

MINI - PROJECT REPORT ON

**“Online Auction System”**

BY

1. Aman Bharti (2193032)
2. Gaurav Kumar (2193104)
3. Chinmay Mandavkar (2193088)
4. Aman raj (2193033)

**Under the Guidance of**

***Dr. Nilima Kulkarni***

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

**MIT SCHOOL of Engineering**

Loni Kalbhor Pune

**UNDERTAKING**

We declare that the work presented in this project titled “**Online Auction System**”, submitted to “**MIT ADT UNIVERSITY, SCHOOL OF ENGINEERING”** is our original work. Carried out from “April 2021 to June 2021” under the guidance of **Dr. Nilima Kulkarni**.

**Group Members**

**Aman Bharti,**

**Gaurav Kumar,**

**Chinmay Mandavkar,**

**Aman raj**

**M.I.T. SCHOOL OF ENGINEERING**

DEPARTMENT OF COMPUTER ENGINEERING

LONI – KALBHOR PUNE

***CERTIFICATE***



This is to certify that the Mini- Project report entitled

“*Online Auction System*”

**Submitted By**

Aman Bharti, Gaurav Kumar, Aman Raj, Chinmay Mandavkar

is a record of Bonafide work carried out by them, under my guidance, in partial fulfillment of the requirement for the Second Year of Engineering (Computer) at M.I.T. School of Engineering, Pune under MIT Art, Design & Technology University.

Date: 20/05/2021 Place: Pune

**Prof. Nilima Kulkarni Dr. RajneeshKaur Sachdeo**

**Guide, Dean Engineering,**

**Department of CSE Head, Department of CSE**

**M.I.T. School of Engineering MIT School of Engineering**

**Loni Kalbhor, Pune Loni-Kalbhor, Pune**

**ACKNOWLEDGEMENT**

We take upon this opportunity to acknowledge the many people whose efforts and

support helped me complete this project. We would like to thank ourMini Project guide **Nilima kulkarni ma’am** for providing us this opportunity. We also express our deepest gratitude to our mentor **Dr. Nilima kulkarni** once again for providing such meaningful and efficient knowledge.

Finally, I would like to wind up by paying my heartfelt thanks all my colleagues who

were there to help each other.

**Group Members**

**Aman Bharti,**

**Gaurav Kumar,**

**Chinmay Mandavkar,**

**Aman raj**

**CONTENTS**

* Acknowledgments
* Abstract
* Introduction
* Project Objective
* Requirements
* Project Purpose
* Overall description
* Modules
* Output Explanation
* Code
* Scope of the project
* Conclusion
* Reference

**ABSTRACT**

The online auction system is a java application that will help the user to purchase or sell an item; they can trade anything they want by posting an ad. This application will allow users to post their products for auction; bidders can register and bid for any available product.

**INTRODUCTION**

The internet has become a very important aspect of life today. The rate at which large populations are seeking to buy items is also on the rise, as many people are seeking advanced and ideal routes of trading services. Some people spend a lot of money on transportation, using a lot of time at the end of the road they might lack to get the desired items which they opted for.

It is open that most people are seeking to buy items every day in most cities and towns, both locally and abroad, desperate if they might get a solution or the right person to deliver a solution to them. These people end up settling at items or service providers who are common or who sell illegal and fake items to desperate buyers.

On the other hand, there are legit business people and store owners who have quality items in the locality of the auctioneer, but they do not meet for business because the auctioneer is not informed about the items available. This is great frustration indeed!

This project shall handle this issue by creating an online platform where a user will be able to post items online for auction. The items will accompany the item name, selling price, and a picture presentation for the bidder to see. The bidder, if interested in the item, will auction for the product and will be able to inspect the item to approve the product then complete the business with the seller.

This is important since the auctioneer does not necessarily need to make a physical consultation with the seller for him/ her to get the required services. The customers will be assured of getting the right products since they will take their time to analyze and compare a range of listed items and choose appropriately according to their needs or desire. This will save time that buyers take in search of items and therefore they will save themselves from worsening conditions which may lead to wastage of time conditions. This will also save money that is spent around traveling and bidding for undesired items. Bidders will at the end of the day have a reason to smile with this online system.

**PROJECT OBJECTIVE**

To develop an online auction system that will provide a platform for sellers to interact with buyers, and sell items to interested bidders.

The Online auction management system shall accomplish the following as a way of achieving the major goal:

1. Create a java application where bidders auction for items posted by the seller through the online system.
2. Have a reliable user checking.
3. To implement and test the work-ability of the newly developed system.

**REQUIREMENTS**

### **1. User Interface Requirements**

Simple Graphical user interface (GUI) for easy navigating through the program

Easy to update profile and items

Dynamically configurable interface

Search functions

Appealing to the eye through coloration and pictorial presentation

### 

### **2. Hardware Interface Requirements**

Processor speed of 0.5 GHz or more for mobile gadgets

Processor speed of 1.5ghz or more for desktop and computer gadgets

Ram of 500Mb and above for all devices

Free storage memory capacity of more than 100Mb

### 

### **3. Communication Interface Requirements**

1. Internet connectivity

### 

### **4.Development tools**

1. MySQL database
2. Eclipse IDE
3. Java swing

## **PROJECT PURPOSE**

To develop an online auction system which will provide a platform for sellers to interact with buyers, and sell items to interested bidders.

**The Online auction management system shall accomplish the following as way of achieving the major goal:**

To build a user friendly auctioning website, where users will be able to auction any product which is available nearby or anywhere in the world. By using Online Auction management system it will be easy for auctioneers to make an auction and time saving also. By making auctions through this application will help to reach the maximum of buyers bidding in the local area.

To implement and test the work-ability of the newly developed system.

## **OVERALL DESCRIPTION**

### 1. Product Perspective

The following are the main features;

1. Cross platform support – it offers operation support for most of the known and commercial operating systems including windows operating system
2. User accounts – this system allows the system users to create system accounts, view and update their profiles.
3. Users supported in the system – the system supports quite a large number of users at one time.

### 2. Product Function

1. Enable the users to view posted products with show item menu
2. Enable creation of accounts and logging in to the accounts
3. Enable users to logout of their accounts
4. Provide an interface for the administrator to view the transactions, add sellers and items for auction to the system.
5. The sellers should also have accounts including their login and seller profile.

### 3. User Characteristics

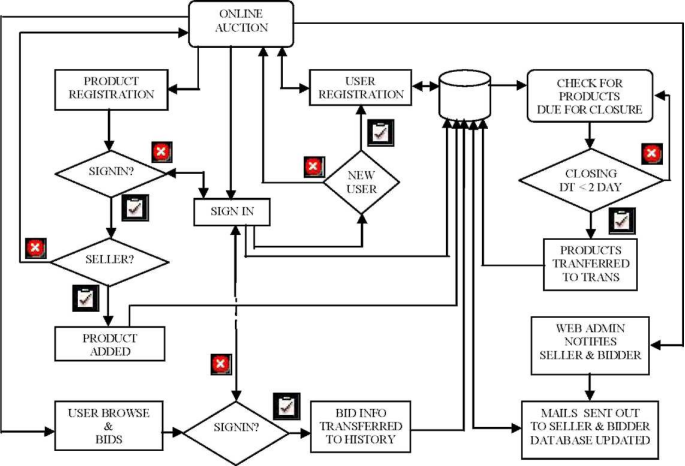
The system requires that the users be equipped with overall internet knowledge and the gadget accessing it. The administrator is expected to have more skills with the interface of the tech support system. The sellers should also have skills of well analyzing and navigating the internet usage and be able to handle customers who may not be so familiar with the internet.

### 4. Constraints

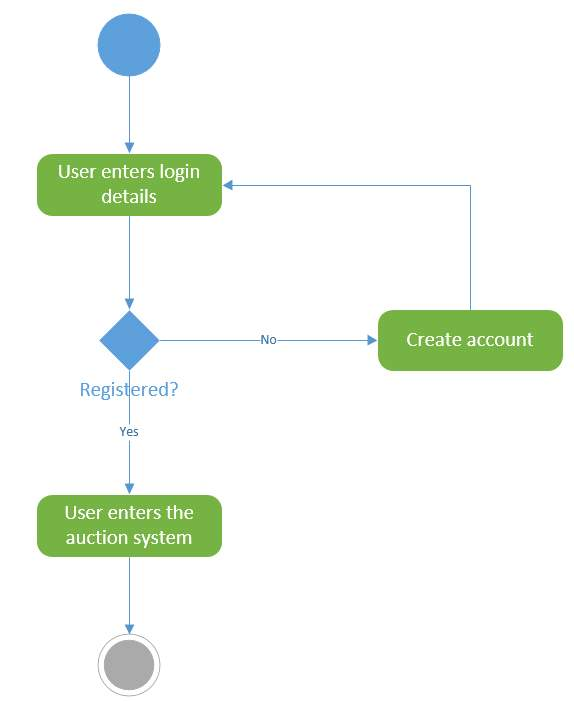
The choice of the database to use should be considered carefully, since there will be a lot of data traffic and the large amounts of data will also judge the database. A good database will yield speed querying of data.

The device should be enabled with internet

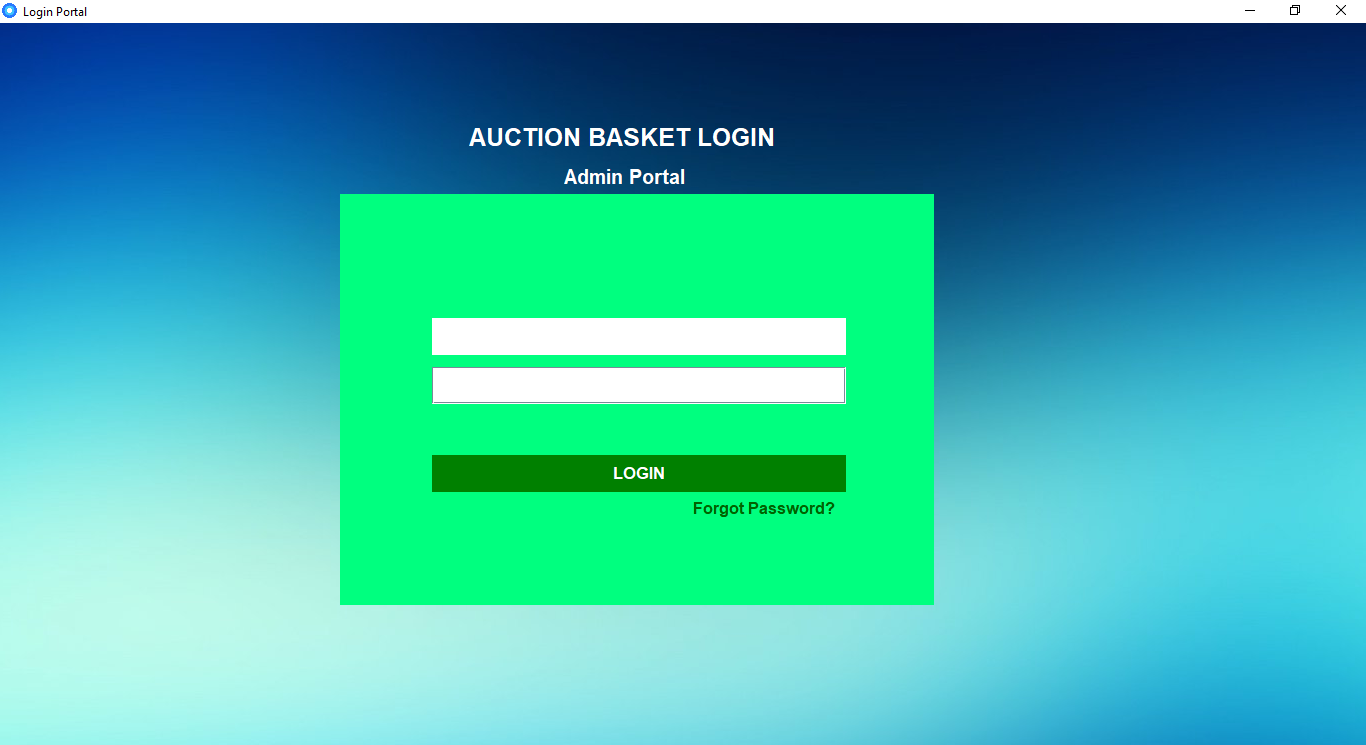
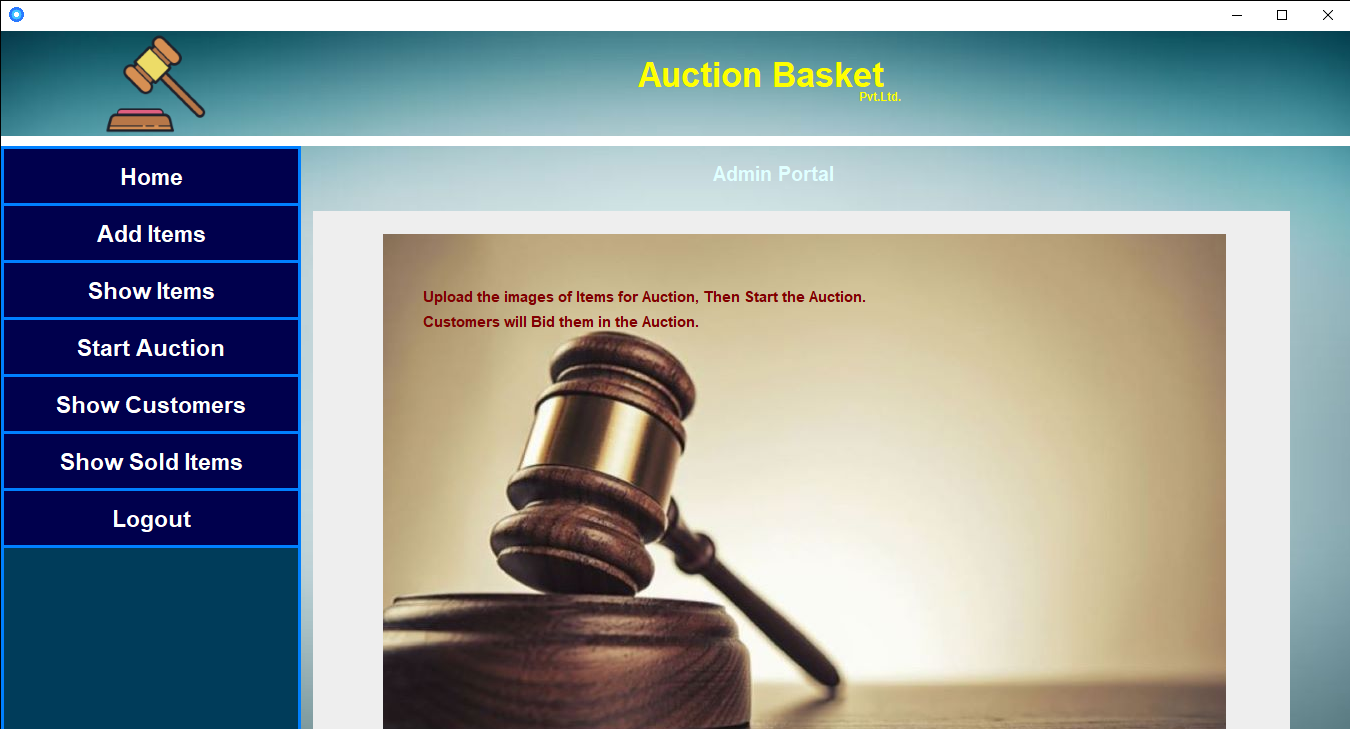
5. Figures

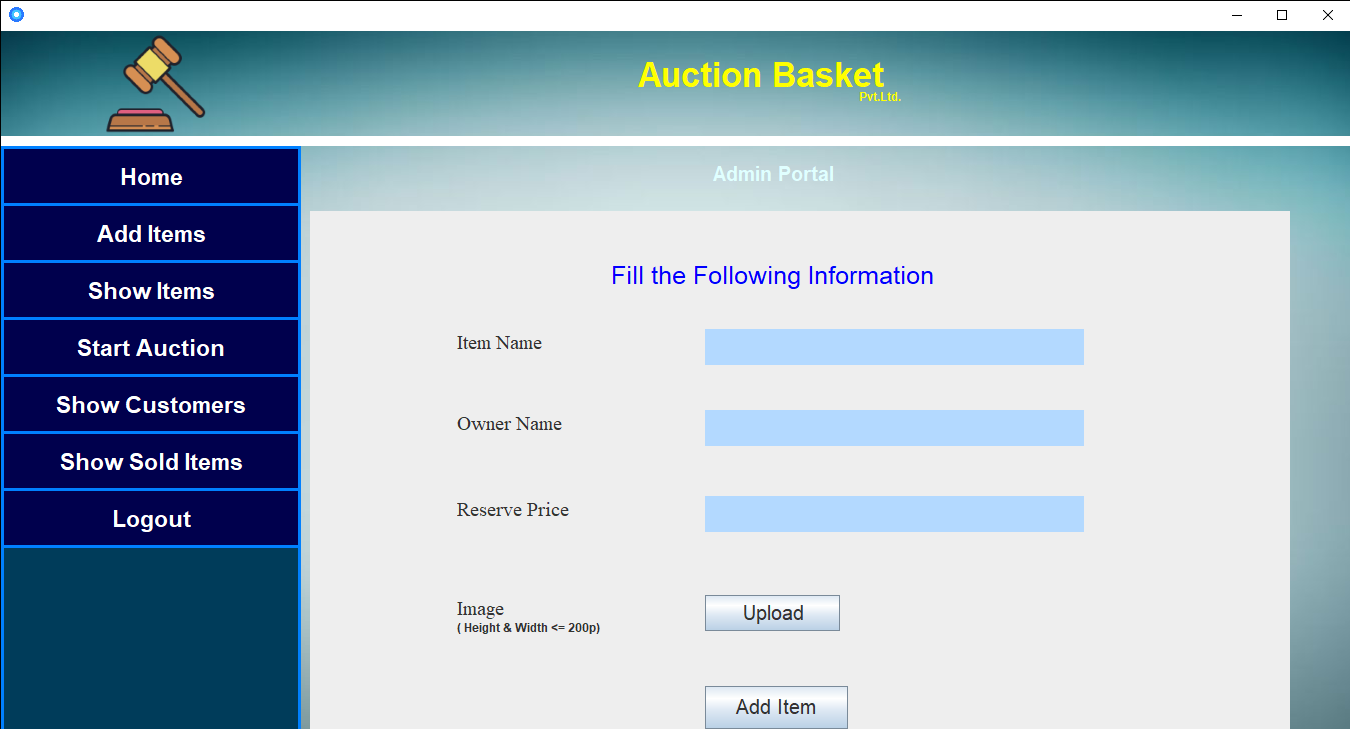
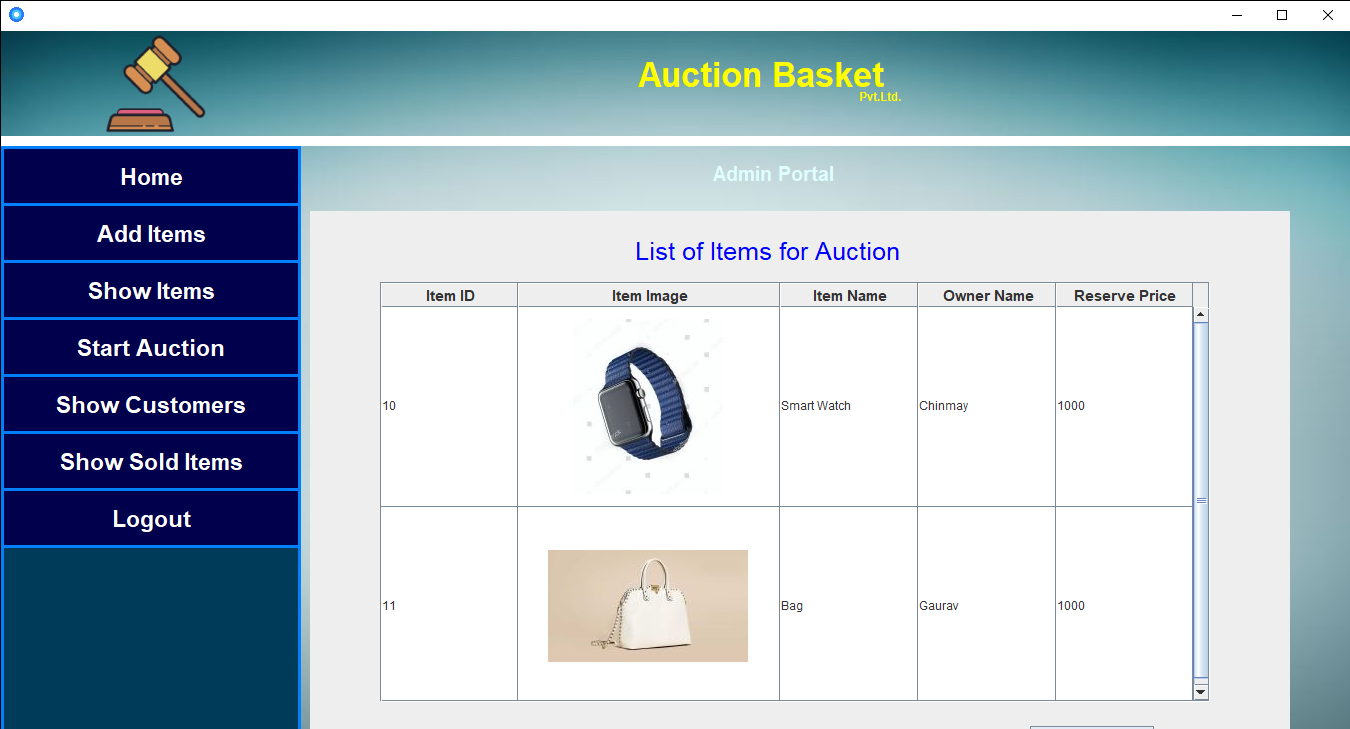
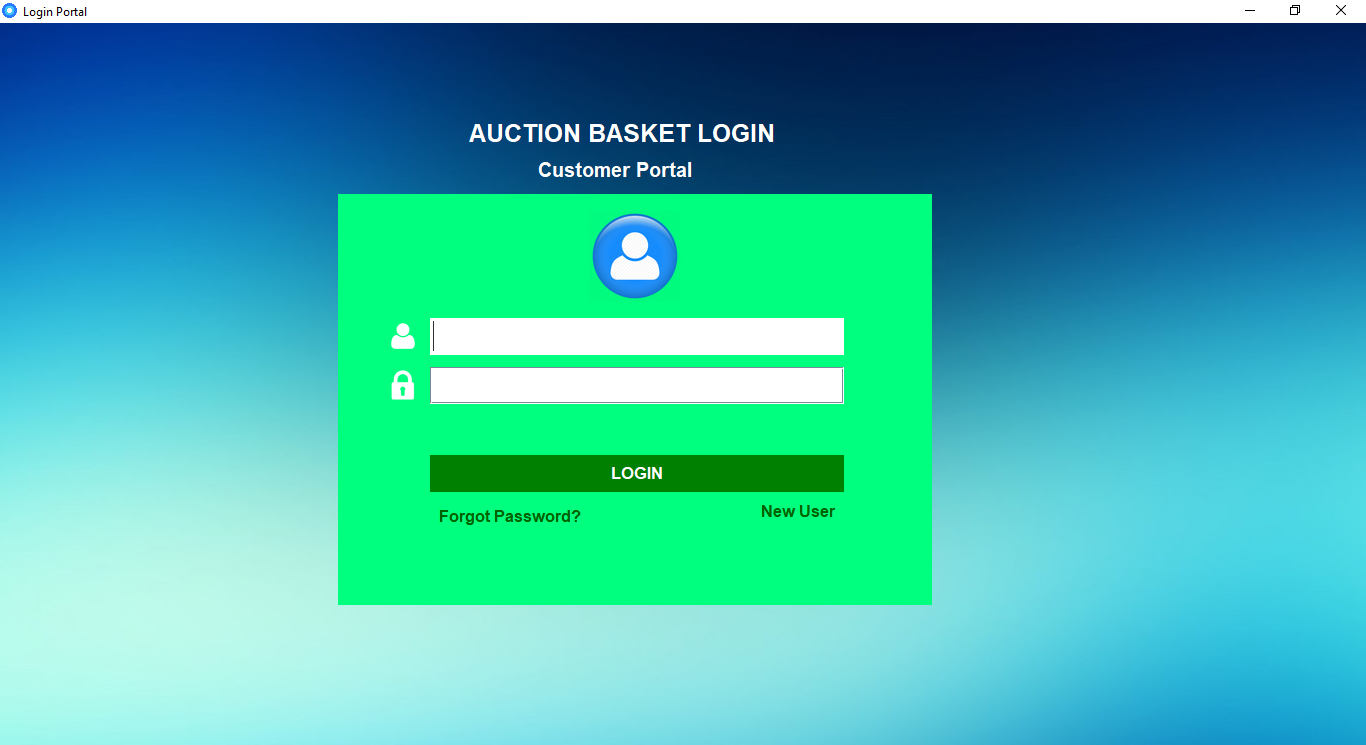
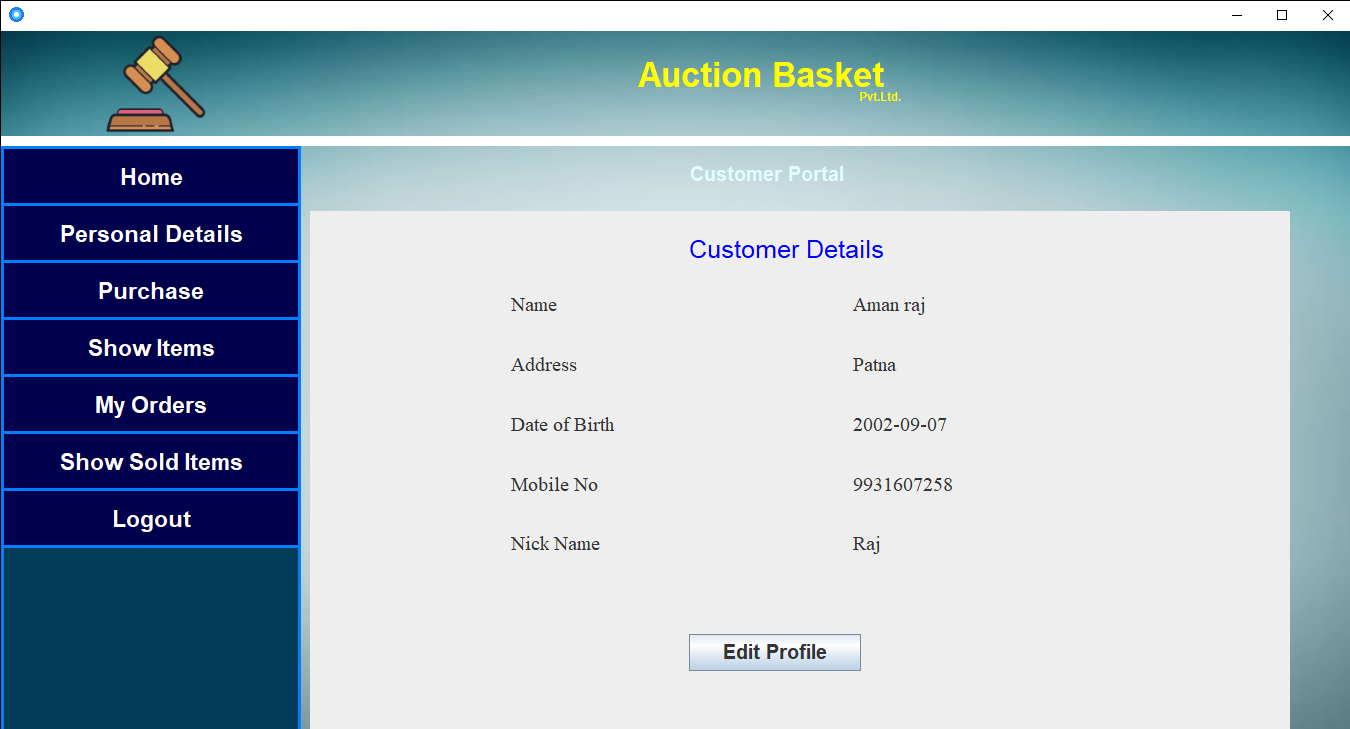
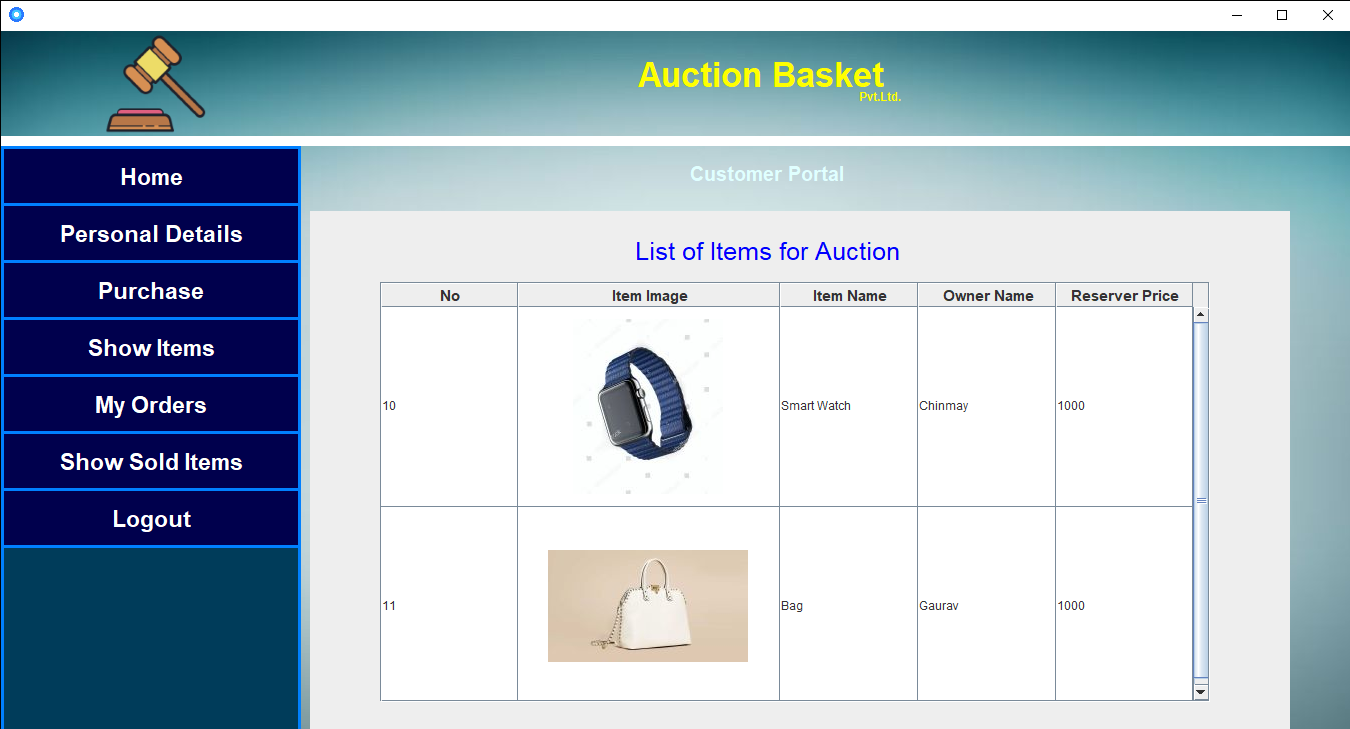
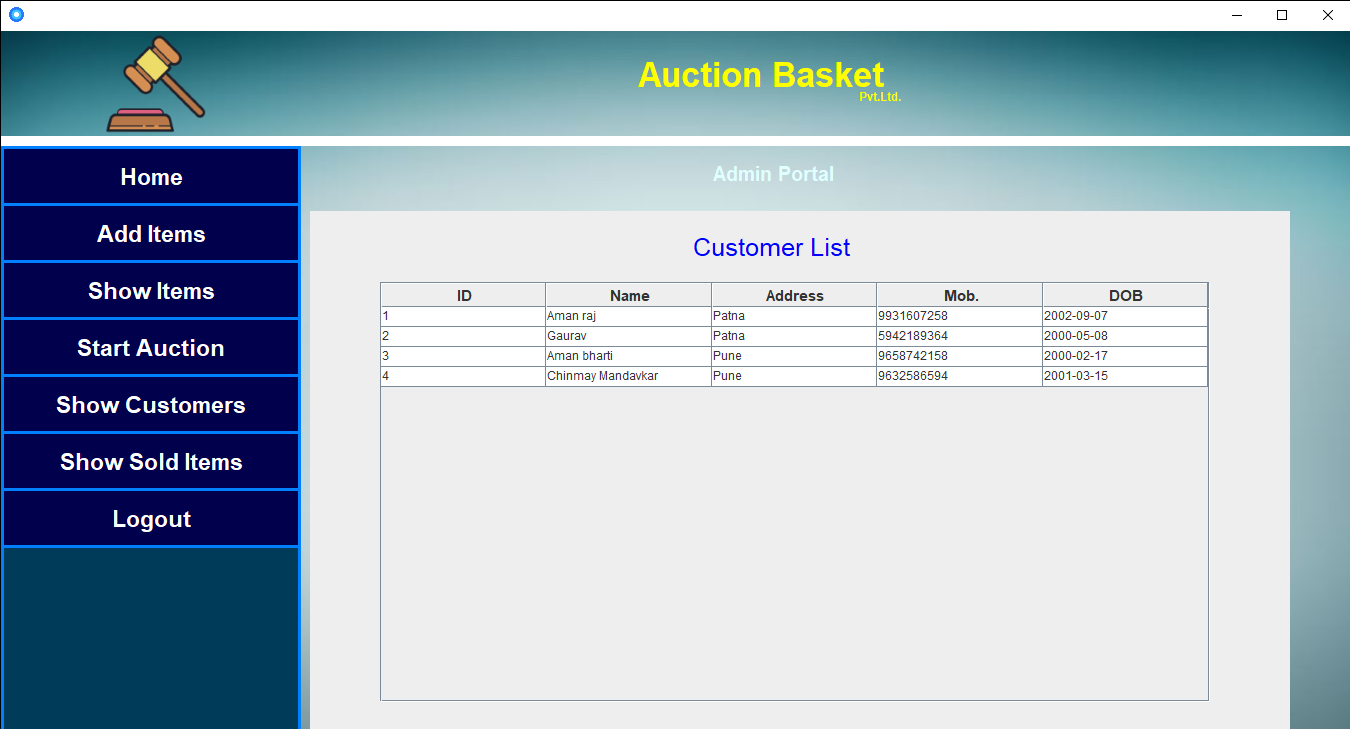


New User Registration

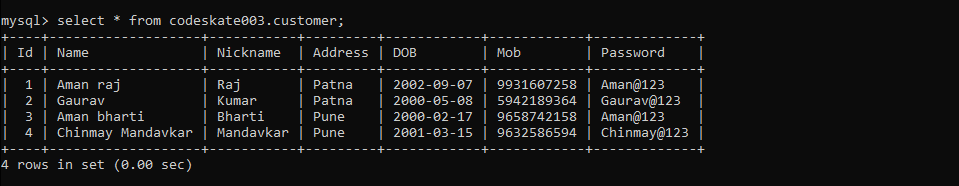
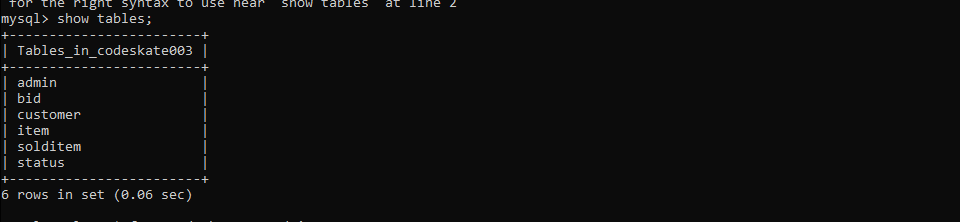


**Modules**

1. **Admin Login Page **
2. **Admin home Page **

1. **Add item **
2. **Show Item **
3. **Customer Login Page **
4. **Customer details **
5. **Customer side item List **
6. **Customer List **

**Database Tables**

****

**Code**

**AdminLogin.java**

package Login;

import java.awt.Color;

import java.awt.Container;

import java.awt.Dimension;

import java.awt.EventQueue;

import java.awt.Font;

import java.awt.Toolkit;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.awt.event.MouseAdapter;

import java.awt.event.MouseEvent;

import java.io.IOException;

import java.io.ObjectInputStream;

import java.io.ObjectOutputStream;

import java.net.ConnectException;

import java.net.Socket;

import java.net.SocketException;

import java.net.UnknownHostException;

import javax.swing.BorderFactory;

import javax.swing.ImageIcon;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JOptionPane;

import javax.swing.JPanel;

import javax.swing.JPasswordField;

import javax.swing.JTextPane;

import javax.swing.Timer;

import javax.swing.WindowConstants;

import javax.swing.border.Border;

import javax.swing.border.EmptyBorder;

import Admin.AdminFrame;

import CommonClasses.AdminInfo;

public class AdminLogin extends JFrame{

private static final long serialVersionUID = 1L;

private JPanel contentPane;

public boolean n;

public String username,pass;

public AdminLogin thisframe;

public static Container c;

public static Socket c2;

public static ObjectInputStream input;

public static ObjectOutputStream output;

public static AdminInfo admin;

public ImageIcon icon,icon2;

public JLabel label3;

public JPanel panel;

public ForgotPass forgotpass;

public JLabel lblNewLabel,lblNewLabel\_1,label,lblPleaseFillAll,lblPleaseFillValid;

public JTextPane textPane;

public JPasswordField password;

public JButton btnNewButton\_1,btnNewButton;

public static void main(String[] args)

{

EventQueue.invokeLater(new Runnable()

{

@Override

public void run()

{

try

{

admin=new AdminInfo();

AdminLogin frame = new AdminLogin();

frame.setVisible(true);

}catch (Exception e)

{

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the frame.

\*/

public Dimension screenSize;

private JLabel lblAdminPortal;

private JLabel label\_1;

public AdminLogin() throws UnknownHostException, IOException {

setExtendedState(java.awt.Frame.MAXIMIZED\_BOTH);

setLocationRelativeTo(null);

try

{

c2=new Socket("localhost",9783);

output=new ObjectOutputStream(c2.getOutputStream());

input=new ObjectInputStream(c2.getInputStream());

}

catch(ConnectException e)

{

JOptionPane.showMessageDialog(null,"Server Down");

System.exit(0);

}

setDefaultCloseOperation(WindowConstants.EXIT\_ON\_CLOSE);

screenSize = Toolkit.getDefaultToolkit().getScreenSize();

setBounds(0, 0, screenSize.width, screenSize.height);

setTitle("Login Portal");

icon=new ImageIcon("C:\\Users\\rajan\\Downloads\\Online-Auction-System-master\\Online-Auction-System-master\\ProjectImages2\\index.png");

setIconImage(icon.getImage());

contentPane = new JPanel();

contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

contentPane.setLayout(null);

lblAdminPortal = new JLabel("Admin Portal");

lblAdminPortal.setBounds(564, 140, 174, 26);

lblAdminPortal.setFont(new Font("Dialog", Font.BOLD, 20));

contentPane.add(lblAdminPortal);

lblAdminPortal.setForeground(Color.WHITE);

icon2=new ImageIcon("C:\\Users\\rajan\\Downloads\\Online-Auction-System-master\\Online-Auction-System-master\\ProjectImages2\\login.jpg");

panel = new JPanel();

panel.setForeground(new Color(154, 205, 50));

panel.setBackground(new Color(0, 255, 127));

panel.setBounds(340, 171, 594, 411);

panel.setLayout(null);

contentPane.add(panel);

label\_1 = new JLabel("AUCTION BASKET LOGIN");

label\_1.setForeground(new Color(255, 255, 255));

label\_1.setFont(new Font("Dialog", Font.BOLD, 25));

label\_1.setBounds(469, 83, 354, 60);

contentPane.add(label\_1);

forgotpass=new ForgotPass(this,input,output);

getContentPane().add(forgotpass);

forgotpass.setVisible(false);

lblNewLabel = new JLabel("");

lblNewLabel.setIcon(new ImageIcon("C:\\Users\\rajan\\Downloads\\Online-Auction-System-master\\Online-Auction-System-master\\ProjectImages2\\Login2.png"));

lblNewLabel.setBounds(250, 12, 100, 100);

panel.add(lblNewLabel);

textPane = new JTextPane();

textPane.setBounds(92, 124, 414, 37);

textPane.setFont(new Font("Arial", Font.PLAIN, 25));

panel.add(textPane);

lblNewLabel\_1 = new JLabel("");

lblNewLabel\_1.setIcon(new ImageIcon("C:\\Users\\rajan\\Downloads\\Online-Auction-System-master\\Online-Auction-System-master\\ProjectImages2\\User5.png"));

lblNewLabel\_1.setBounds(49, 124, 37, 37);

panel.add(lblNewLabel\_1);

password = new JPasswordField();

password.setBounds(92, 173, 414, 37);

password.setFont(new Font("Arial", Font.PLAIN, 25));

panel.add(password);

label = new JLabel("");

label.setIcon(new ImageIcon("C:\\Users\\rajan\\Downloads\\Online-Auction-System-master\\Online-Auction-System-master\\ProjectImages2\\pwd3.png"));

label.setBounds(49, 173, 37, 37);

panel.add(label);

btnNewButton\_1 = new JButton("Forgot Password?");

btnNewButton\_1.setFont(new Font("Arial", Font.BOLD, 16));

btnNewButton\_1.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent arg0) {

try {

output.reset();

} catch (SocketException e1)

{

JOptionPane.showMessageDialog(null,"Server Down");

return;

}

catch (IOException e) {

e.printStackTrace();

}

panel.setVisible(false);

forgotpass.setVisible(true);

}

});

btnNewButton\_1.addMouseListener(new MouseAdapter() {

@Override

public void mouseExited(MouseEvent e) {

btnNewButton\_1.setForeground(new Color(0, 100, 0));

}

@Override

public void mouseEntered(MouseEvent e) {

btnNewButton\_1.setForeground(new Color(77, 153, 0));

}

});

btnNewButton\_1.setForeground(new Color(0, 100, 0));

btnNewButton\_1.setBackground(new Color(0, 255, 127));

btnNewButton\_1.setFocusable(false);

btnNewButton\_1.setBorder(null);

btnNewButton\_1.setBounds(339, 305, 171, 19);

panel.add(btnNewButton\_1);

label3=new JLabel(icon2);

label3.setBounds(0, 0, screenSize.width, screenSize.height);

contentPane.add(label3);

lblPleaseFillAll = new JLabel("Please fill all the fields");

lblPleaseFillAll.setForeground(new Color(178, 34, 34));

lblPleaseFillAll.setBounds(223, 234, 161, 15);

lblPleaseFillAll.setVisible(false);

panel.add(lblPleaseFillAll);

lblPleaseFillValid = new JLabel("Please fill valid credentials");

lblPleaseFillValid.setForeground(new Color(139, 0, 0));

lblPleaseFillValid.setBounds(199, 234, 196, 15);

lblPleaseFillValid.setVisible(false);

panel.add(lblPleaseFillValid);

thisframe=this;

btnNewButton = new JButton("LOGIN");

btnNewButton.setBackground(new Color(0, 128, 0));

btnNewButton.setForeground(Color.WHITE);

btnNewButton.setBounds(92, 261, 414, 37);

btnNewButton.setFont(new Font("Arial", Font.BOLD, 17));

btnNewButton.setFocusable(false);

btnNewButton.addActionListener(new ActionListener() {

@Override

@SuppressWarnings("deprecation")

public void actionPerformed(ActionEvent arg0) {

try {

output.reset();

} catch (SocketException e1)

{

JOptionPane.showMessageDialog(null,"Server Down");

return;

}

catch (IOException e) {

e.printStackTrace();

}

if(textPane.getText().isEmpty() || password.getText().isEmpty())

{

lblPleaseFillAll.setVisible(true);

Timer t=new Timer(2000,new ActionListener()

{

@Override

public void actionPerformed(ActionEvent e)

{

lblPleaseFillAll.setVisible(false);

}

});

t.start();

t.setRepeats(false);

return;

}

username=textPane.getText();

pass=password.getText();

textPane.setText("");

password.setText("");

try

{

admin.name=username;

admin.password=pass;

output.writeObject(admin);

output.reset();

n=(boolean) input.readObject();

}

catch (Exception e)

{

e.printStackTrace();

}

if(n)

{

AdminFrame frame;

try {

frame = new AdminFrame(thisframe,input,output);

frame.setVisible(true);

thisframe.setVisible(false);

} catch (Exception e)

{

e.printStackTrace();

}

}

else

{

lblPleaseFillValid.setVisible(true);

Timer t=new Timer(2000,new ActionListener()

{

@Override

public void actionPerformed(ActionEvent e)

{

lblPleaseFillValid.setVisible(false);

}

});

t.start();

t.setRepeats(false);

return;

}

}

});

btnNewButton.addMouseListener(new MouseAdapter() {

@Override

public void mouseExited(MouseEvent e) {

btnNewButton.setBackground(new Color(0,128,0));

btnNewButton.setForeground(Color.white);

}

@Override

public void mouseEntered(MouseEvent e) {

btnNewButton.setBackground(new Color(50,205,50));

btnNewButton.setForeground(Color.white);

}

});

panel.add(btnNewButton);

Border emptyBorder = BorderFactory.createEmptyBorder();

btnNewButton.setBorder(emptyBorder);

}

}

**CustomerLogin.java**

package Login;

import java.awt.Color;

import java.awt.Container;

import java.awt.Dimension;

import java.awt.EventQueue;

import java.awt.Font;

import java.awt.Toolkit;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.awt.event.MouseAdapter;

import java.awt.event.MouseEvent;

import java.io.IOException;

import java.io.ObjectInputStream;

import java.io.ObjectOutputStream;

import java.net.ConnectException;

import java.net.Socket;

import java.net.SocketException;

import java.net.UnknownHostException;

import javax.swing.BorderFactory;

import javax.swing.ImageIcon;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JOptionPane;import javax.swing.JPanel;

import javax.swing.JPasswordField;

import javax.swing.JTextPane;

import javax.swing.Timer;

import javax.swing.border.Border;

import javax.swing.border.EmptyBorder;

import CommonClasses.CustomerInfo;

import Customer.CustomerFrame;

public class Login extends JFrame {

private static final long serialVersionUID = 1L;

private JPanel contentPane;

/\*\*

\* Launch the application.

\*/

public boolean n;

public String username,pass;

public Login thisframe;

public static Container c;

public static CustomerInfo customer;

public static Socket c2;

public static ObjectInputStream input;

public static ObjectOutputStream output;

public static void main(String[] args)

{

EventQueue.invokeLater(new Runnable()

{

public void run()

{

try

{

customer=new CustomerInfo();

Login frame = new Login();

frame.setVisible(true);

} catch (Exception e)

{

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the frame.

\*/

public JLabel label3;

public JPanel panel;

public Dimension screenSize;

public ImageIcon icon,icon2;

public JLabel label2,lblNewLabel,lblNewLabel\_1,label,lblPleaseFillAll,lblPleaseFillValid,lblAdminPortal;

public ForgotPass forgotpass;

public NewUser newuser;

public JTextPane textPane;

public JPasswordField password;

public JButton btnNewButton\_1,btnNewUser,btnNewButton;

public Login() throws UnknownHostException, IOException {

setExtendedState(java.awt.Frame.MAXIMIZED\_BOTH);

setLocationRelativeTo(null);

try {

c2=new Socket("localhost",9783);

output=new ObjectOutputStream(c2.getOutputStream());

input=new ObjectInputStream(c2.getInputStream());

}

catch(ConnectException e)

{

JOptionPane.showMessageDialog(null,"Server Down");

System.exit(0);

}

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

screenSize = Toolkit.getDefaultToolkit().getScreenSize();

setBounds(0, 0, screenSize.width, screenSize.height);

setTitle("Login Portal");

icon=new ImageIcon("C:\\Users\\rajan\\Downloads\\Online-Auction-System-master\\Online-Auction-System-master\\ProjectImages2\\index.png");

setIconImage(icon.getImage());

contentPane = new JPanel();

contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

contentPane.setLayout(null);

lblAdminPortal = new JLabel("Customer Portal");

lblAdminPortal.setBounds(538, 133, 211, 26);

lblAdminPortal.setFont(new Font("Dialog", Font.BOLD, 20));

contentPane.add(lblAdminPortal);

lblAdminPortal.setForeground(Color.WHITE);

label2=new JLabel("AUCTION BASKET LOGIN");

label2.setForeground(Color.WHITE);

label2.setFont(new Font("Dialog", Font.BOLD, 25));

label2.setBounds(469,79,354,60);

contentPane.add(label2);

icon2=new ImageIcon("C:\\Users\\rajan\\Downloads\\Online-Auction-System-master\\Online-Auction-System-master\\ProjectImages2\\login.jpg");

panel = new JPanel();

panel.setBackground(new Color(0, 255, 127));

panel.setBounds(338, 171, 594, 411);

panel.setLayout(null);

contentPane.add(panel);

forgotpass=new ForgotPass(this,input,output);

getContentPane().add(forgotpass);

forgotpass.setVisible(false);

newuser=new NewUser(this,input,output);

getContentPane().add(newuser);

newuser.setVisible(false);

lblNewLabel = new JLabel("");

lblNewLabel.setIcon(new ImageIcon("C:\\Users\\rajan\\Downloads\\Online-Auction-System-master\\Online-Auction-System-master\\ProjectImages2\\Login2.png"));

lblNewLabel.setBounds(250, 12, 100, 100);

panel.add(lblNewLabel);

textPane = new JTextPane();

textPane.setBounds(92, 124, 414, 37);

textPane.setFont(new Font("Arial", Font.PLAIN, 25));

panel.add(textPane);

lblNewLabel\_1 = new JLabel("");

lblNewLabel\_1.setIcon(new ImageIcon("C:\\Users\\rajan\\Downloads\\Online-Auction-System-master\\Online-Auction-System-master\\ProjectImages2\\User5.png"));

lblNewLabel\_1.setBounds(49, 124, 37, 37);

panel.add(lblNewLabel\_1);

password = new JPasswordField();

password.setBounds(92, 173, 414, 37);

password.setFont(new Font("Arial", Font.PLAIN, 25));

panel.add(password);

label = new JLabel("");

label.setIcon(new ImageIcon("C:\\Users\\rajan\\Downloads\\Online-Auction-System-master\\Online-Auction-System-master\\ProjectImages2\\pwd3.png"));

label.setBounds(49, 173, 37, 37);

panel.add(label);

btnNewButton\_1 = new JButton("Forgot Password?");

btnNewButton\_1.setFont(new Font("Arial", Font.BOLD, 16));

btnNewButton\_1.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent arg0) {

try {

output.reset();

} catch (SocketException e1)

{

JOptionPane.showMessageDialog(null,"Server Down");

return;

}

catch (IOException e) {

e.printStackTrace();

}

panel.setVisible(false);

forgotpass.setVisible(true);

}

});

btnNewButton\_1.setBorder(null);

btnNewButton\_1.setBounds(87, 310, 171, 25);

btnNewButton\_1.addMouseListener(new MouseAdapter() {

public void mouseExited(MouseEvent e) {

btnNewButton\_1.setForeground(new Color(0, 100, 0));

}

public void mouseEntered(MouseEvent e) {

btnNewButton\_1.setForeground(new Color(77, 153, 0));

}

});

btnNewButton\_1.setForeground(new Color(0, 100, 0));

btnNewButton\_1.setBackground(new Color(0, 255, 127));

btnNewButton\_1.setFocusable(false);

panel.add(btnNewButton\_1);

thisframe=this;

btnNewUser = new JButton("New User");

btnNewUser.setFont(new Font("Arial", Font.BOLD, 16));

btnNewUser.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

try

{

output.writeObject("Check");

output.reset();

Object obj=input.readObject();

if(obj instanceof Boolean)

{

n=(boolean)obj;

if(!n)

{

JOptionPane.showMessageDialog(thisframe,"Addmission Process is Closed");

return;

}

}

} catch (SocketException e1)

{

JOptionPane.showMessageDialog(null,"Server Down");

return;

}

catch (Exception e1)

{

e1.printStackTrace();

}

panel.setVisible(false);

newuser.setVisible(true);

}

});

btnNewUser.setFocusable(false);

btnNewUser.setBorder(null);

btnNewUser.setBounds(413, 310, 94, 15);

btnNewUser.addMouseListener(new MouseAdapter() {

public void mouseExited(MouseEvent e) {

btnNewUser.setForeground(new Color(0, 100, 0));

}

public void mouseEntered(MouseEvent e) {

btnNewUser.setForeground(new Color(77, 153, 0));

}

});

btnNewUser.setForeground(new Color(0, 100, 0));

btnNewUser.setBackground(new Color(0, 255, 127));

panel.add(btnNewUser);

label3=new JLabel(icon2);

label3.setBounds(0, 0, screenSize.width, screenSize.height);

contentPane.add(label3);

lblPleaseFillAll = new JLabel("Please fill all the fields");

lblPleaseFillAll.setForeground(new Color(178, 34, 34));

lblPleaseFillAll.setBounds(223, 234, 161, 15);

lblPleaseFillAll.setVisible(false);

panel.add(lblPleaseFillAll);

lblPleaseFillValid = new JLabel("Please fill valid credentials");

lblPleaseFillValid.setForeground(new Color(178, 34, 34));

lblPleaseFillValid.setBounds(199, 234, 196, 15);

lblPleaseFillValid.setVisible(false);

panel.add(lblPleaseFillValid);

btnNewButton = new JButton("LOGIN");

btnNewButton.setBackground(new Color(0, 128, 0));

btnNewButton.setForeground(Color.WHITE);

btnNewButton.setBounds(92, 261, 414, 37);

btnNewButton.setFont(new Font("Arial", Font.BOLD, 17));

btnNewButton.setFocusable(false);

btnNewButton.addActionListener(new ActionListener() {

@SuppressWarnings("deprecation")

public void actionPerformed(ActionEvent arg0) {

try {

output.reset();

} catch (SocketException e1)

{

JOptionPane.showMessageDialog(null,"Server Down");

return;

}

catch (IOException e) {

e.printStackTrace();

}

if(textPane.getText().isEmpty() || password.getText().isEmpty())

{

lblPleaseFillAll.setVisible(true);

Timer t=new Timer(2000,new ActionListener()

{

public void actionPerformed(ActionEvent e)

{

lblPleaseFillAll.setVisible(false);

}

});

t.start();

t.setRepeats(false);

return;

}

username=textPane.getText();

pass=password.getText();

textPane.setText("");

password.setText("");

try

{

customer.name=username;

customer.password=pass;

output.writeObject(customer);

output.reset();

n=(boolean) input.readObject();

}

catch (Exception e)

{

e.printStackTrace();

}

if(n)

{

try {

CustomerFrame frame = new CustomerFrame(thisframe,input,output);

frame.setVisible(true);

thisframe.setVisible(false);

} catch (Exception e)

{

e.printStackTrace();

}

}

else

{

lblPleaseFillValid.setVisible(true);

Timer t=new Timer(2000,new ActionListener()

{

public void actionPerformed(ActionEvent e)

{

lblPleaseFillValid.setVisible(false);

}

});

t.start();

t.setRepeats(false);

return;

}

}

});

btnNewButton.addMouseListener(new MouseAdapter() {

public void mouseExited(MouseEvent e) {

btnNewButton.setBackground(new Color(0,128,0));

btnNewButton.setForeground(Color.white);

}

public void mouseEntered(MouseEvent e) {

btnNewButton.setBackground(new Color(50,205,50));

btnNewButton.setForeground(Color.white);

}

});

panel.add(btnNewButton);

Border emptyBorder = BorderFactory.createEmptyBorder();

btnNewButton.setBorder(emptyBorder);

}

}

ShowItems.java

package Admin;

import java.awt.Color;

import java.awt.Component;

import java.awt.Font;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.io.File;

import java.io.FileOutputStream;

import java.io.ObjectInputStream;

import java.io.ObjectOutputStream;

import java.util.HashMap;

import java.util.Iterator;

import java.util.Vector;

import javax.swing.ImageIcon;

import javax.swing.JButton;

import javax.swing.JLabel;

import javax.swing.JPanel;

import javax.swing.JScrollPane;

import javax.swing.JTable;

import javax.swing.Timer;

import javax.swing.table.DefaultTableModel;

import javax.swing.table.TableCellRenderer;

import javax.swing.table.TableColumn;

import CommonClasses.\*;

public class ShowItems extends JPanel {

private static final long serialVersionUID = 1L;

/\*\*

\* Create the panel.

\*/

public Item item;

public int row,rowcount;

public DefaultTableModel model;

public JTable table;

public JScrollPane j1;

private JLabel label,label2,lblPleaseSelectThe;

public byte Barray[]=new byte[120000]; //120 KBytes

public FileOutputStream fos;

public ImageIcon image;

public ShowItems(ObjectInputStream input,ObjectOutputStream output) throws Exception {

setLayout(null);

setBounds(309,180,980,564);

model=new DefaultTableModel();

table = new JTable(model){

private static final long serialVersionUID = 1L;

@Override

public boolean isCellEditable(int row, int column) {

return false;

};

};

model.addColumn("Item ID");

model.addColumn("Item Image");

model.addColumn("Item Name");

model.addColumn("Owner Name");

model.addColumn("Reserve Price");

table.getTableHeader().setFont(new Font("SansSerif", Font.BOLD, 15));

table.setFont(new Font("Arial", Font.LAYOUT\_LEFT\_TO\_RIGHT, 12));

table.setRowHeight(20);

j1=new JScrollPane(table);

j1.setBounds(70,71,830,420);

add(j1);

table.getColumn("Item Image").setCellRenderer(new LabelRenderer());

label = new JLabel("List of Items for Auction");

label.setForeground(Color.BLUE);

label.setFont(new Font("Dialog", Font.PLAIN, 25));

label.setBounds(325, 24, 308, 30);

add(label);

lblPleaseSelectThe = new JLabel("Please select the row to delete.");

lblPleaseSelectThe.setForeground(Color.RED);

lblPleaseSelectThe.setBounds(70, 503, 253, 15);

lblPleaseSelectThe.setVisible(false);

add(lblPleaseSelectThe);

JButton btnDeleteRow = new JButton("Delete Row");

btnDeleteRow.addActionListener(new ActionListener() {

@Override

@SuppressWarnings("unchecked")

public void actionPerformed(ActionEvent arg0) {

row=table.getSelectedRow();

rowcount=table.getRowCount();

if(row < 0 || row >= rowcount)

{

lblPleaseSelectThe.setVisible(true);

Timer t=new Timer(2000,new ActionListener()

{

@Override

public void actionPerformed(ActionEvent e)

{

lblPleaseSelectThe.setVisible(false);

}

});

t.start();

t.setRepeats(false);

return;

}

try

{

output.writeObject(table.getValueAt(row,0));

model.removeRow(row);

output.reset();

output.writeObject("ItemList");

output.reset();

ServerCollection.ItemList=(Vector<Item>) input.readObject();

row=-1;

} catch (Exception e2)

{

return; // return from ActionPerformed

}

}

});

btnDeleteRow.setBounds(720, 515, 124, 37);

add(btnDeleteRow);

Iterator<Item> i=ServerCollection.ItemList.iterator();

ServerCollection.items=new HashMap<Integer,Item>();

ServerCollection.Images=new HashMap<Integer,ImageIcon>();

while(i.hasNext())

{

item=i.next();

Barray=item.b;

File file=new File("C:\\Users\\rajan\\Downloads\\Online-Auction-System-master\\Online-Auction-System-master\\Admin2\\AdminImages\\"+String.valueOf(item.ID)+".txt");

fos=new FileOutputStream(file);

fos.write(Barray);

image=new ImageIcon("C:\\Users\\rajan\\Downloads\\Online-Auction-System-master\\Online-Auction-System-master\\Admin2\\AdminImages\\"+String.valueOf(item.ID)+".txt");

label2=new JLabel(image);

ServerCollection.items.put(item.ID,item);

ServerCollection.Images.put(item.ID,image);

model.addRow(new Object[]{item.ID,label2,item.name,item.Owner\_name,item.Reserve\_Price});

}

}

class LabelRenderer implements TableCellRenderer

{

@Override

public Component getTableCellRendererComponent(JTable table, Object value, boolean isSelected, boolean hasFocus, int row,

int column) {

TableColumn tc=table.getColumn("Item Image");

tc.setMinWidth(200);

table.setRowHeight(200);

return (Component)value;

}

}

public void refresh() throws Exception

{

int rows = model.getRowCount();

for(int i = rows - 1; i >=0; i--)

{

model.removeRow(i);

}

Iterator<Item> i=ServerCollection.ItemList.iterator();

ServerCollection.items.clear();

ServerCollection.Images.clear();

while(i.hasNext())

{

item=i.next();

Barray=item.b;

File file=new File("C:\\Users\\rajan\\Downloads\\Online-Auction-System-master\\Online-Auction-System-master\\Admin2\\AdminImages\\"+String.valueOf(item.ID)+".txt");

fos=new FileOutputStream(file);

fos.write(Barray);

image=new ImageIcon("C:\\Users\\rajan\\Downloads\\Online-Auction-System-master\\Online-Auction-System-master\\Admin2\\AdminImages\\"+String.valueOf(item.ID)+".txt");

label2=new JLabel(image);

ServerCollection.items.put(item.ID,item);

ServerCollection.Images.put(item.ID,image);

model.addRow(new Object[]{item.ID,label2,item.name,item.Owner\_name,item.Reserve\_Price});

}

}

}

**Future Scope**

One of the main future enhancements of our system is to make it functioning with online transactions and interactions between buyer and sellers. We would also like to improve the security by adding the option of blacklisting defaulting bidders. There also can be options for rating sellers and products. Online payment settlement can be incorporated in the system. Real time chat boxes where sellers and customers can discuss with each other as well as public chat boxes can be incorporated in the system.

**Conclusion**

The developed system is flexible and changes can be made easily. The system is developed with an insight of the necessary modification that may be required in the future. Hence the system can be maintained successfully without much rework.

**Reference**

* YouTube
* GitHub
* **Dr. Nilima Kulkarni**
* Google

**THANK YOU**