGridView2.Columns[2].HeaderText = "CREDITS";

dataGridView2.Columns[2].DataPropertyName = "CREDITS";

dataGridView2.Columns[2].ReadOnly = true;

DataGridViewCheckBoxColumn chk\_Column = new DataGridViewCheckBoxColumn();

chk\_Column.DataPropertyName = "PAID\_OR\_NOT";

chk\_Column.HeaderText = "PAID-OR\_NOT";

chk\_Column.Name = "Paid\_or\_Not";

dataGridView2.Columns.Add(chk\_Column);

dataGridView2.Columns[4].Name = "Paid\_Date";

dataGridView2.Columns[4].HeaderText = "PAID\_DATE";

dataGridView2.Columns[4].DataPropertyName = "PAID\_DATE";

dataGridView2.Columns[4].ReadOnly = true;

DataGridViewButtonColumn btn\_col = new DataGridViewButtonColumn();

btn\_col.HeaderText = "PAID";

btn\_col.Text = "PAID";

btn\_col.Name = "paid";

btn\_col.UseColumnTextForButtonValue = true;

dataGridView2.Columns.Add(btn\_col);

dataGridView2.DataSource = ds.Tables[0];//Set AutoGenerateColumns False

conn.Close();

}

if (radioButton2\_Dept.Checked)

{

conn.Open();

qry = "select \* from DEPARTMENT\_CREDITS where ID = (select DEPARTMENT\_ID from DEPARTMENT\_DETAILS where DEPARTMENT\_NAME = '" + comboBox\_to.SelectedItem.ToString() + "') and PAID\_OR\_NOT = " + false;

da = new OleDbDataAdapter(qry, conn);

ds = new DataSet();

dataGridView2.DataSource = null;

ds.Clear();

da.Fill(ds);

dataGridView2.AllowUserToAddRows = false;

dataGridView2.AutoGenerateColumns = false;

dataGridView2.ColumnCount = 5;

dataGridView2.Columns[0].Name = "ID";

dataGridView2.Columns[0].HeaderText = "ID";

dataGridView2.Columns[0].DataPropertyName = "ID";

}

}

catch (Exception ex)

{

MessageBox.Show("Error : " + ex);

}

}

private void dataGridView2\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

try

{

decimal old\_credit,tot\_credit;

if (e.ColumnIndex == dataGridView2.Columns["paid"].Index)

{

dataGridView2.Rows[e.RowIndex].Cells["paid\_or\_not"].Value = true;

var dte = DateTime.Today.ToShortDateString();

dataGridView2.Rows[e.RowIndex].Cells["Paid\_Date"].Value = dte;

old\_credit = Convert.ToDecimal(textBox\_Old\_Credit.Text) - Convert.ToDecimal(dataGridView2.Rows[e.RowIndex].Cells["Credits"].Value);

tot\_credit = Convert.ToDecimal(textBox\_Today\_Credit.Text) + old\_credit;

textBox\_Old\_Credit.Text = Convert.ToString(old\_credit);

textBox\_Total\_Credit.Text = Convert.ToString(tot\_credit);

}

}

catch (Exception ex)

{

MessageBox.Show("Error : " + ex);

}

}

**8.1.2 Purchase**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

using System.Data.OleDb;

namespace AmuruthAgencies

{

public partial class Purchase\_Entry : Form

{

decimal amt,tot;

OleDbConnection conn;

public Purchase\_Entry()

{

InitializeComponent();

conn = new OleDbConnection();

conn.ConnectionString = @"Provider=Microsoft.ACE.OLEDB.12.0;Data Source=AmuruthDB.accdb";

conn.Open();

conn.Close();

}

private void Purchase\_Entry\_Load(object sender, EventArgs e)

{

OleDbCommand cmd;

string qry;

OleDbDataReader reader;

textBox1\_To.Multiline = true;

textBox1\_To.Height = 30;

textBox\_Address.Multiline = true;

textBox\_Address.Height = 50;

textBox\_Address1.Multiline = true;

textBox\_Address1.Height = 50;

try

{

conn.Open();

conn.Close();

this.dataGridView1.Hide();

conn.Open();

cmd = new OleDbCommand();

cmd.Connection = conn;

qry = "select \* from CATEGORY\_DETAILS ";

cmd.CommandText = qry;

reader = cmd.ExecuteReader();

while (reader.Read())

{

comboBox1.Items.Add(reader["CATEGORY"].ToString());

}

comboBox1.Focus();

qry = "select \* from DEALERS\_DETAILS where DEALER\_NAME = 'MILKY MIST DAIRY FOOD PVT LTD'";

cmd = new OleDbCommand(qry, conn);

reader = cmd.ExecuteReader();

if (reader.HasRows)

{

reader.Read();

textBox\_from.Text = reader.GetString();

conn.Open();

string query = "select \* from PRODUCT\_DETAILS p where p.CATEGORY =(select c.CATEGORY\_ID from CATEGORY\_DETAILS c where c.CATEGORY = '" + selected\_category + "')";

OleDbDataAdapter da = new OleDbDataAdapter(query, conn);

DataSet ds = new DataSet();

dataGridView2.DataSource = null;

ds.Clear();

da.Fill(ds);

DataGridViewCheckBoxColumn chk = new DataGridViewCheckBoxColumn();

dataGridView2.Columns.Add(chk);

chk.HeaderText = "Select The Item";

chk.Name = "chk";

dataGridView2.ColumnCount = 3;

dataGridView2.Columns[2].Name = "Product\_ID";

dataGridView2.Columns[2].HeaderText = "Product ID";

dataGridView2.Columns[2].DataPropertyName = "ID";

dataGridView2.Columns[2].ReadOnly = true;

dataGridView2.Columns[1].HeaderText = "Product Description";

dataGridView2.Columns[1].Name = "Name";

dataGridView2.Columns[1].Width = 150;

dataGridView2.Columns[1].DataPropertyName = "Product\_Description";

dataGridView2.Columns[1].ReadOnly = true;

dataGridView2.DataSource = ds.Tables[0];

dataGridView1.ColumnCount = 9;

dataGridView1.Columns[0].Name = "SNo";

dataGridView1.Columns[0].HeaderText = "SNO";

dataGridView1.Columns[0].ReadOnly = true;

dataGridView1.Columns[1].Name = "Product\_ID";

dataGridView1.Columns[1].HeaderText = "Product ID";

dataGridView1.Columns[1].DataPropertyName = "ID";

private void dataGridView1\_RowPostPaint(object sender, DataGridViewRowPostPaintEventArgs e)

{

this.dataGridView1.Rows[e.RowIndex].Cells["SNo"].Value = (e.RowIndex + 1).ToString();

}

private void button1\_save\_Click(object sender, EventArgs e)

{

try

{

string qry;

OleDbCommand cmd;

foreach (DataGridViewRow row in dataGridView1.Rows)

{

var dte = DateTime.Today.ToString("dd-MMM-yy");

string pro\_desc = row.Cells["Name"].Value.ToString();

Int32 qnty = Convert.ToInt32(row.Cells["Quantity"].Value);

decimal mrp = Convert.ToDecimal(row.Cells["MRP"].Value);

decimal amount = Convert.ToDecimal(row.Cells["Amount"].Value);

qry = "INSERT INTO PURCHASE\_DETAILS values ('" + dte + "','" + pro\_desc + "','" + qnty + "','" + mrp + "','" + amount + "')";

cmd = new OleDbCommand();

conn.Open();

cmd.Connection = conn;

cmd.CommandText = qry;

cmd.Prepare();

cmd.ExecuteNonQuery();

conn.Close();

conn.Open();

qry = "select STOCK from STOCK\_DETAILS where PRODUCT\_DESCRIPTION = '"+pro\_desc+"'";

cmd = new OleDbCommand(qry, conn);

int stock = (int)cmd.ExecuteScalar();

conn.Close();

conn.Open();

qry = "update STOCK\_DETAILS set STOCK = " + (stock + qnty) + " where PRODUCT\_DESCRIPTION = '" + pro\_desc + "'";

cmd = new OleDbCommand(qry, conn);

cmd.ExecuteNonQuery();

conn.Close();

}

conn.Open();

string b = textBox\_from.Text;

string c = textBox1\_To.Text;

Int32 d = Convert.ToInt32(textBox\_Grand\_Total.Text);

qry = "insert into PURCHASE\_INVOICE\_DETAILS (INV\_NO, DATE\_DAILY, FROM\_SHOP, TO\_SHOP, TOTAL\_AMOUNT) values ('" + textBox\_Invoice.Text + "', #" + dateTimePicker1.Value.Date.ToShortDateString() + "#, '" + b + "','" + c + "'," + d + ")";

cmd = new OleDbCommand(qry, conn);

cmd.ExecuteNonQuery();

conn.Close();

MessageBox.Show("File Saved");

Purchase\_Entry pur = new Purchase\_Entry();

pur.Show();

this.Dispose(false);

}

catch (Exception ex)

{

}

private void dataGridView1\_CellEnter(object sender, DataGridViewCellEventArgs e)

{

try

{

int amt\_index = dataGridView1.Columns["Amount"].Index;

int qty\_index = dataGridView1.Columns["Quantity"].Index;

int rate\_index = dataGridView1.Columns["Rate"].Index;

if (e.ColumnIndex == amt\_index)

{

if (dataGridView1.Rows[e.RowIndex].Cells[qty\_index].EditedFormattedValue.ToString() == "" || dataGridView1.Rows[e.RowIndex].Cells[rate\_index].EditedFormattedValue.ToString() == "")

{

MessageBox.Show("Enter the value");

}

else

{

decimal qty = Convert.ToDecimal(dataGridView1.Rows[e.RowIndex].Cells[qty\_index].Value.ToString());

decimal rte = Convert.ToDecimal(dataGridView1.Rows[e.RowIndex].Cells[rate\_index].Value.ToString());

amt = qty \* rte;

tot = tot + amt;

dataGridView1.Rows[e.RowIndex].Cells[amt\_index].Value = amt;

textBox1\_total.Text = Convert.ToString(tot);

}

}

}

private void button\_calc\_GST\_Click(object sender, EventArgs e)

{

try

{

decimal amt,gst;

foreach (DataGridViewRow row in dataGridView1.Rows)

{

amt = Convert.ToDecimal(row.Cells["Amount"].Value);

gst = amt \* (Convert.ToDecimal(row.Cells["GST"].Value) / 100);

foreach (DataGridViewRow row1 in dataGridView3.Rows)

{

if ((Convert.ToDecimal(row1.Cells["GST\_CGST"].Value) \* 2) == Convert.ToDecimal(row.Cells["GST"].Value))

{

if (row1.Cells["Amount"].EditedFormattedValue.ToString() == "")

{

row1.Cells["Amount"].Value = Convert.ToString(gst);

row1.Cells["CGST\_Amount"].Value = Convert.ToString(Convert.ToDecimal(row1.Cells["Amount"].Value) / 2);

row1.Cells["SGST\_Amount"].Value = Convert.ToString(Convert.ToDecimal(row1.Cells["Amount"].Value) / 2);

}

else

{

row1.Cells["Amount"].Value = Convert.ToString(Convert.ToDecimal(row1.Cells["Amount"].Value) + gst);

row1.Cells["CGST\_Amount"].Value = Convert.ToString(Convert.ToDecimal(row1.Cells["Amount"].Value) / 2);

row1.Cells["SGST\_Amount"].Value = Convert.ToString(Convert.ToDecimal(row1.Cells["Amount"].Value) / 2);

}

}

}

}

decimal grand\_tot = 0;

foreach (DataGridViewRow row in dataGridView3.Rows)

{

grand\_tot = grand\_tot + Convert.ToDecimal(row.Cells["Amount"].Value);

}

grand\_tot = grand\_tot + Convert.ToDecimal(textBox1\_total.Text);

textBox\_Grand\_Total.Text = Convert.ToString(Convert.ToInt32(grand\_tot));

}

catch (Exception ex)

{

MessageBox.Show("Error : " + ex);

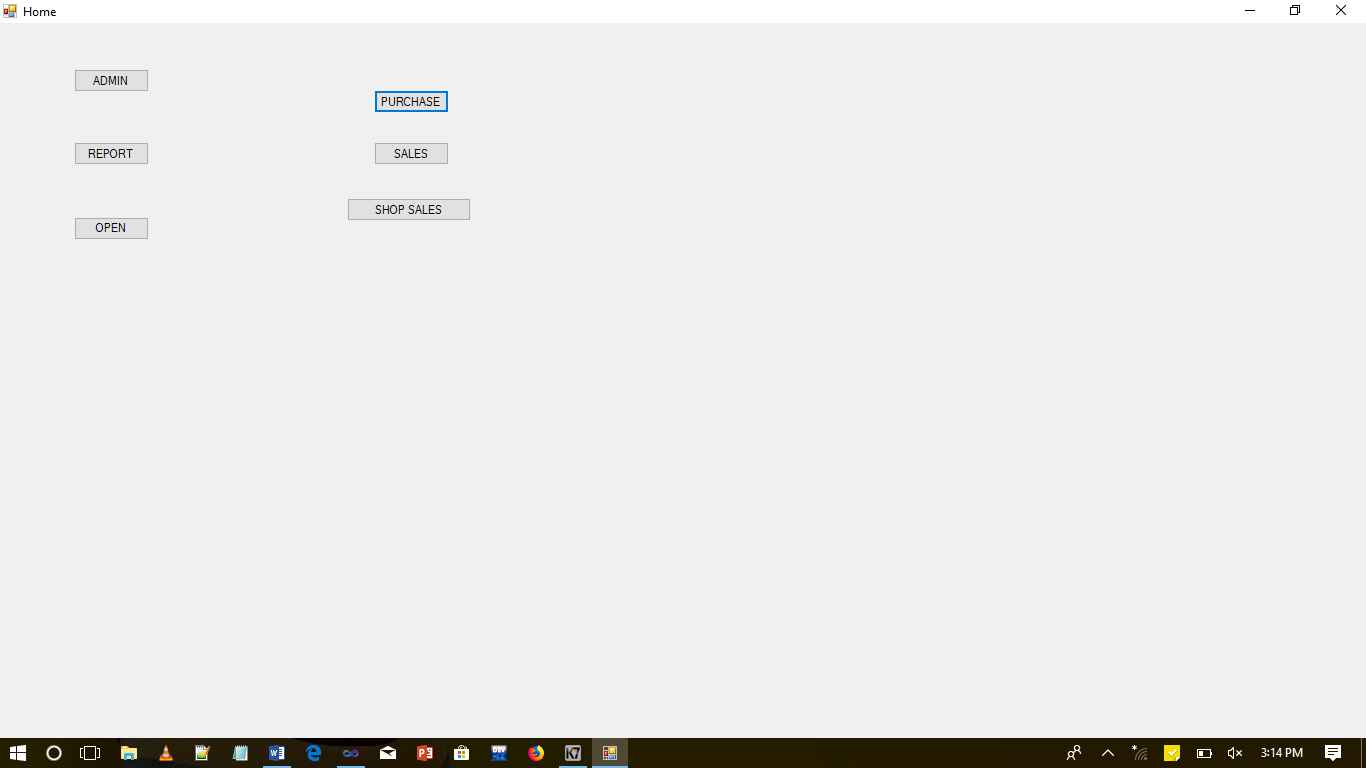
}

}

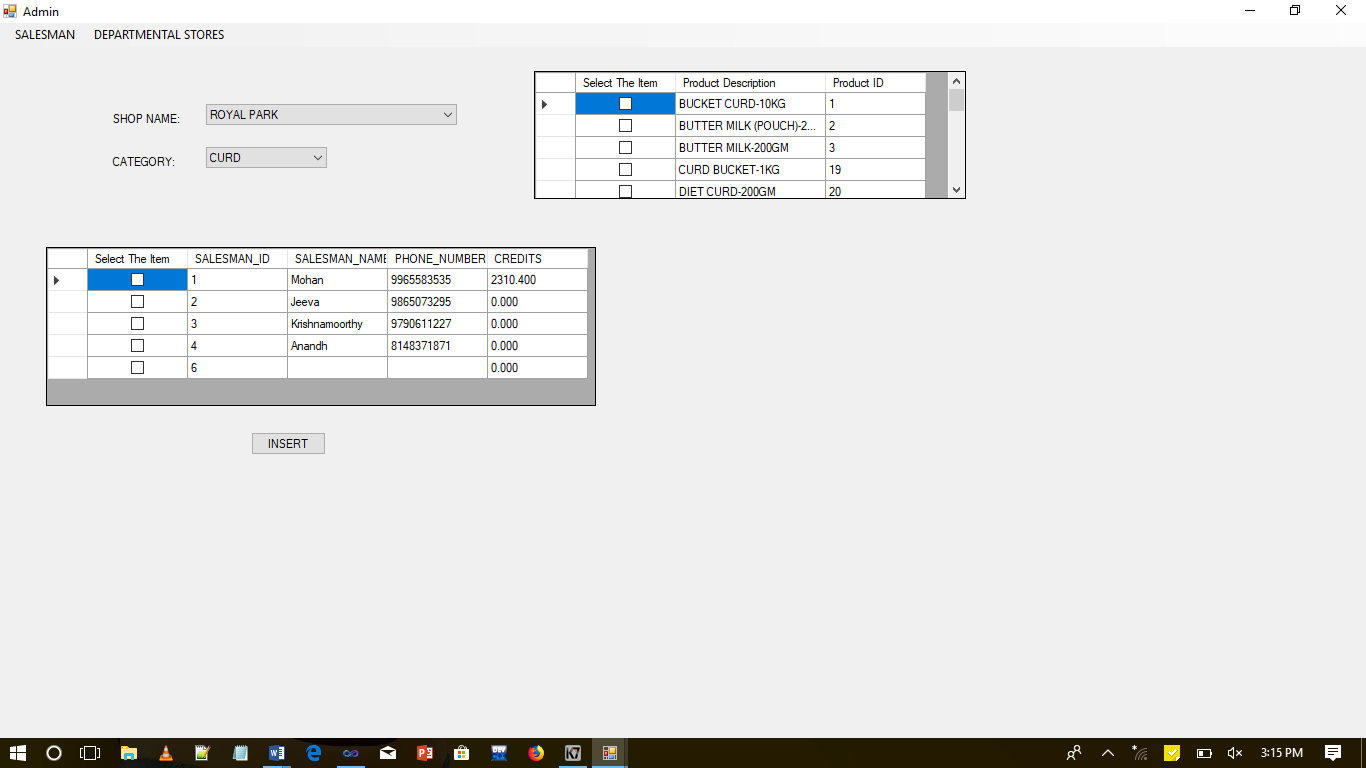
}

}

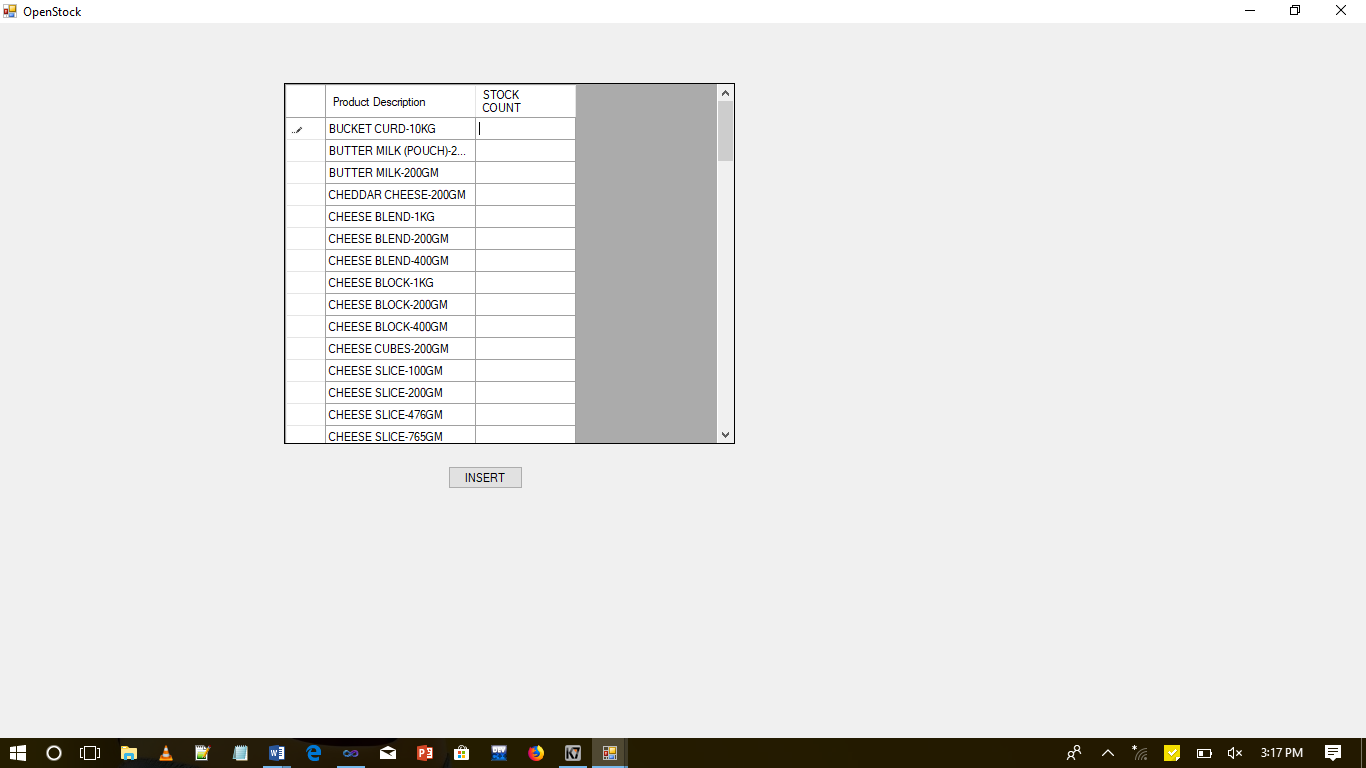
**8.2 APPENDIX – 2 SNAPSHOTS**



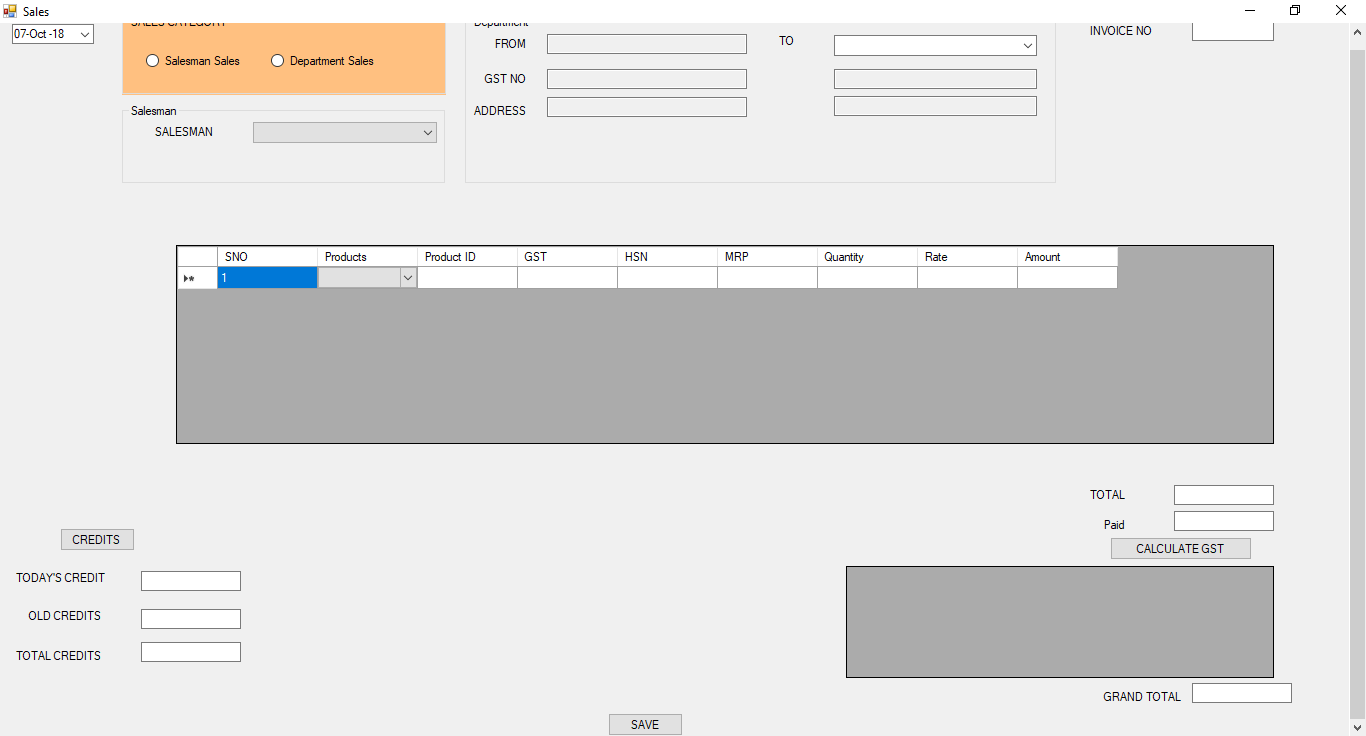
**Fig 8.1 : Index page**



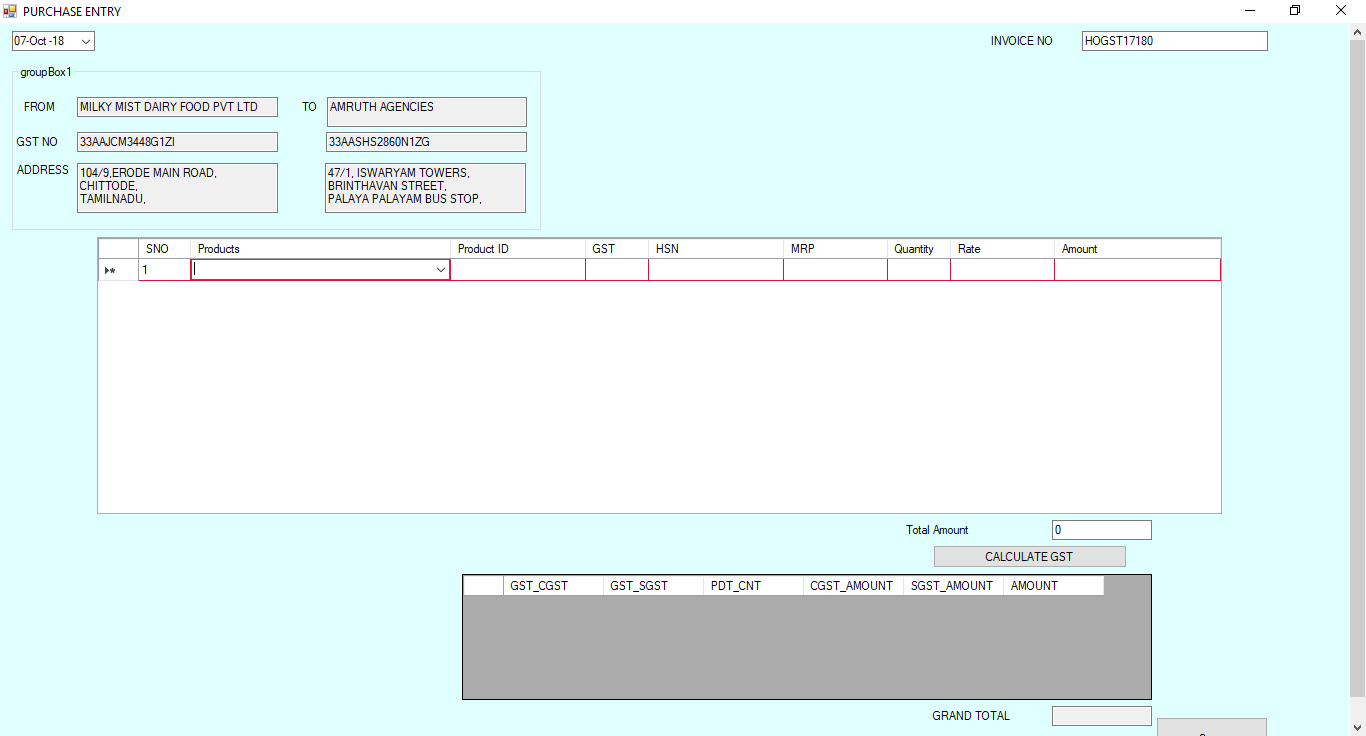
**Fig 8.2: Admin product entry page**



**Fig 8.3 : Opening stock entry page**



**Fig 8.4 : Sales entry page**



**Fig 8.5 : Purchase entry page**

**CHAPTER 9**

**9. REFRENCES**

Overview of DataGridview in .net:

<https://docs.microsoft.com/en-us/dotnet/api/system.windows.forms.datagridview.rowsdefaultcellstyle?view=netframework-4.7.2>

Uploading files : <https://www.c-sharpcorner.com/uploadfile/aa04e6/autocomplete-combobox-in-datagridview-using-C-Sharp-net-windows-application/>

Generating pdf: <https://bytes.com/topic/c-sharp/answers/442061-autocompletestringcollection-text-box-datagridview-2-0-a>