AUCTION MANAGEMENT SYSTEM

GUI VERSION

SESSION: 2021-2024

SUBMITTED BY:

Amna Imran Nagi

2021-CS-96

SUBMITTED TO:

Dr Awais

DEPARMENT OF COMPTER SCIENCE

UNIVERSITY OF ENGINEERING AND TECHNOLOGY

TABLE OF CONTENTS:

1. Class Diagram ….. 3 Page

2. Sequence Diagrams .... 4 Page

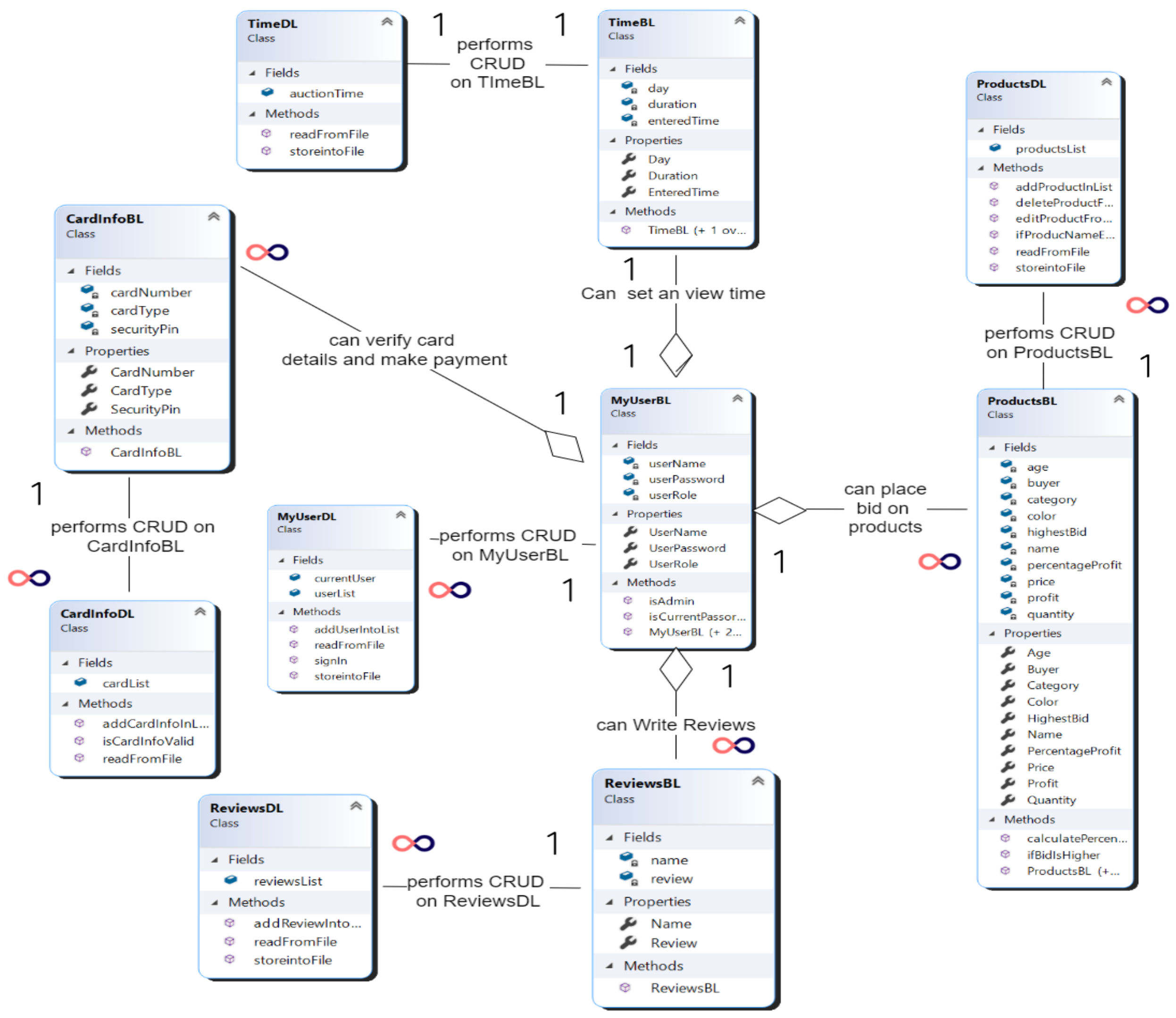
3. Wireframes .... 6 Page

4.Complete Code .... 13 Page

Auction Management System

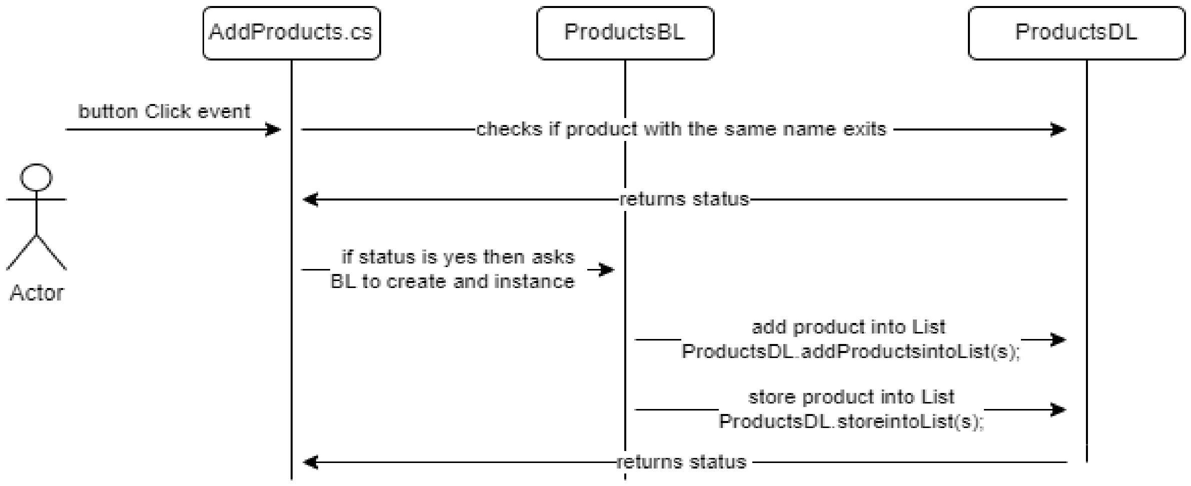
It is an Online Auction System where users can register as Auctioneer and Auction Goer. Where Auctioneers add products, sets the time of end of auction, view sales report, and see reviews with many other functionalities. Auction goers can place bids on the products on the products, make payments at the end of auction write reviews, change password and many others.

Class Diagram

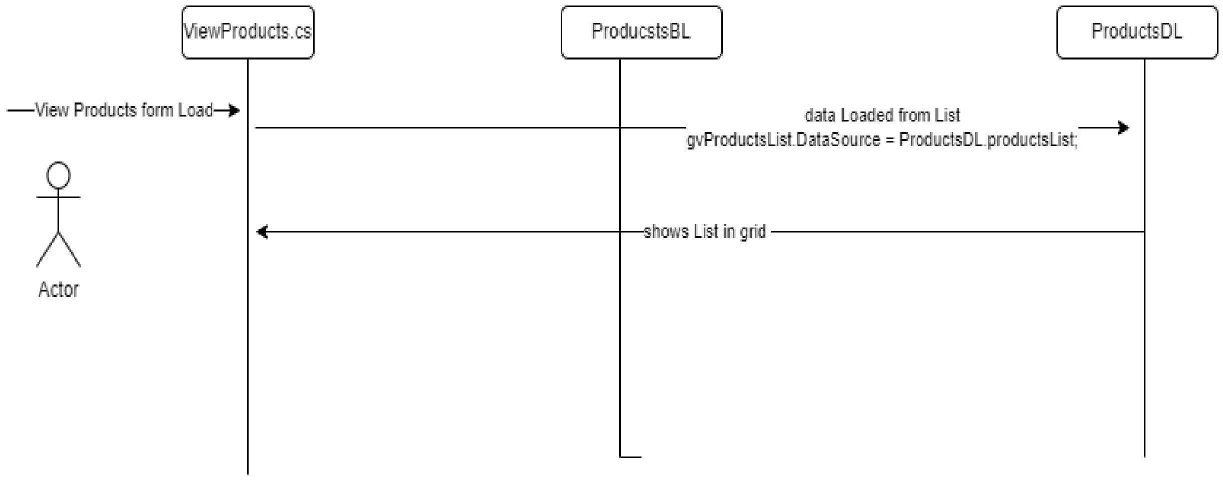


Sequence Diagrams of CRUD of ProductsBL

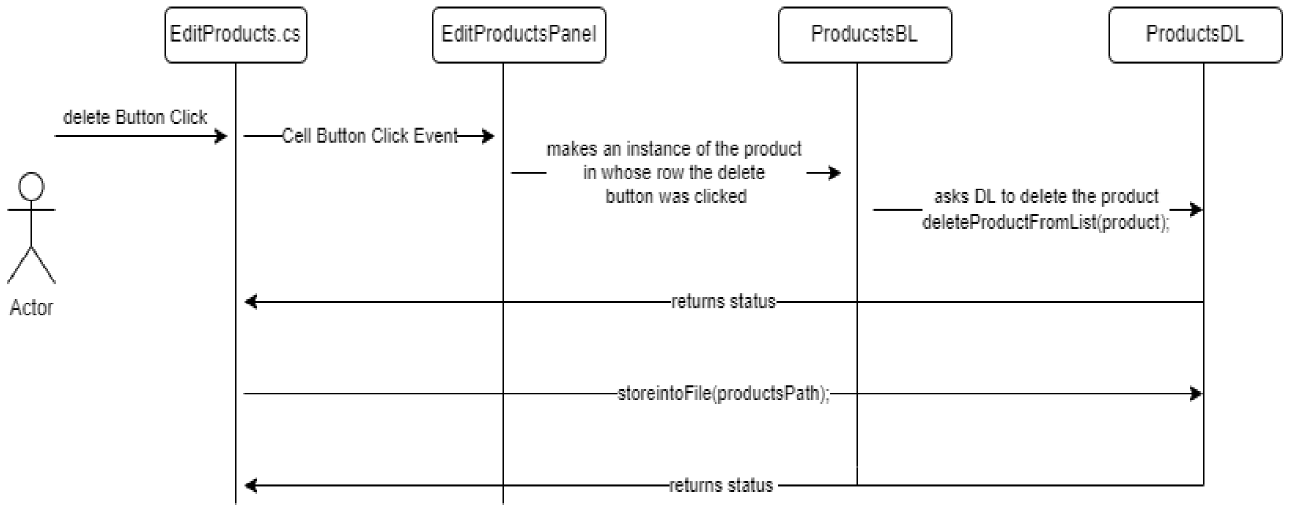
ADD Products


AddProducts.cs
ProductsBL
ProductsDL
button Click event
Actor
checks if product with the same name exits
returns status
if status is yes then asks BL to create and instance
add product into List ProductsDL.addProductsintoList(s);
store product into List ProductsDL.storeintoList(s);
returns status

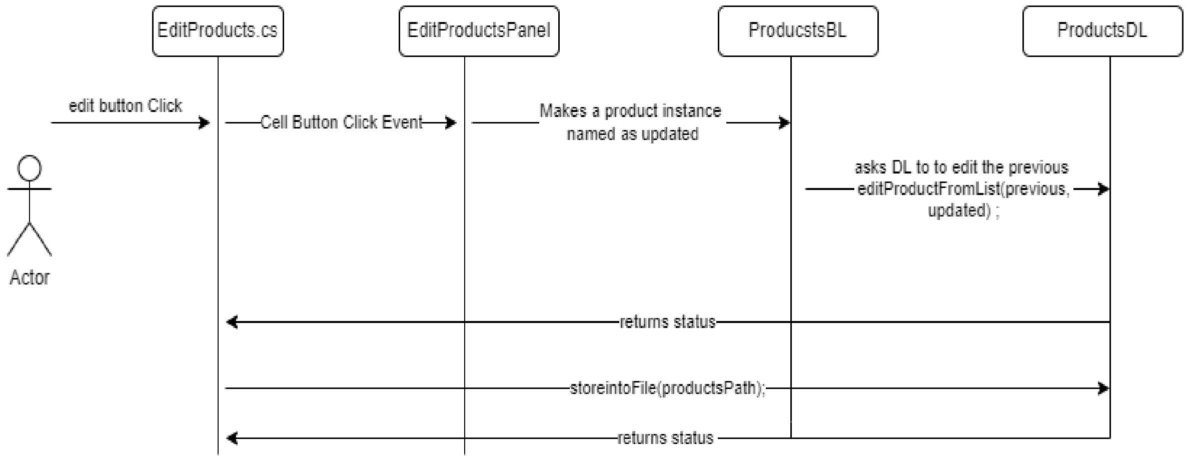
View Products



Delete Products

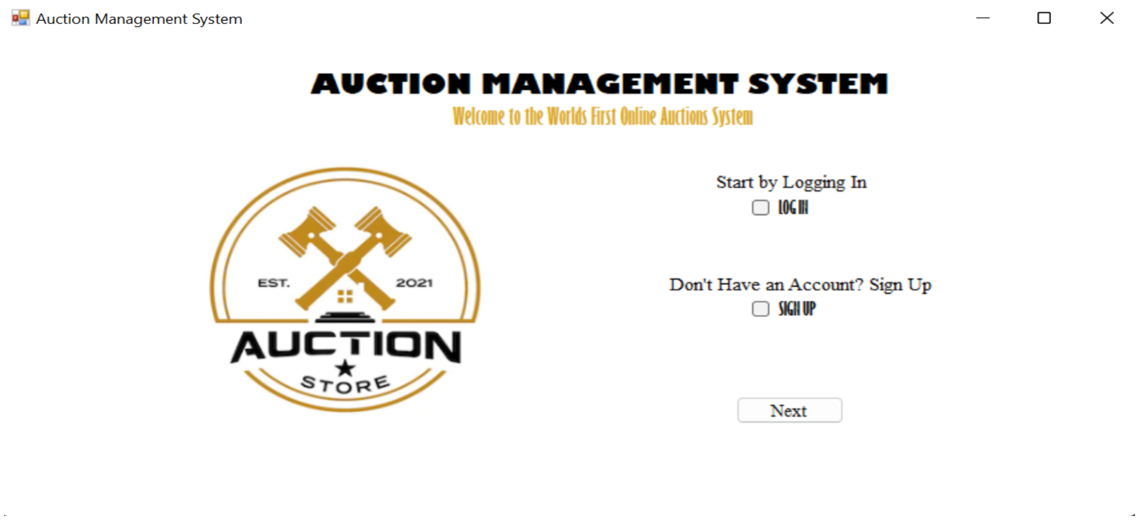

EditProducts.cs
EditProductsPanel
ProducstsBL
ProductsDL
Actor
delete Button Click
Cell Button Click Event
makes an instance of the product in whose row the delete button was clicked
asks DL to delete the product deleteProductFromList(product);
returns status
storeintoFile(productsPath);
returns status

Edit Products

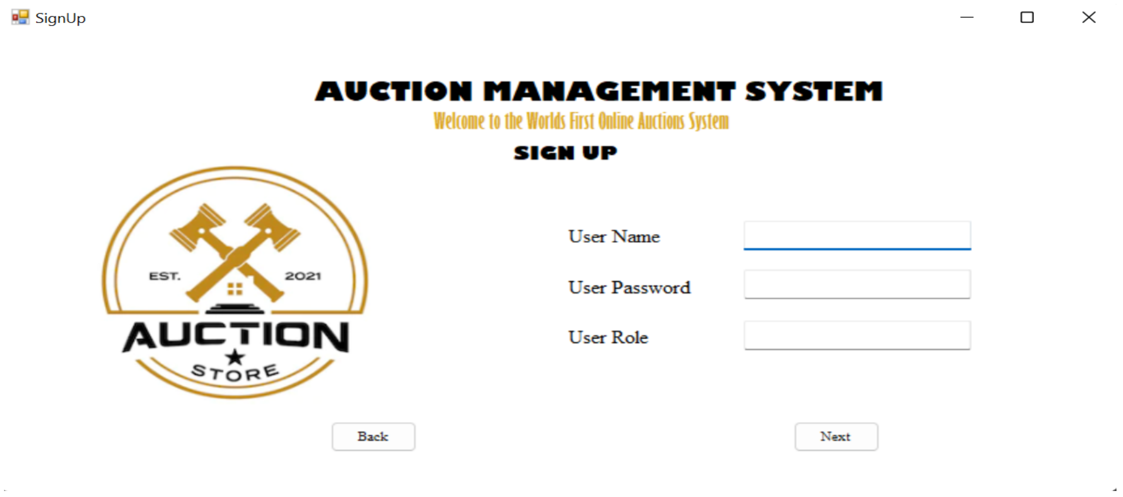

EditProducts.cs
EditProductsPanel
ProducstsBL
ProductsDL
Actor
edit button Click
Cell Button Click Event
Makes a product instance named as updated
asks DL to to edit the previous editProductFromList(previous, updated) ;
returns status
storeintoFile(productsPath);
returns status

Wireframes

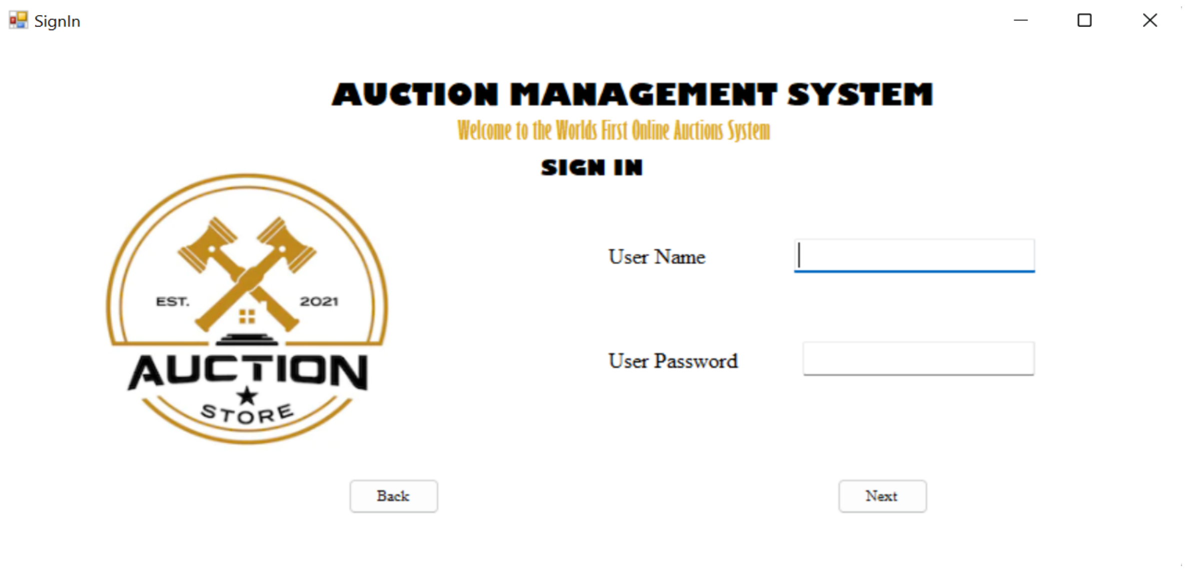
Log In Page


Auction Management System
AUCTION MANAGEMENT SYSTEM
Welcome to the Worlds First Online Auctions System
EST. 2021
AUCTION
STORE
Start by Logging In
LOG IN
Don't Have an Account? Sign Up
SIGN UP
Next

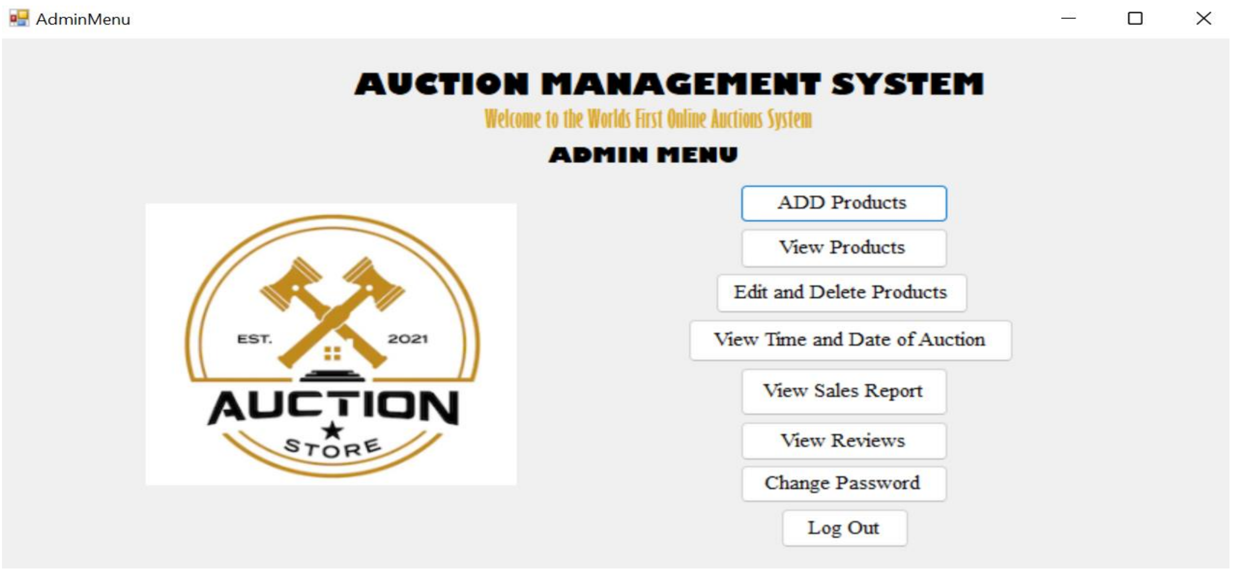
Sign Up

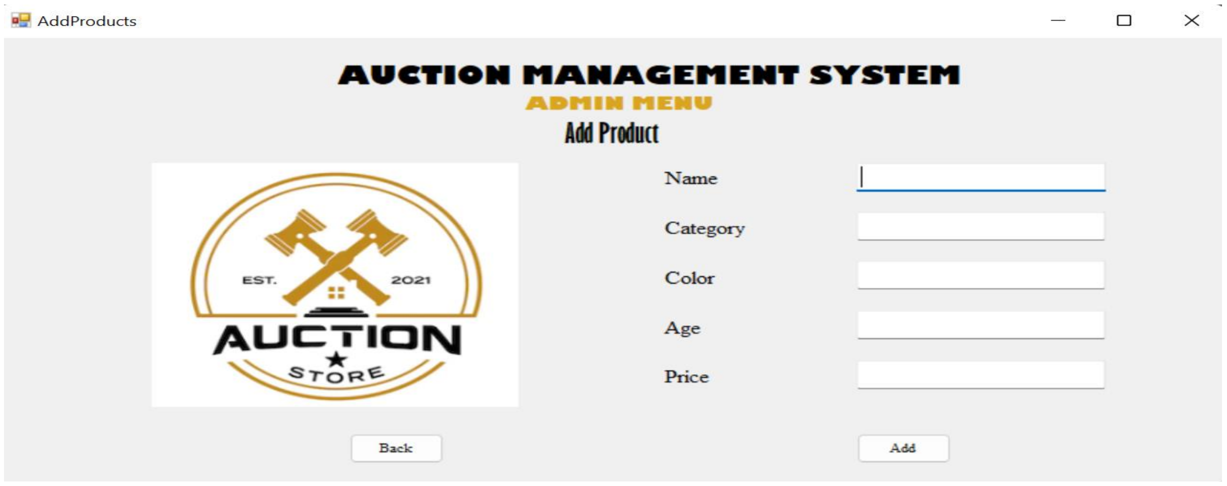

SignUp
AUCTION MANAGEMENT SYSTEM
Welcome to the Worlds First Online Auctions System
SIGN UP
EST. 2021
AUCTION
STORE
User Name
User Password
User Role
Back
Next

Sign In


SignIn
AUCTION MANAGEMENT SYSTEM
Welcome to the Worlds First Online Auctions System
SIGN IN
EST. 2021
AUCTION
STORE
User Name
User Password
Back
Next

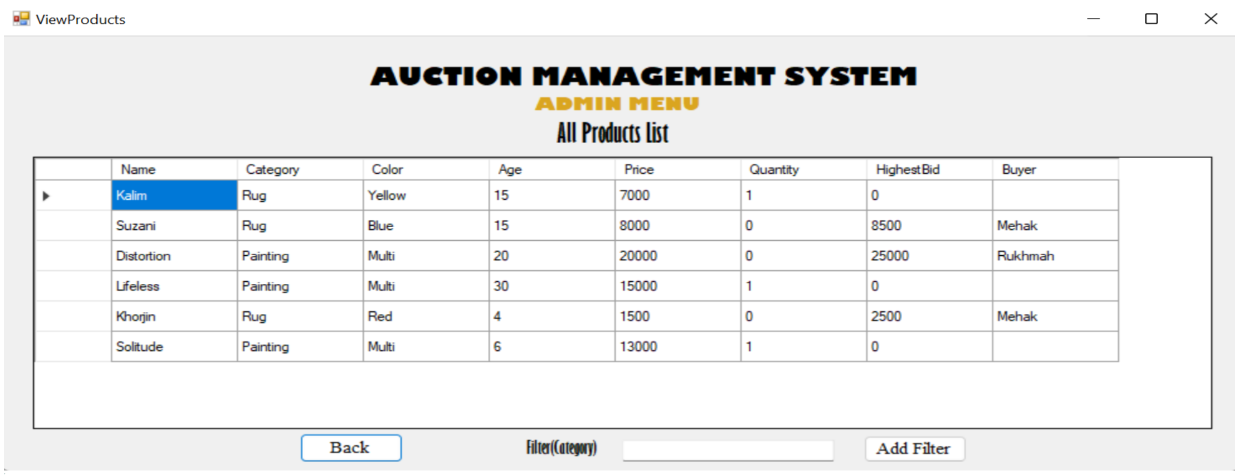
Admin Menu





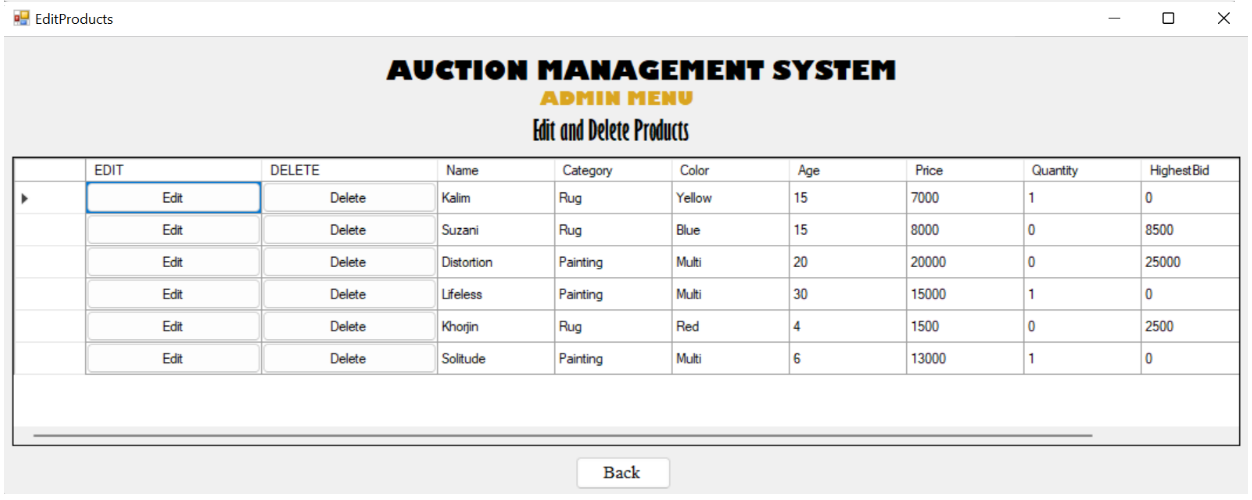
AUCTION MANAGEMENT SYSTEM
ADMIN MENU
Add Product

Name	
Category	
Color	
Age	
Price	



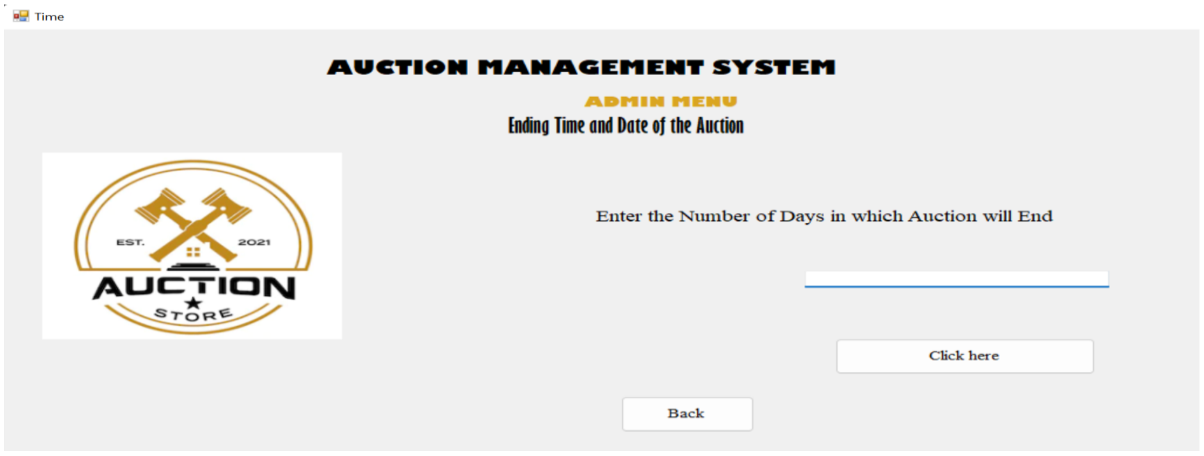
AUCTION MANAGEMENT SYSTEM
ADMIN MENU
All Products List

Name	Category	Color	Age	Price	Quantity	HighestBid	Buyer
Kalim	Rug	Yellow	15	7000	1	0	
Suzani	Rug	Blue	15	8000	0	8500	Mehak
Distortion	Painting	Multi	20	20000	0	25000	Rukhmah
Lifeless	Painting	Multi	30	15000	1	0	
Khorjin	Rug	Red	4	1500	0	2500	Mehak
Solitude	Painting	Multi	6	13000	1	0	



AUCTION MANAGEMENT SYSTEM
ADMIN MENU
Edit and Delete Products

EDIT	DELETE	Name	Category	Color	Age	Price	Quantity	HighestBid
Edit	Delete	Kalim	Rug	Yellow	15	7000	1	0
Edit	Delete	Suzani	Rug	Blue	15	8000	0	8500
Edit	Delete	Distortion	Painting	Multi	20	20000	0	25000
Edit	Delete	Lifeless	Painting	Multi	30	15000	1	0
Edit	Delete	Khorjin	Rug	Red	4	1500	0	2500
Edit	Delete	Solitude	Painting	Multi	6	13000	1	0

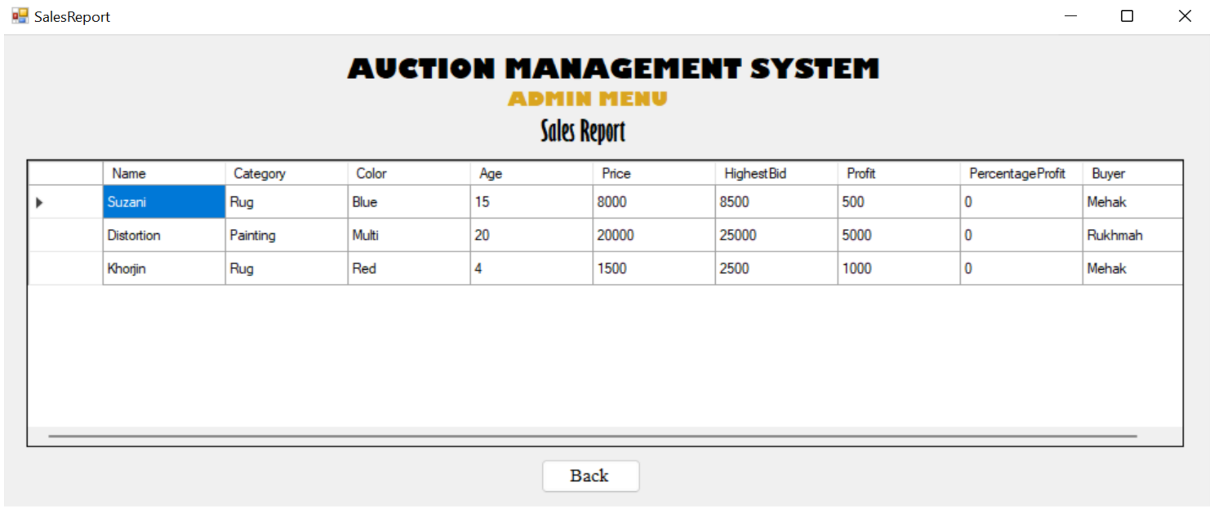

Time

AUCTION MANAGEMENT SYSTEM

ADMIN MENU

Ending Time and Date of the Auction

Enter the Number of Days in which Auction will End

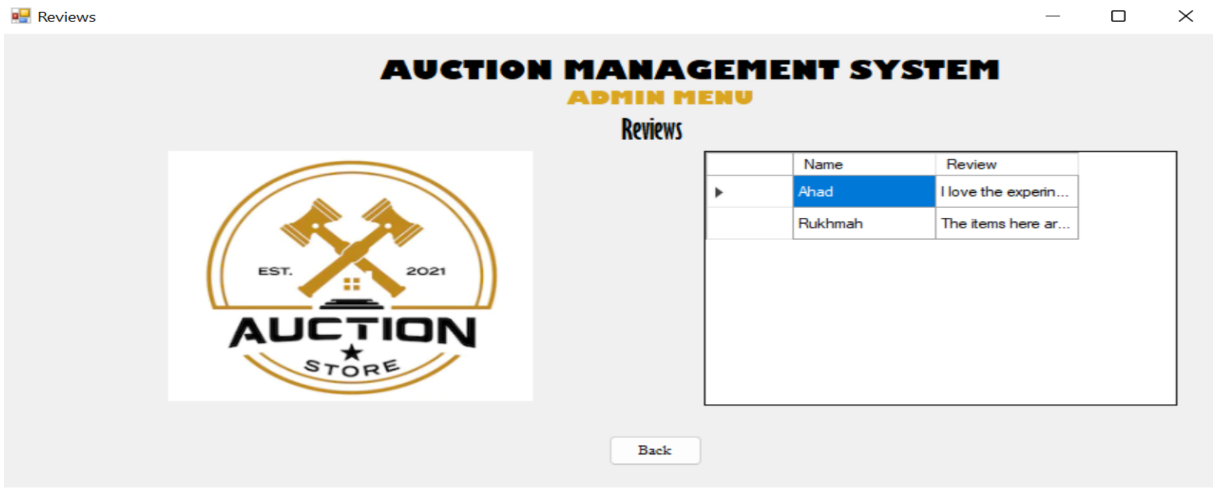

SalesReport

AUCTION MANAGEMENT SYSTEM

ADMIN MENU

Sales Report

Name	Category	Color	Age	Price	HighestBid	Profit	PercentageProfit	Buyer
Suzani	Rug	Blue	15	8000	8500	500	0	Mehak
Distortion	Painting	Multi	20	20000	25000	5000	0	Rukhmah
Khorjin	Rug	Red	4	1500	2500	1000	0	Mehak

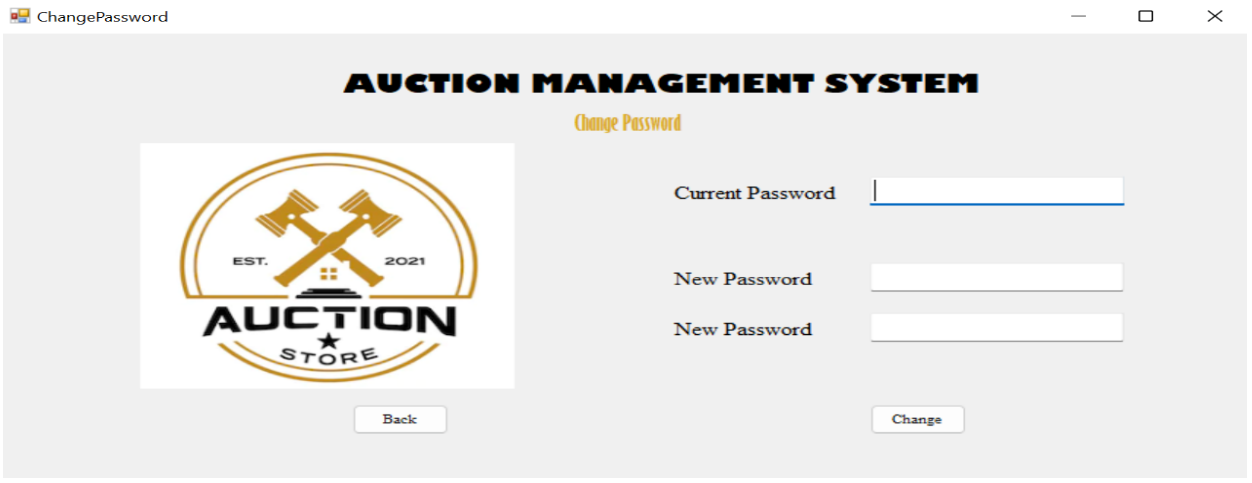

Reviews

AUCTION MANAGEMENT SYSTEM

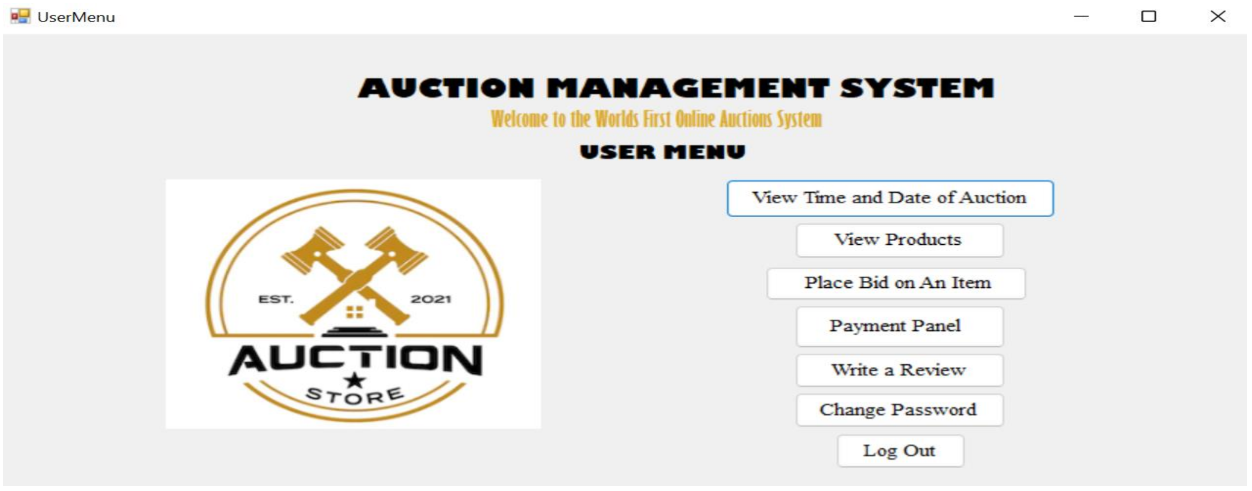
ADMIN MENU

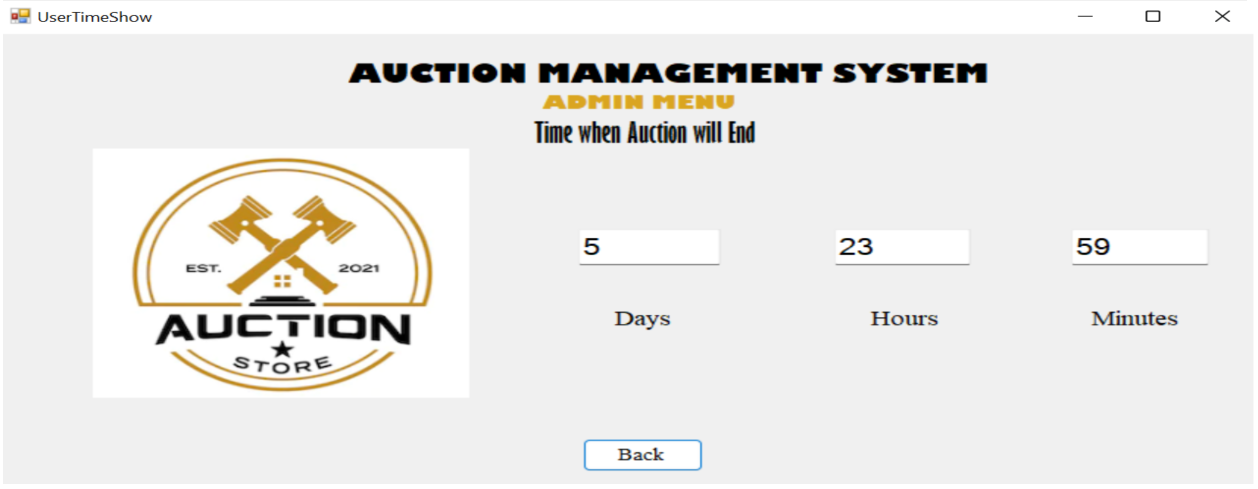
Reviews

Name	Review
Ahad	I love the experin...
Rukhmah	The items here ar...

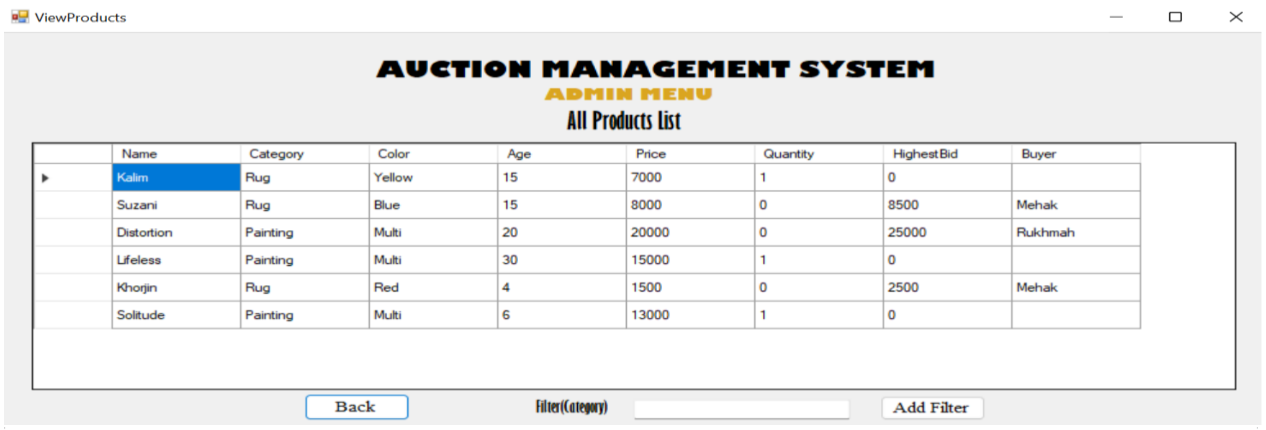


User Menu





Days	Hours	Minutes
5	23	59



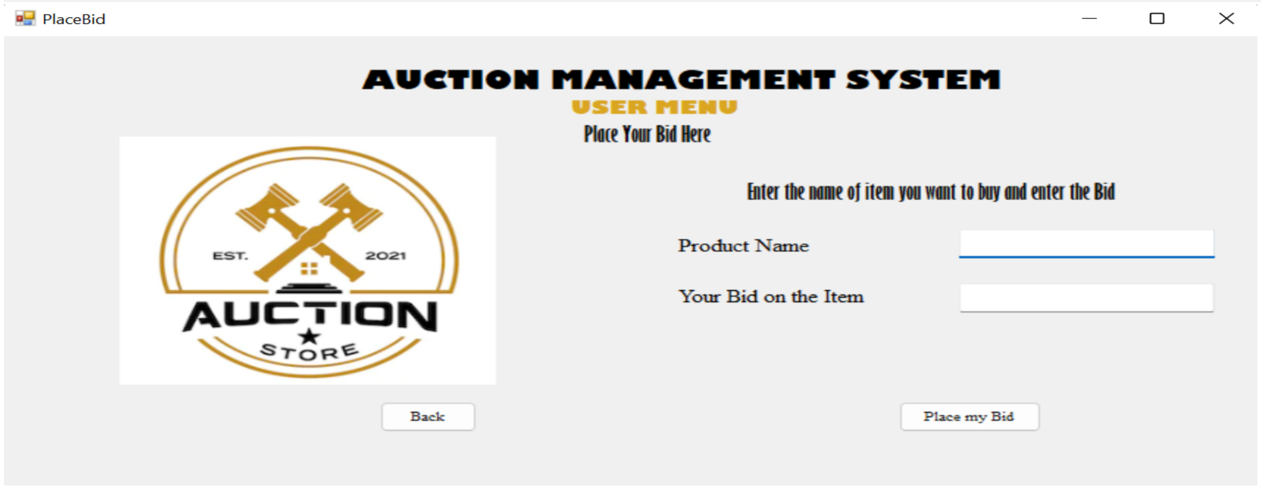
ViewProducts

AUCTION MANAGEMENT SYSTEM

ADMIN MENU

All Products List

Name	Category	Color	Age	Price	Quantity	HighestBid	Buyer
Kalim	Rug	Yellow	15	7000	1	0	
Suzani	Rug	Blue	15	8000	0	8500	Mehak
Distortion	Painting	Multi	20	20000	0	25000	Rukhmah
Lifeless	Painting	Multi	30	15000	1	0	
Khorjin	Rug	Red	4	1500	0	2500	Mehak
Solitude	Painting	Multi	6	13000	1	0	



PlaceBid

AUCTION MANAGEMENT SYSTEM

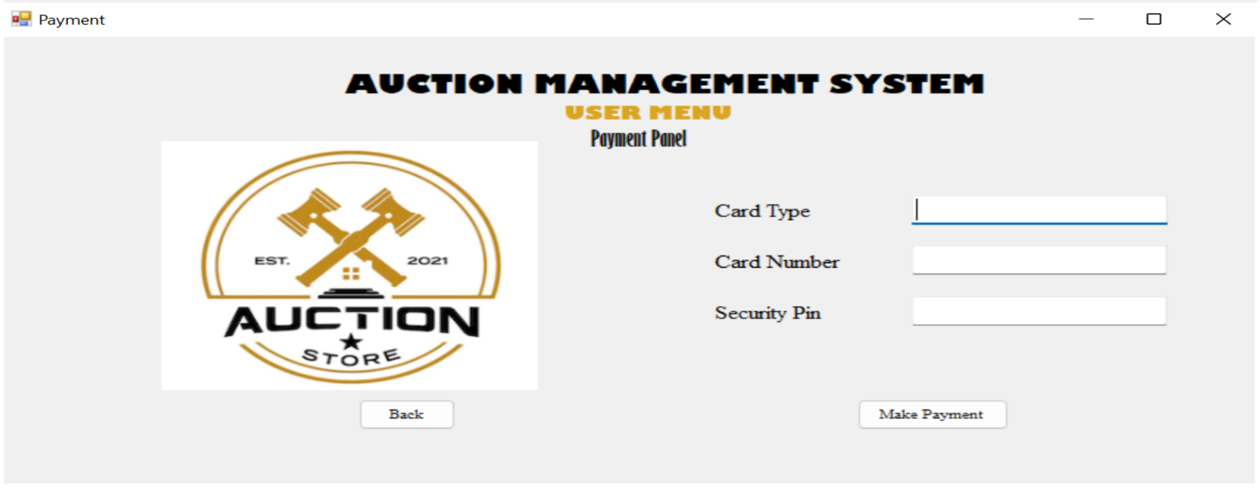
USER MENU

Place Your Bid Here

Enter the name of item you want to buy and enter the Bid

Product Name

Your Bid on the Item



Payment

AUCTION MANAGEMENT SYSTEM

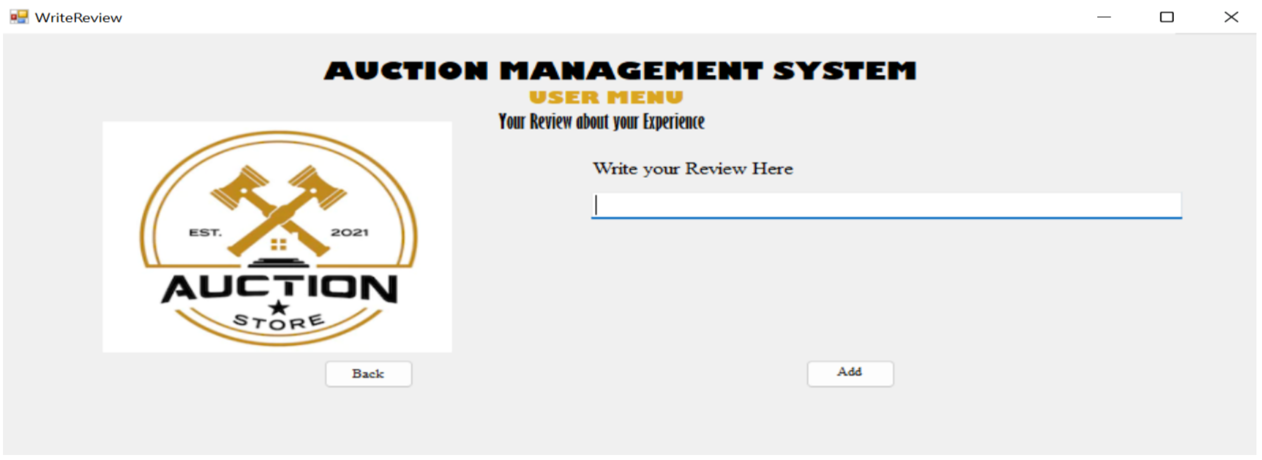
USER MENU

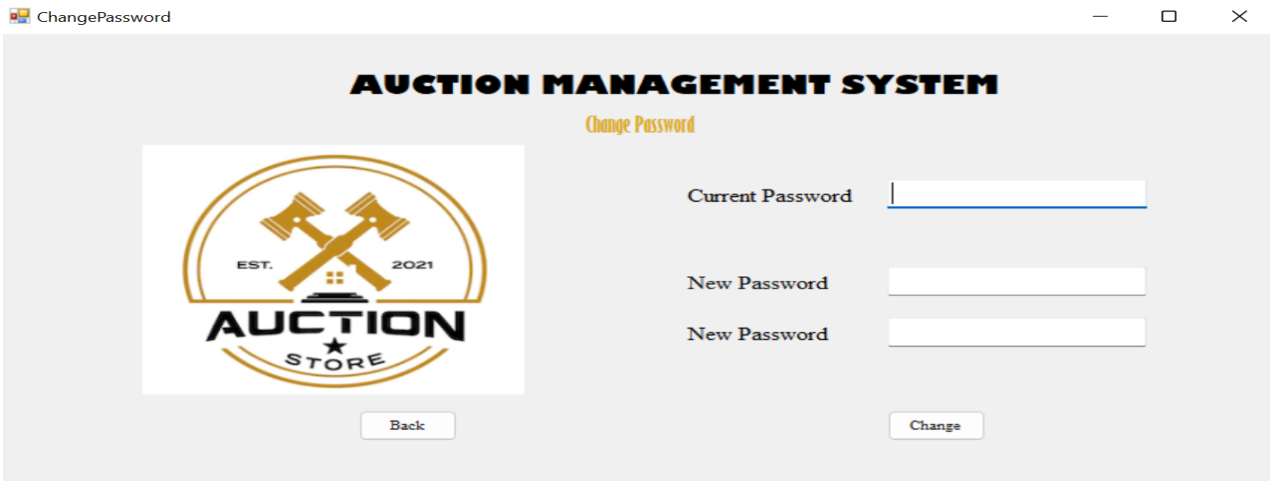
Payment Panel

Card Type

Card Number

Security Pin


WriteReview
AUCTION MANAGEMENT SYSTEM
USER MENU
Your Review about your Experience
Write your Review Here
EST. 2021
AUCTION
STORE
Back
Add


ChangePassword
AUCTION MANAGEMENT SYSTEM
Change Password
Current Password
New Password
New Password
EST. 2021
AUCTION
STORE
Back
Change

Full Code

MyUser BL

```
class MyUserBL
  {
      private string userName;
      private string userPassword;
      private string userRole;
      public string UserName { get => userName; set => userName = value; }
      public string UserPassword { get => userPassword; set => userPassword = value; }
      public string UserRole { get => userRole; set => userRole = value; }
      public MyUserBL()
      {
      }
      public MyUserBL(string userName, string userPassword, string userRole) // for signup
      {
          this.UserName = userName;
          this.UserPassword = userPassword;
          this.UserRole = userRole;
      }
      public MyUserBL(string userName, string userPassword) // for signin
      {
          this.UserName = userName;
          this.UserPassword = userPassword;
          this.UserRole = "NA";
      }
```

```
        public bool isAdmin() // checks if user is admin
        { if (UserRole == "Admin")
            {
                return true;
            }
            else
            {
                return false;
            }
        }
        public bool isCurrentPassordValid(string currentPassword)
        {
            if (currentPassword == UserPassword)
            {
                return true;
            }
            else
            {
                return false;
            }
        }
    }
```

MyUser DL

```
class MyUserDL
    {
        public static MyUserBL currentUser = new MyUserBL();
        public static List<MyUserBL> userList = new List<MyUserBL>();
        static public void addUserIntoList(MyUserBL User)
```

```
       {
           userList.Add(User);
        }
        static public MyUserBL signIn(MyUserBL User)
        {
           foreach (MyUserBL storedUser in userList)
           {
               if (User.UserName == storedUser.UserName && User.UserPassword ==
storedUser.UserPassword)
               {
                   currentUser = storedUser;
                   return storedUser;
               }
           }
           return null;
        }
        public static void storeintoFile(string path)
        {
           StreamWriter f = new StreamWriter(path);
           foreach (MyUserBL p in userList)
           {
               f.WriteLine(p.UserName + "," + p.UserPassword + "," + p.UserRole);
           }
          f.Flush();
           f.Close();
        }
        public static bool readFromFile(string path)
        {
           StreamReader f = new StreamReader(path);
```

```
            string record;
            if (File.Exists(path))
                    {
               while ((record = f.ReadLine()) != null)
               {
                  string[] splittedRecord = record.Split(',');
                  string userName = splittedRecord[0];
                  string userPassword = splittedRecord[1];
                  string userRole = splittedRecord[2];
                  MyUserBL s = new MyUserBL(userName, userPassword, userRole);
                  addUserIntoList(s);
               }
               f.Close();
               return true;
            }
            else
            {
               return false;
            }
         }
   }
```

Products BL

```
public class ProductsBL
   {
      private string name;
      private string category;
      private string color;
      private int age;
```

```
        private int price;
        private int quantity;
        private int highestBid;
        private int profit;
        private double percentageProfit;
        private string buyer;
        public string Name { get => name; set => name = value; }
        public string Category { get => category; set => category = value; }
        public string Color { get => color; set => color = value; }
        public int Age { get => age; set => age = value; }
        public int Price { get => price; set => price = value; }
        public int Quantity { get => quantity; set => quantity = value; }
        public int HighestBid { get => highestBid; set => highestBid = value; }
        public int Profit { get => profit; set => profit = value; }
        public double PercentageProfit { get => percentageProfit; set => percentageProfit = value; }
        public string Buyer { get => buyer; set => buyer = value; ;
        public ProductsBL()
        {}
 public ProductsBL(string name, string category, string color, int age, int price)
        {
            this.Name = name;
            this.Category = category;
            this.Color = color;
            this.Age = age;
            this.Price = price;
            Quantity = 1;
            HighestBid = 0;
            Profit = 0;
            Buyer = "";
            PercentageProfit = calculatePercentageProfit();
        }
        public ProductsBL(string name, string category, string color, int age, int price, int quantity, int
highestBid, int profit, double percenatgeProfit, string buyer)
        {
            this.Name = name;
            this.Category = category;
            this.Color = color;
            this.Age = age;
            this.Price = price;
            this.Quantity = quantity;
            this.HighestBid = highestBid;
            this.Profit = profit;
            this.Buyer = buyer;
            percenatgeProfit = calculatePercentageProfit();
        }
        public double calculatePercentageProfit()
        {
            return (Profit / Price) * 100;
        }


        public bool ifBidIsHigher(int bid, string username)
        {
            if (bid == Price || bid >= Price)
            {
                HighestBid = bid;
                Quantity = 0;
                Profit = HighestBid - Price;
                Buyer = username;
                calculatePercentageProfit();
                return true;
            }
            else
            {
                return false;
            }
        }
    }
```

Products DL

```
class ProductsDL
    {
        static public List<ProductsBL> productsList = new List<ProductsBL>();
        static public void addProductInList(ProductsBL p)
            {
            productsList.Add(p);
        }
        static public ProductsBL ifProducNameExists(string name)
        {
            bool nameCheck = false;
            ProductsBL product = new ProductsBL();
            foreach (ProductsBL p in ProductsDL.productsList)
            {
                if (p.Name == name)
                {
                    nameCheck = true;
                    product = p;
                }
            }
            if (nameCheck == true)
            {
                return product;
            }
            else
            {
                return null;
            }
        }
        public static void storeintoFile(string path)
        {
            StreamWriter f = new StreamWriter(path);
            foreach (ProductsBL m in productsList)
            {
                f.WriteLine(m.Name + "," + m.Category + "," + m.Color + "," + m.Age + "," + m.Price + "," +
m.Quantity + "," + m.HighestBid + "," + m.Profit + "," + m.PercentageProfit + "," + m.Buyer);
            }
            f.Flush();
            f.Close();
        }
        public static bool readFromFile(string path)
        {
            StreamReader f = new StreamReader(path);
            string record;
            if (File.Exists(path))
            {
                while ((record = f.ReadLine()) != null)
                {
                    string[] splittedRecord = record.Split(',');
                    string name = splittedRecord[0];
                    string category = splittedRecord[1];
                    string color = splittedRecord[2];
                    int age = int.Parse(splittedRecord[3]);
                    int price = int.Parse(splittedRecord[4]);
                    int quantity = int.Parse(splittedRecord[5]);
                    int highestBid = int.Parse(splittedRecord[6]);
                    int profit = int.Parse(splittedRecord[7]);
                    int percentageProfit = int.Parse(splittedRecord[8]);
                    string buyer = splittedRecord[9];
                    ProductsBL s = new ProductsBL(name, category, color, age, price, quantity, highestBid,
profit, percentageProfit, buyer);
                    productsList.Add(s);
                }
                f.Close();
                return true;
            }
            else
            {
                return false;
            }
        }
        public static void deleteProductFromList(ProductsBL product)
        {
            for (int index = 0; index < productsList.Count; index++)
            {
                if (productsList[index].Name == product.Name)
                {
                    productsList.RemoveAt(index);
                }
            }
     }
  public static void editProductFromList(ProductsBL previous, ProductsBL updated)
        {
            foreach (ProductsBL product in productsList)
            {
                if (product.Name == previous.Name)
                {
                    product.Name = updated.Name;
                    product.Category = updated.Category;
                    product.Color = updated.Color;
                    product.Age = updated.Age;
                    product.Price = updated.Price;
                }
            }
        }
     }
```

CardInfo BL

```
class CardInfoBL
    {
        private string cardType;
        private int cardNumber;
        private int securityPin;
        public CardInfoBL(string cardType, int cardNumber, int securityPin)
              {
            this.CardType = cardType;
            this.CardNumber = cardNumber;
            this.SecurityPin = securityPin;
        }
        public string CardType { get => cardType; set => cardType = value; }
        public int CardNumber { get => cardNumber; set => cardNumber = value; }
        public int SecurityPin { get => securityPin; set => securityPin = value; }
    }
```

CardInfo DL

```
class CardInfoDL
    {
        static public List<CardInfoBL> cardList = new List<CardInfoBL>();
        static public void addCardInfoInList(CardInfoBL p)
        {
            cardList.Add(p);
        }
        static public bool isCardInfoValid(CardInfoBL m)
        {
            foreach (CardInfoBL storedCard in cardList)
            {
                if (m.CardNumber == storedCard.CardNumber && m.CardType == storedCard.CardType &&
m.SecurityPin == storedCard.SecurityPin)
                {
                    return true;
                }
            }
            return false;
        }


        public static bool readFromFile(string path)
        {
            StreamReader f = new StreamReader(path);
            string record;
            if (File.Exists(path))
            {
                while ((record = f.ReadLine()) != null)
                {
                    string[] splittedRecord = record.Split(',');
                    string cardType = splittedRecord[0];
                    int cardNumber = int.Parse(splittedRecord[1]);
                    int securityPin = int.Parse(splittedRecord[2]);
                    CardInfoBL s = new CardInfoBL(cardType, cardNumber, securityPin);
                    addCardInfoInList(s);
                }
                f.Close();
                return true;
            }
            else
            {
                return false;
            }
        }
    }
```

Reviews BL

```
class ReviewsBL
    {
        private string name;
        private string review;
        public ReviewsBL(string name, string review)
        {
            this.Name = name;
            this.Review = review;
        }
        public string Name { get => name; set => name = value; }
        public string Review { get => review; set => review = value; }
    }
```

Reviews DL

```
class ReviewsDL
    {
        public static List<ReviewsBL> reviewsList = new List<ReviewsBL>();
        public static void addReviewIntoList(ReviewsBL s)
        {
            reviewsList.Add(s);
        }
        public static void storeintoFile(string path)
        {
            StreamWriter f = new StreamWriter(path);
                        foreach (ReviewsBL m in reviewsList)
            {
                f.WriteLine(m.Name + "," + m.Review);
            }
            f.Flush();
            f.Close();
        }
        public static bool readFromFile(string path)
        {
            StreamReader f = new StreamReader(path);
            string record;
            if (File.Exists(path))
            {
                while ((record = f.ReadLine()) != null)
                {
                    string[] splittedRecord = record.Split(',');
                    string name = splittedRecord[0];
                    string review = splittedRecord[1];
                    ReviewsBL s = new ReviewsBL(name, review);
                    reviewsList.Add(s);
                }
                f.Close();
                return true;
            }
            else
            {
                return false;
            }
        }
    }
```

Time BL

```
class TimeBL
    {
        private int day;
        private int enteredTime;
        private int duration;
        public TimeBL()
        {
        }
        public TimeBL(int day, int enteredTime, int duration)
        {
            this.Day = day;
            this.EnteredTime = enteredTime;
            this.Duration = duration;
        }
        public int Day { get => day; set => day = value; }
        public int EnteredTime { get => enteredTime; set => enteredTime = value; }
        public int Duration { get => duration; set => duration = value; }
    }
```

Time DL

```
class TimeDL
    {
        static public TimeBL auctionTime = new TimeBL();

        public static void storeintoFile(string path)
        {
            StreamWriter f = new StreamWriter(path);
            f.WriteLine(auctionTime.Day + "," + auctionTime.EnteredTime + "," + auctionTime.Duration);
            f.Flush();
            f.Close();
        }

        public static bool readFromFile(string path)
        {
            StreamReader f = new StreamReader(path);
            string record;
            if (File.Exists(path))
            {
                while ((record = f.ReadLine()) != null)
                {
                    string[] splittedRecord = record.Split(',');
                    int day = int.Parse(splittedRecord[0]);
                    int enteredTime  = int.Parse(splittedRecord[1]);
                    int duration = int.Parse(splittedRecord[2]);
                    TimeBL s = new TimeBL(day, enteredTime, duration);
                    auctionTime = s;
                }
                f.Close();
                return true;
            }
            else
            {
                return false;
            }
        }
    }
```

Form1

```
public partial class LogInPage : Form
    {
        public LogInPage()
        {
            InitializeComponent();
            string userPath = "userInfo.txt";
            if (MyUserDL.readFromFile(userPath))
            {
                MessageBox.Show("User Data Loaded Successfully");
            }
            else
            {
                MessageBox.Show("Data not Loaded");
            }
            string productsPath = "productsInfo.txt";
```

```
if (ProductsDL.readFromFile(productsPath))
{
   MessageBox.Show("Products Data Loaded Successfully");
}
else
{
   MessageBox.Show("Products not Loaded");
}
string cardsPath = "cardsInfo.txt";
if (CardInfoDL.readFromFile(cardsPath))
{
   MessageBox.Show("Cards Data Loaded Successfully");
}
else
{
   MessageBox.Show("Cards not Loaded");
}
string reviewsPath = "reviewsInfo.txt";
if (ReviewsDL.readFromFile(reviewsPath))
{
   MessageBox.Show("Reviews Data Loaded Successfully");
}
else
{
   MessageBox.Show("Reviews not Loaded");
}
string timePath = "timeInfo.txt";
            if (TimeDL.readFromFile(timePath))
{
```

```
            MessageBox.Show("Date and Time Data Loaded Successfully");
        }
        else
        {
            MessageBox.Show("Date and Time not Loaded");
        }
    }
    private void Form1_Load(object sender, EventArgs e)
    {
    }
    private void btnNext_Click(object sender, EventArgs e)
    {
        if (checkBoxSignIn.Checked)
        {
            SignIn form = new SignIn();
            form.Show();
            checkBoxSignIn.Checked = false;
        }
        else if (checkBoxSignUp.Checked)
        {
            SignUp form = new SignUp();
                        form.Show();
            checkBoxSignUp.Checked = false;
        }
    }
}
```

Sign In Form

```
public partial class SignIn : Form
{
```

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

public void ClearDataFromForm()
{
    txtUserName.Text = "";
    txtUserPassword.Text = "";
}

private void SignIn_Load(object sender, EventArgs e)
{

}

private void btnNext_Click(object sender, EventArgs e)
{
    string userName = txtUserName.Text;
    string userPassword = txtUserPassword.Text;
    MyUserBL User = new MyUserBL(userName, userPassword);
    MyUserBL validUser = MyUserDL.signIn(User);
    if (validUser != null)
    {
        MessageBox.Show("User is Valid");
        if (validUser.isAdmin())
        {
            AdminMenu form = new AdminMenu();
```

```
                form.Show();
            }
            else
            {
                UserMenu form = new UserMenu();
                form.Show();
            }
        }
        else
        {
            MessageBox.Show("User is Invalid");

        }
        ClearDataFromForm();
    }
     private void btnBack_Click(object sender, EventArgs e)
    {
        this.Close();
    }
}
```

Sign Up Form

```
public partial class SignUp : Form
    {
        public SignUp()
        {
            InitializeComponent();
        }


        private void SignUp_Load(object sender, EventArgs e)
```

```
{


}


private void ClearDataFromForm()
{
    txtUserName.Text = "";
    txtUserPassword.Text = "";
    txtUserRole.Text = "";
}
private void btnNext_Click(object sender, EventArgs e)
{
    string userName = txtUserName.Text;
    string userPassword = txtUserPassword.Text;
    string userRole = txtUserRole.Text;
    string userPath = "userInfo.txt";
    bool flag = false;
    foreach (MyUserBL m in MyUserDL.userList)
    {
        if (userName == m.UserName)
        {
            MessageBox.Show("This UserName Already Exists");
            flag = true;
        }
    }
    if (flag == false)
    {
        MyUserBL User = new MyUserBL(userName, userPassword, userRole);
        MyUserDL.addUserIntoList(User);
```

```
                MyUserDL.storeintoFile(userPath);

                MessageBox.Show("User Added Successfully");

            }

            ClearDataFromForm();

        }

        private void btnBack_Click(object sender, EventArgs e)

        {

            this.Close();

        }

    }
```

AdminMenu Form

```
public partial class AdminMenu : Form

    {

        public AdminMenu()

        {

            InitializeComponent();

        }

        private void btnAddProducts_Click(object sender, EventArgs e)

        {

            AddProducts form = new AddProducts();

            form.Show();

        }

        private void btnViewProducts_Click(object sender, EventArgs e)

        {

            ViewProducts form = new ViewProducts();

            form.Show();

        }


        private void button1_Click(object sender, EventArgs e)
        {
            EditProducts form = new EditProducts();
            form.Show();
        }
        private void btnTImeDate_Click(object sender, EventArgs e)
        {
            Time form = new Time();
            form.Show();
        }
        private void btnSalesReport_Click(object sender, EventArgs e)
        {
            SalesReport form = new SalesReport();
            form.Show();
        }
        private void btnReviews_Click(object sender, EventArgs e)
        {
            Reviews form = new Reviews();
            form.Show();
        }
        private void btnChangePassword_Click(object sender, EventArgs e)
        {
            ChangePassword form = new ChangePassword();
            form.Show();
        }
        private void btnBack_Click(object sender, EventArgs e)
        {
            this.Close();
        }
    }
```

AddProducts Form

```
public partial class AddProducts : Form
    {
        public AddProducts()
        {
            InitializeComponent();
        }
        private void ClearDataFromForm()
        {
            txtName.Text = "";
            txtCategory.Text = "";
            txtColor.Text = "";
            txtAge.Text = "";
            txtPrice.Text = "";
        }
         private void btnAdd_Click(object sender, EventArgs e)
        {
            string productsPath = "productsInfo.txt";
            string name = txtName.Text;
            string category = txtCategory.Text;
            string color = txtColor.Text;
            int age = int.Parse(txtAge.Text);
            int price = int.Parse(txtPrice.Text);
            bool flag = false;
            foreach (ProductsBL m in ProductsDL.productsList)
            {
                if (name == m.Name)
                {
                    MessageBox.Show("This Product Name Already Exists");
                    flag = true;
                }
            }
            if (flag == false)
            {
                ProductsBL p = new ProductsBL(name, category, color, age, price);
                ProductsDL.addProductInList(p);
                ProductsDL.storeintoFile(productsPath);
                MessageBox.Show("Product Added Successfully");
            }
            ClearDataFromForm();
        }

        private void btnBack_Click(object sender, EventArgs e)
        {
            this.Close();
        }
    }
```

ViewProducts Form

```
public partial class ViewProducts : Form
{
    public ViewProducts()
    {
        InitializeComponent();
    }
    private void lblAddProduct_Click(object sender, EventArgs e)
    {
    }
    private void ViewProducts_Load(object sender, EventArgs e)
```

```
        {
            gvProductsList.DataSource = ProductsDL.productsList;
            gvProductsList.Columns["Profit"].Visible = false;
            gvProductsList.Columns["PercentageProfit"].Visible = false;
        }
        private void btnBack_Click(object sender, EventArgs e)
        {
            this.Close();
        }
        private void gvProducts_CellContentClick(object sender, DataGridViewCellEventArgs e)
        {
        }
 private void btnAddFilter_Click(object sender, EventArgs e)
        {
            List<ProductsBL> rugs = new List<ProductsBL>();
            List<ProductsBL> paintings = new List<ProductsBL>();
            foreach(ProductsBL p in ProductsDL.productsList)
            {
                if(p.Category == "Rug")
                {
                    rugs.Add(p);
                }
                else if (p.Category == "Painting")
                {
                    paintings.Add(p);
                }
            }
            if (txtFilterCategory.Text == "Rug")
            {
                gvProductsList.DataSource = rugs;
                gvProductsList.Columns["Profit"].Visible = false;
                gvProductsList.Columns["PercentageProfit"].Visible = false;
            }
            else if (txtFilterCategory.Text == "Painting")
            {
                gvProductsList.DataSource = paintings;
                gvProductsList.Columns["Profit"].Visible = false;
                gvProductsList.Columns["PercentageProfit"].Visible = false;
            }
        }
    }
```

EditProducts Form

```
public partial class EditProducts : Form
    {
        public EditProducts()
        {
            InitializeComponent();
        }
        private void btnBack_Click(object sender, EventArgs e)
        {
            this.Close();
        }
        private void lblEditProducts_Load(object sender, EventArgs e)
        {
            gvEditProductsList.DataSource = ProductsDL.productsList;
            gvEditProductsList.Columns["Profit"].Visible = false;
            gvEditProductsList.Columns["PercentageProfit"].Visible = false;
        }
        public void BindGrid()
        {
            gvEditProductsList.DataSource = null;
            gvEditProductsList.DataSource = ProductsDL.productsList;
            gvEditProductsList.Refresh();
        }
        private void gvEditProductsList_CellContentClick(object sender, DataGridViewCellEventArgs e)
        {
            string productsPath = "productsInfo.txt";
            ProductsBL product = (ProductsBL)gvEditProductsList.CurrentRow.DataBoundItem;
            if (gvEditProductsList.Columns["Delete"].Index == e.ColumnIndex)
            {
                ProductsDL.deleteProductFromList(product);
                ProductsDL.storeintoFile(productsPath);
                BindGrid();
            }
            else if (gvEditProductsList.Columns["Edit"].Index == e.ColumnIndex)
            {
                EditProductPanel myform = new EditProductPanel(product);
                myform.ShowDialog();
                ProductsDL.storeintoFile(productsPath);
                BindGrid();
            }
        }
    }
```

EditProductsPanel Form

```
public partial class EditProductPanel : Form
    {
        private ProductsBL previous;
        public EditProductPanel(ProductsBL previous)

        {

            InitializeComponent();

            this.previous = previous;

        }

        private void btnBack_Click(object sender, EventArgs e)

        {

            this.Close();

        }

        private void btnEdit_Click(object sender, EventArgs e)

        {

            ProductsBL updated = new ProductsBL(txtName.Text, txtCatgeory.Text, txtColor.Text,
int.Parse(txtAge.Text), int.Parse(txtPrice.Text));

            ProductsDL.editProductFromList(previous, updated);

            this.Close();

        }

        private void EditProductPanel_Load(object sender, EventArgs e)

        {

            txtName.Text = previous.Name;

            txtCatgeory.Text = previous.Category;

            txtColor.Text = previous.Color;

            txtAge.Text = previous.Age.ToString();

            txtPrice.Text = previous.Price.ToString();

        }

    }
```

Time Form

```
public partial class Time : Form

    {

        public Time()

        {
            InitializeComponent();
        }
         private void btnShowEndOfAuction_Click(object sender, EventArgs e)
        {
            string timePath = "timeInfo.txt";
            int days = int.Parse(txtDays.Text);
            int duration = days * 86400;
            TimeSpan t = (DateTime.UtcNow - new DateTime(1970, 1, 1));
            int timestamp = (int)t.TotalSeconds;
            TimeDL.auctionTime.Day = days;
            TimeDL.auctionTime.EnteredTime = timestamp;
            TimeDL.auctionTime.Duration = duration;
            TimeDL.storeintoFile(timePath);
            MessageBox.Show("The Auction will End in " + days + " Days");
        }
        private void btnBack_Click(object sender, EventArgs e)
        {
            this.Close();
        }
    }
```

SalesReport Form

```
public partial class SalesReport : Form
    {
        public SalesReport()
        {
            InitializeComponent();
        }
        private void btnBack_Click(object sender, EventArgs e)
        {
            this.Close();
        }
        private void SalesReport_Load(object sender, EventArgs e)
        {
            gvSalesReport.DataSource = ProductsDL.productsList;
            gvSalesReport.Columns["Quantity"].Visible = false;
            for (int i = 0; i < gvSalesReport.Rows.Count; i++)
            {
                if (gvSalesReport.Rows[i].Cells[6].Value.ToString()=="0")
                {
                    CurrencyManager c = (CurrencyManager)BindingContext[gvSalesReport.DataSource];
                    c.SuspendBinding();
                    gvSalesReport.Rows[i].Visible = false;
                }
            }
        }
    }
```

Reviews Form

```
public partial class Reviews : Form
    {
        public Reviews()
        {
            InitializeComponent();
        }
        private void Reviews_Load(object sender, EventArgs e)
        {
            gvReviews.DataSource = ReviewsDL.reviewsList;
        }
        private void btnBack_Click(object sender, EventArgs e)
        {
            this.Close();
        }
    }
```

ChangePassword Form

```
public partial class ChangePassword : Form
    {
        public ChangePassword()
        {
            InitializeComponent();
        }
         private void ClearDataFromForm()
        {
            txtCurrentPassword.Text = "";
            txtNewPassword.Text = "" ;
            txtNewPassword2.Text = "";
        }
        private void ChangePassword_Load(object sender, EventArgs e)
        {
        }
        private void btnChange_Click(object sender, EventArgs e)
        {
            string userPath = "userInfo.txt";
            string currentUserName = MyUserDL.currentUser.UserName;
            string currentPassword = MyUserDL.currentUser.UserPassword;
            string currentPassordEntered = txtCurrentPassword.Text;
            string newPassword = txtNewPassword.Text;
            string newPassword2 = txtNewPassword2.Text;
            if (currentPassword == currentPassordEntered)
            {
                if (newPassword == newPassword2)
                {
                    foreach (MyUserBL p in MyUserDL.userList)
                    {
                        if (currentUserName == p.UserName)
                         {
                            p.UserPassword = newPassword;
                            MyUserDL.storeintoFile(userPath);
                            MessageBox.Show("Password has been Changed");
                         }
                    }
                }
                else
                {
                    MessageBox.Show("New Passwords Do not Match");
                }
            }
            else
            {
                MessageBox.Show("Wrong Current Password Entered");
            }
            ClearDataFromForm();
        }

        private void btnBack_Click(object sender, EventArgs e)
        {
            this.Close();
        }
    }
```

UserMenu Form

```
public partial class UserMenu : Form
    {
        public UserMenu()
        {
            InitializeComponent();
        }
        private void btnTImeDate_Click(object sender, EventArgs e)
        {
            UserTimeShow form = new UserTimeShow();
            form.Show();
        }
        private void btnViewProducts_Click(object sender, EventArgs e)
        {
            ViewProducts form = new ViewProducts();
            form.Show();
        }
        private void btnPlaceBid_Click(object sender, EventArgs e)
        {
            PlaceBid form = new PlaceBid();
            form.Show();
        }
        private void btnPayment_Click(object sender, EventArgs e)
        {
            Payment form = new Payment();
            form.Show();
        }
        private void btnReview_Click(object sender, EventArgs e)
        {
           WriteReview form = new WriteReview();
            form.Show();
        }
        private void btnChangePassword_Click(object sender, EventArgs e)
        {
            ChangePassword form = new ChangePassword();
            form.Show();
        }

        private void btnBack_Click(object sender, EventArgs e)
        {
            this.Close();
        }

        private void UserMenu_Load(object sender, EventArgs e)
        {

        }
    }
```

UserTimeShow Form

```
public partial class UserTimeShow : Form
    {
        public UserTimeShow()
        {
            InitializeComponent();
        }
```

```
private void UserTimeShow_Load(object sender, EventArgs e)
{
            TimeSpan t = (DateTime.UtcNow - new DateTime(1970, 1, 1));
   int userTime = (int)t.TotalSeconds;
   int diffSecs1 = userTime - TimeDL.auctionTime.EnteredTime;
   int diffSecs2 = TimeDL.auctionTime.Duration - diffSecs1;
   float diffDays = diffSecs2 / 86400.0F;
   int days = (int)diffDays;
   float decimalHours = diffDays - days;
   float durHours = decimalHours * 24.0F;
   int hours = (int)durHours;
   float decimalMins = diffSecs1 / 60.0F;
   float decimalMinutes = 60.0F - decimalMins;
   int mins = (int)decimalMinutes;
   if (diffSecs1 < TimeDL.auctionTime.Duration)
   {
      txtDays.Text = days.ToString();
      txtHours.Text = hours.ToString();
      txtMinutes.Text = mins.ToString();
   }
   else if (diffSecs1 > TimeDL.auctionTime.Duration)
   {
      MessageBox.Show("Auction has ended or it has not started yet!");
   }
}
 private void btnBack_Click(object sender, EventArgs e)
{
   this.Close();
}
```

```
    }
```

PlaceBid Form

```
public partial class PlaceBid : Form
    {
        public PlaceBid()
        {
            InitializeComponent();
        }
        private void ClearDataFromForm()
        {
                    txtProductName.Text = "";
            txtBid.Text = "";
        }
        private void btnPlaceBid_Click(object sender, EventArgs e)
        {
            string productsPath = "productsInfo.txt";
            string name = txtProductName.Text;
            int bid = int.Parse(txtBid.Text);
            ProductsBL productSelected = ProductsDL.ifProducNameExists(name);
            if (productSelected == null)
            {
                MessageBox.Show("No Item with this name exists! Please Try Again!");
            }
            else
            {
                bool checkBid = productSelected.ifBidIsHigher(bid, MyUserDL.currentUser.UserName);
                if (checkBid == true)
                {
                    MessageBox.Show("Item Sold!");
                    ProductsDL.storeintoFile(productsPath);
                }
                else
                {
                    MessageBox.Show("Bid is lower than Starting Price or Highest Bid! Please Try Again!");
                }
            }
            ClearDataFromForm();
        }
        private void btnBack_Click(object sender, EventArgs e)
        {
            this.Close();
        }
    }
```

Payment Form

```
public partial class Payment : Form
  {
    public Payment()
    {
        InitializeComponent();
    }

    private void ClearDataFromForm()
    {
        txtCardType.Text = "";
        txtCardNumber.Text = "";
        txtSecurityPin.Text = "";
    }
     private void btnMakePayment_Click(object sender, EventArgs e)
```

```
        {
            string cardType = txtCardType.Text;
            int cardNumber = int.Parse(txtCardNumber.Text);
            int securityPin = int.Parse(txtSecurityPin.Text);
                        CardInfoBL m = new CardInfoBL(cardType, cardNumber, securityPin);
            bool status = CardInfoDL.isCardInfoValid(m);
            if (status == true)
            {
                MessageBox.Show("Payment Made!");
            }
            else
            {
                MessageBox.Show("Wrong Card information entered! Please try again!");
            }
            ClearDataFromForm();
        }

        private void btnBack_Click(object sender, EventArgs e)
        {
            this.Close();
        }
    }
```

WriteReview Form

```
public partial class WriteReview : Form
    {
        public WriteReview()
        {
            InitializeComponent();
        }
        private void ClearDataFromForm()
        {
            txtReview.Text = "";
        }
        private void btnAddReview_Click(object sender, EventArgs e)
        {
            string reviewPath = "reviewsInfo.txt";
            string review = txtReview.Text;
            string name = MyUserDL.currentUser.UserName;
            ReviewsBL s = new ReviewsBL(name, review);
            ReviewsDL.addReviewIntoList(s);
            ReviewsDL.storeintoFile(reviewPath);
            MessageBox.Show("Your Review has been Added");
            ClearDataFromForm();
        }
        private void btnBack_Click(object sender, EventArgs e)
        {
            this.Close();
        }
    }
```