INDEX

Sr. No. Title's of Contents Page No.

1. Content of INDEX 1
2. Introduction 2 - 7
3. Objective 8 - 10
4. Front-End & Back-End 11 - 13
5. Hardware/ Software Requirement 14 - 15
6. Data Flow Diagram and ER-Model 16 - 22
7. Modules/ Forms & Reports 23 - 24
8. Database Table Description 24 - 27

12. Form Design (Output Window) with coding 28 - 139

13. Conclusion 140 - 141

14. Bibliography 142 - 143

**INTRODUCTION**

**INTRODUCTION**

The Supermarket Management System is a project that deals with supermarket automation and it includes both purchasing a selling of items. This project is designed with a goal to making the existing system more informative, reliable, fast and easier. There are many reasons for the starting of the project because in the selling of items through the manual system of salesperson faces a lot of inefficiencies. It requires handling of large record books that consist of both irrelevant and important information’s thus making it difficult to find out the required information as per necessity.

This is also a clumsy and untidy process that disturbs the smooth flow of work. But this system introduced by us will reduce the huge number of paper works while on the other side there are many more problems that persist in the manual system. They reduce efficiency as well as the productivity level of human. Administrative module, purchase module, employee module, sales module and billing module are some of the various modules that make up our project.

This is one of the best system that can be introduced in the supermarket for efficient management. It will also reduce the cumbersome job of finding the most accurate data from the huge log books. It also helps the management to keep efficient records of all the purchases and sales. The introduction of ID and password will further reduce the manipulation and thus providing the accurate and transparent data. This system will increase the productivity and reduce the need of manual system to a large extent.

The administrators consist of a unique password and names of the employees. It helps the employees to make secure login. The ids and passwords are kept secret from others. The modules of sales and purchase include all the details of selling and purchasing. In the billing module the details of payments are clearly shown.

The system will display all the items whose name start with the letter selected by the user. He can select out of those displayed. Finally a separate bill will be generated for each customer. This will be saved in the database. Any periodic records can be viewed at any time. If the stock is not available, the supermarket orders and buys from a prescribed vendor. The amount will be paid by deducting the total amount acquired in the sales activity.

**PURPOSE OF THE PROJECT**

In the general Super market management system, a super market owner can faced many of problems. Some of these problems can be managed easily, but some of them have many challenging

Some of these challenges in the general Library management System can be figure-out

as:-

* A person entering the shopping mall does not know what are the items located in the certain place to purchase the shopping mall.
* He/She has to look for particular item through the shopping mall.
* Manual system faces a lot of inefficiencies.
* Interruption in smooth flow of work.
* Lack of security of data.
* Although more man power but less efficiency.
* Time consuming.
* Consume large volume of paper work.
* Need manually calculation which are facing error.
* Poor communication may lead to serious errors.

So for leaving this problem we are developed a project titled “Super Market Management System”

**SCOPE & FEATURES APPLICATION**

The scope of future application for this system is:

1. A large customer and sale database can be accommodated in our application in the future.
2. The related aspects like product details, order details and customer data can be maintained more easily.
3. In the purposed application examination monitoring, so that in future an increased number of customer can be accommodated by the system.
4. Super Market Management System currently takes care of sales data.
5. This remove burden of customer.
6. The format of different types of reports can be suit the requirements in future though major modifications will be required but the base of the project will remain the same and all the working will remain same in this project.
7. Payment also can be done through mobile or computer.

**OVERVIEW OF THIS PROJECT**

1. To provide security as only an authorized employee can interact with the system.
2. To help the staff in working easily and efficiently.
3. To help the staff in reducing errors that are encountered frequently during manual operations by concurrently updating the data stored in many places.
4. To reduce the workload on the staff by providing automatic calculation.
5. To help the staff or manager in tracing the transaction quickly.
6. To help the staff in generating various reports at various nick of time as per their requirements e.g. daily, weekly, monthly or annually.
7. To reduce the manpower. 8. To maintain the data with complete consistency and minimum redundancy

**OBJECTIVE**

**OBJECTIVE**

**It is a live project which deals with a fully Shopping System for daily using things in human life. During the course of the development of this project a lot of thinking went into defining the different modules and the procedures used and a lot of common discrepancies have been taken care of.**

**GARMENTS:** Women, men, girls, boys, baby

**HOME FURNISING:** Bedroom, drawing, kitchen, bathroom, covers.

**FOODMAR:** Indian cooked, Chinese cooked, fruits, vegetable.

**HOUSE HOLD:** Thermo ware, cleaning aids, plastic aids, kitchen, ware, etc.

**OUR PRODUCST:**

* **Bedroom :** pillows, pillows cover, bed sheet
* **Drawing room:** door mat curtain, carpet
* **Bathroom:** soap, towel, towel gift set
* **Kitchen:** apron, kitchen napkin

**FOOD & BEVERAGES:**

* Snakes
* Drink

**SPORTS & FITNESS:**

* **INDOOR GAMES**

Sports kit, chess board, bawling. TT bat, boxing kit

* **OUTDOOR GAMES**

Football, badminton, tennis, cricket.

* **FOOTWEAR**

**WOMEN MEN**

Shoes Shoes

Sandal Sandal

Slipper Slipper

**Communication:**

* **MOBILE:**

Cover, sets, battery, charger.

**MEN'SGARMENTS:**

Kurta, shirt, pants, jeans, T-shirts, trouser, blazers.

* **WOMEN'S GARMENTS**:

Sari, Shirt top , Dupatta , Jeans top , Night suit, Jacket.

* **KIDS GARMENTS:**:

Hot Pant, Night Suit, Shirt, Tops, Hipster Set, Frock

**TRAVEL ACCEORIES:**

Suitcase, Shoulder Bags, Executive Bags, School Bags

**HOUSEHOLD:**

Dinner set, Jug, Cake saver, Cooker, Bruch, Plastic goods, Glass, Tea set, Tiffin etc.

**OTHER DAILY USE THINGS:**

Shaving cream, Cream, Wrist watch, Sun glass, Key chain, Belt

Bracelet, Spray, Deo, Perfume

**TOYS:**

Musical, nonmusical

**FRONTEND/BACKEND**

**FRONT-END**

Notepad++ is a free (as in “free speech” and also as in “free beer”) source code editor and Notepad replacement that supports several languages. Running in the MS Windows environment, its use is governed by [GNU General Public License](https://www.gnu.org/licenses/old-licenses/gpl-2.0.html).

Based on the powerful editing component [Scintilla](https://www.scintilla.org/), Notepad++ is written in C++ and uses pure Win32 API and STL which ensures a higher execution speed and smaller program size Notepad++ is a free text and source code editor for Windows. You can use Notepad++ to write programs in Java and other programming languages. Using a plugin called "NppExec", you can compile and run Java programs using Notepad++. You'll need to have the proper Windows environment variables set up on your computer in order to do so. You will also need to write a short script to compile and run Java programs using the plugin. It’s easy to set to your system variables and compile and run Java programs in Notepad++.

**Feature of Notepad++**

* Language
* [Autosave](https://en.wikipedia.org/wiki/Autosave)
* Finding and replacing strings of text with [regular expressions](https://en.wikipedia.org/wiki/Regular_expression)
* Searching text strings within opened tabs
* Searching text strings in a directory
* Guided indentation
* Line bookmarking
* Controls
* Data access

Use Notepad++ to enter your Java code. It works like a regular Notepad, but also **has predictive text and autofill features for most programming languages**. This makes it much easier to write code in Notepad++ rather than regular Notepad.

**BACK-END**

A database is a collection of information that's related to a particular subject or purpose, such as tracking customer orders or maintaining a product collection .If your database isn't stored on a computer, or only parts of it are, you may be tracking information from a variety of sources that you have having to coordinate and organize yourself.

Using Microsoft Access, you can manage all your information from a single database file. Within the file, you can use:

* Tables to store your data.
* Queries to find and retrieve just the data you want
* Forms to view add and update data in tables.
* Reports to analyze or print data in a specific layout.
* Data access pages to view, update, or analyze the database's data from the internet or an intranet.

**HARDWARE SOFTWARE REQUIREMENT**

**HARWARE/SOFTWARE**

**REQURIMENT**

This software runs within any environment. This software needs at least Pentium processor 32MB RAM memory, 2.1GB of Hard disk ,SVGA display of 1 MB video memory and color monitor but with this lot of processor power consumed and speed of the system will be very slow. Therefore a realistic hardware and software configuration for this software is:

**HARDWARE REQUIREMENT’S**

Processor : Core i3 or higher processor

Display : VGA card

Keyboard : 104 keys

Ram : Memory 4GB

Monitor : LED Screen or high resolution.

**SOFTWARE REQUIREMENT'S**

Operating System : Microsoft Windows 7 or above

Notepad++ : 6.2.3 or above

JDK : Java SE 9 or above

Microsoft Access : 2007 version

**DATA FLOW DIAGRAM**

**(DFD)**

**DATA FLOW DIAGRAME**

A Data Flow Diagram (DFD) is a graphical tool used to describe and analyze the movement of data through a system - manual or automated including the processes, stores of data and delays in the system. They are central tools and the basis from which other components are developed. It depicts the transformation of data, independent of the physical components, from input to output through logical processes and the interaction between processes.

DFD's are an excellent mechanism of communicating with the customers and are widely used for representing external and top-level internal design specification. In the later situations, DFD's are quite valuable for establishing naming conventions and names of system components such as subsystems, files and data links.

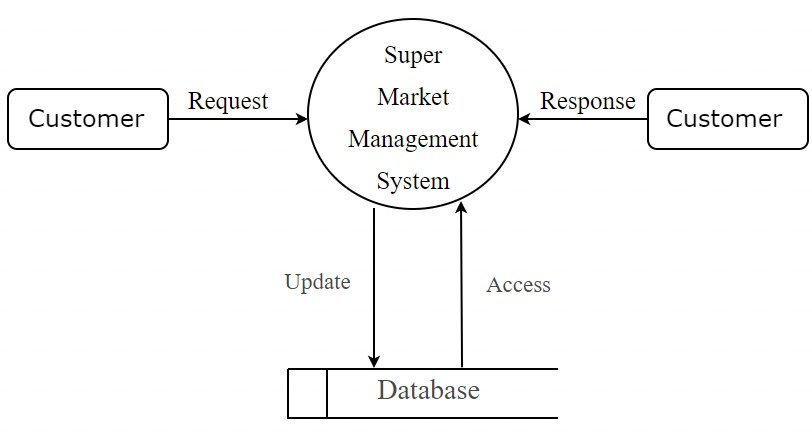
In a DFD there are four components: Sources or Destinations of data such as human, entities that interact with system, outside the system boundary, who form the source and the recipient of information are depicted in the form of a closed rectangle.

Data flow is a packet of data. It identifies data flow. It is a pipeline through which information flows. It is depicted in DFD as an arrow with the pointer pointing in the direction of flow. This connecting symbol connects an entity, process and data stores. This arrow mark also specifies the sender and the receiver.

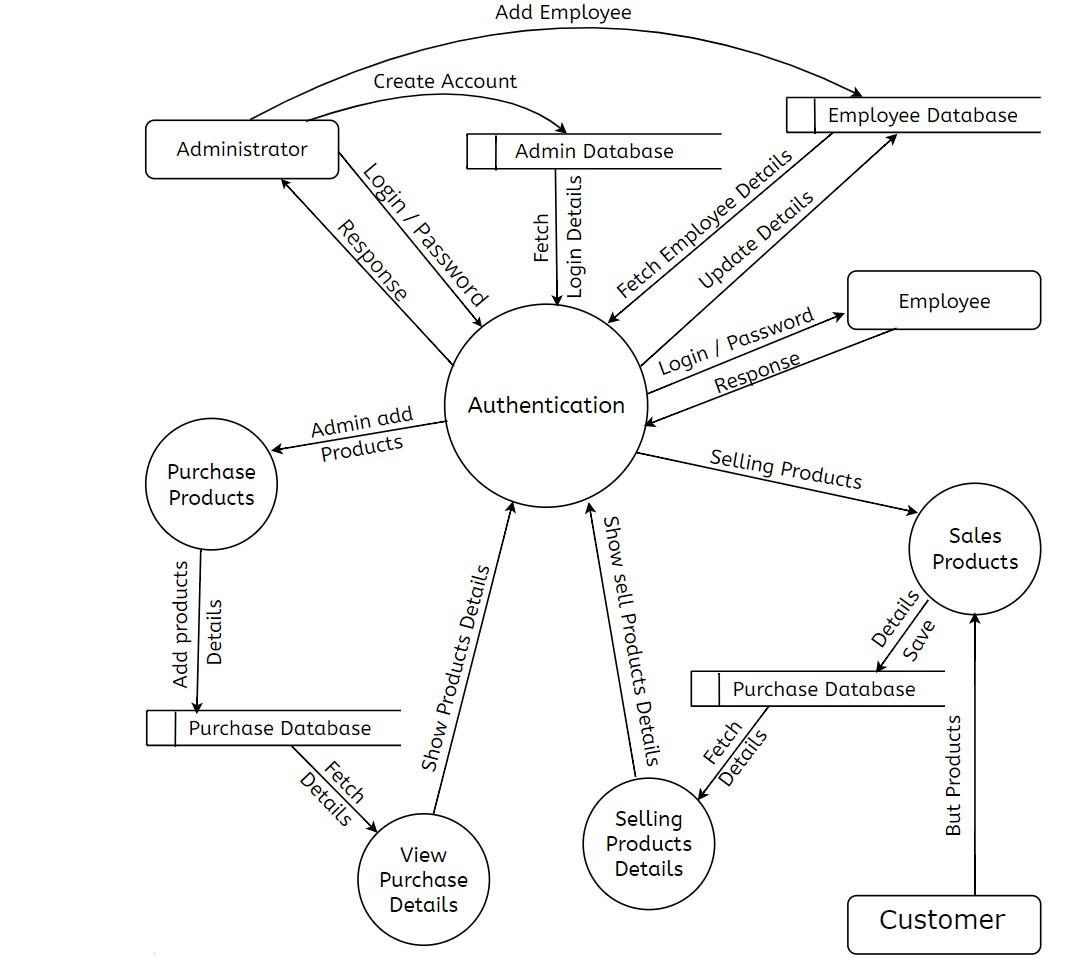
Process depicts procedure, function or module that transforms input data into output data. It is represented as a circle or a bubble with the procedure name and a unique number inside the circle.

Data stores are the physical areas in the computer’s hard disk where groups of related data are stored in the form of files. They are depicted as an open-ended rectangle. The data store is used either for storing data into the files or for reference purpose.

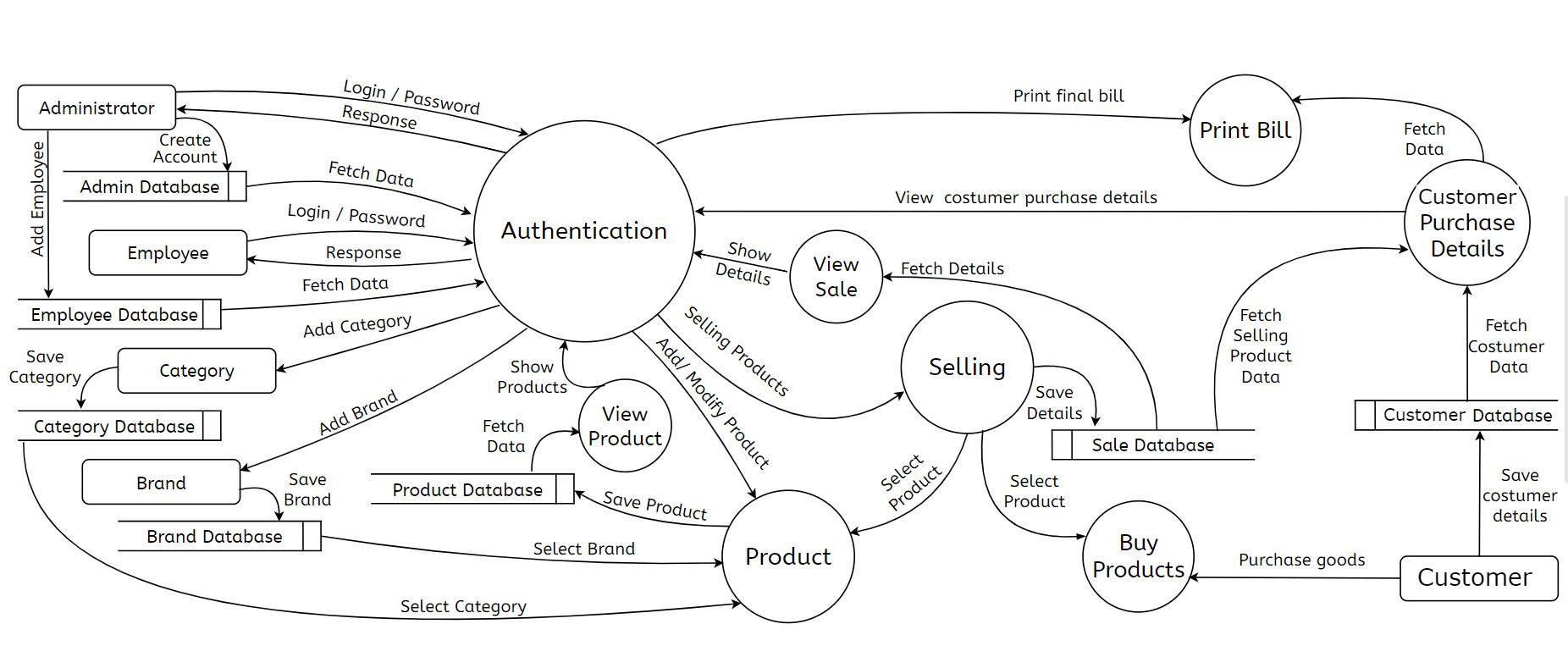
**ZERO- LEVEL DFD**



**FIRST- LEVEL DFD**



**SECOND- LEVEL DFD**

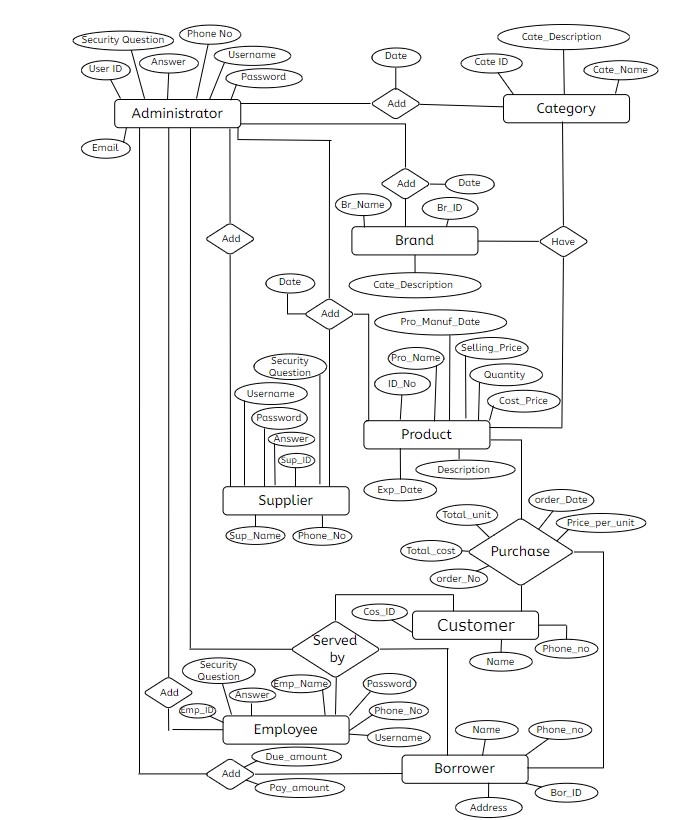
**ENTITY - RELATIONSHIP DIAGRAM**

The entity – relationship (E – R) data model is based on a perception of a real world that consists of a collection of basic objects, called entities, and of relationships among these objects. An entity is a thing or objects in the real world that is distinguishable from other objects. Entities are described in database by a set of attributes.

A relationship is an association among several entities. The set of all entities of the same type and the set of all relationships of the same type are termed as entity set and relationship set, respectively.

The overall logical structure (schema) of a database can be expressed graphically by an E-R Diagram, which is built up from the following components:

* + Rectangles, which represents entity sets.
  + Ellipses, which represents attributes.
  + Diamonds, which represents relationships among entity sets.
  + Lines, which link attributes to entity sets and entity sets to relationships.

**ER DIAGRAM**

**OPERTATION OF PRESENT SYSTM**

The SUPER MARKET MANAGEMENT SYSTEM is further integration of various sub - system. Broadly speaking these sub – system are based on following broadly classified activities.

1. Selling Management

* Add customer name & number.
* Select Products and add into the table.
* Add its quantities.
* Select the payment method.
* Print the bill.

1. Employee Management

* Add Employees.
* Edit employee details.
* Delete employees.
* Search employee

1. Product Management

* Add Products
* Modify products details.
* Delete products .
* Show product report.

1. Category or Brand Management

* Add category or brand.
* Edit category or brand details.
* Delete category or brand
* Select category or brand in product selection.

1. Borrower Management.

* Add borrower.
* Modify borrower details.
* Set notification on borrower payment amount.
* Delete borrower.

DESIGN PHASE

DATABASE TABLE DESCRIPTION

All the tables, sequence, triggers are stored in database for storing the information about entrance test mode by the users. Following are the tables:

Administrator Table

|  |  |
| --- | --- |
| User\_ID | Number |
| Username | Text |
| Password | Text |
| Business Name | Text |
| Address | Text |
| Security Question | Text |
| Answer | Text |
| First Name | Text |
| Last Name | Text |
| Email | Text |
| Contact Number | Text |
| City | Text |
| Pincode | Number |
| State | Text |
| Country | Text |
| PAN No. | Text |
| GSTN No | Text |

Employee Table

|  |  |
| --- | --- |
| Emp\_ID | Number |
| Name | Text |
| Username | Text |
| Password | Text |
| Security Question | Text |
| Answer | Text |
| ID Proof | Text |

Category Table

|  |  |
| --- | --- |
| Category\_ID | Number |
| Category\_Name | Text |
| Category\_Description | Text |

Brand Table

|  |  |
| --- | --- |
| Brand\_ID | Number |
| Brand\_Name | Text |
| Brand\_Description | Text |

Purchase Table

|  |  |
| --- | --- |
| Pro\_ID | Number |
| Pro\_Code | Text |
| Pro\_Name | Text |
| Pro\_Price | Number |
| Pro\_\_Unit | Number |
| Pro\_Manufacture\_Date | Date \ Time |
| Pro\_Expire\_Date | Date \ Time |
| Pro\_Brand | Text |
| Pro\_Category | Text |

Customer Table

|  |  |
| --- | --- |
| Customer\_Name | Text |
| Customer\_Contact\_Number | Text |
| Order\_ID | Number |
| Order\_Date | Date \ Time |

Selling Table

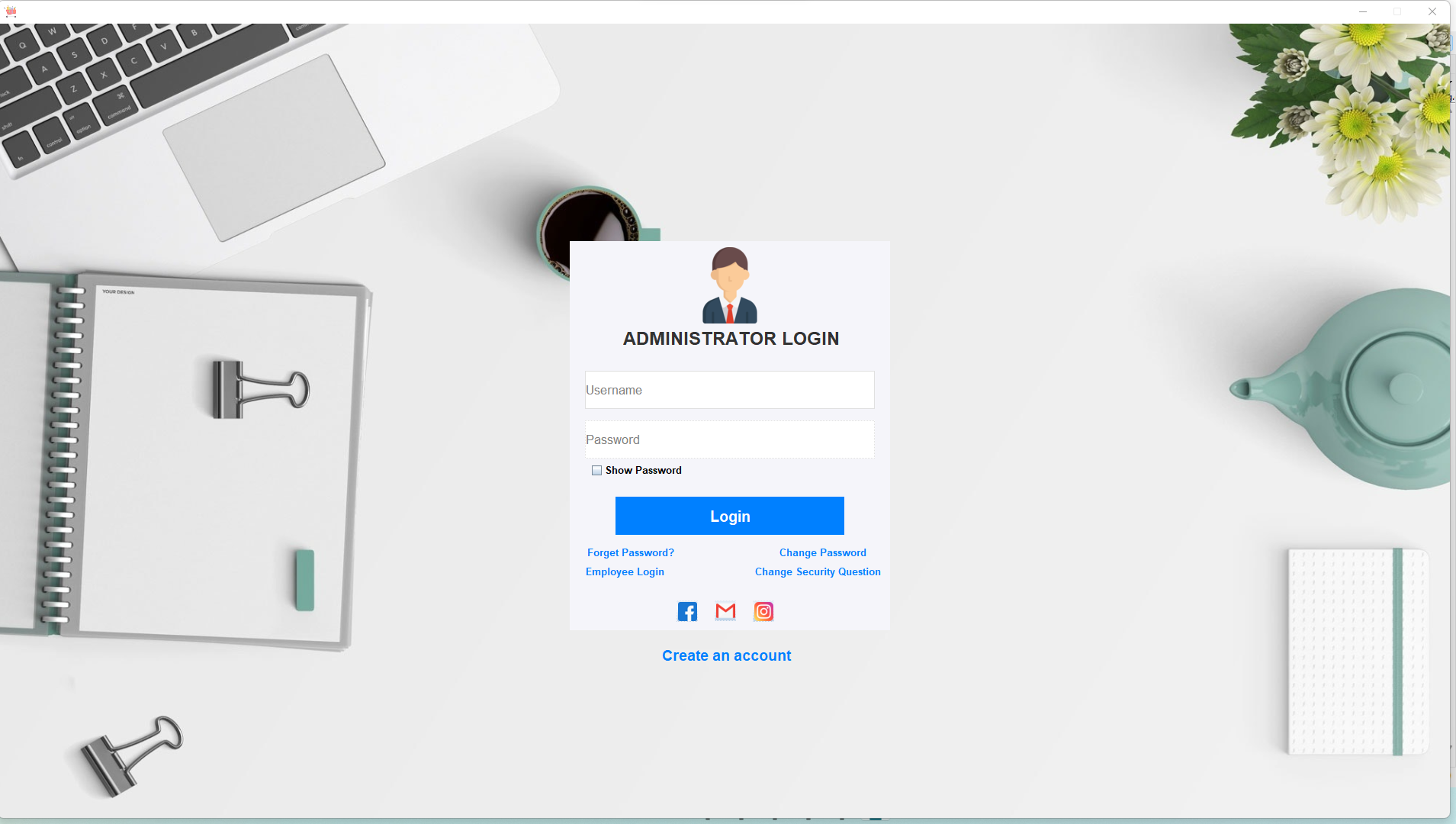
|  |  |
| --- | --- |
| Sal\_ID | Number |
| Sal\_Product\_Name | Text |
| Sal\_Unit | Number |
| Sal\_Price | Number |
| Sal\_Date | Date \ Time |
| Sal\_Product\_Brand | Text |
| Sale\_Product\_Category | Text |
| Sal\_Product\_Status | Text |
| Sal\_Product\_Payment | Text |

OUTPUT

SCREENS

FORM DESIGN PHASE

LOGIN FORM OUTPUT WINDOW SCREEN



**Output form information**

This form is used to login the main page of super market management system.

**Coding of Login Form**

**import** java.awt.\*;

**import** java.awt.event.\*;

**import** javax.swing.\*;

**import** java.sql.\*;

**public** **class** Login {

JFrame frame;

JPanel panel;

Login()

{

frame =**new** JFrame(); //Main Frame Object

Imageicon=Toolkit.*getDefaultToolkit*().getImage("C:\\Users\\AHsanUsmani\\Desktop\\Programs\\Projet GMS\\src\\su1.png");

frame.setIconImage(icon);

// For gettting Screen Width and Height

Toolkit tk=Toolkit.*getDefaultToolkit*();

**int** width=(**int**) tk.getScreenSize().getWidth();

**int** height=(**int**) tk.getScreenSize().getHeight();

ImageIcon adImg = **new** ImageIcon("C:\\Users\\AHsan Usmani\\Desktop\\Programs\\Project GMS\\src\\man1.png");

JLabel adminImage=**new** JLabel(adImg,JLabel.***CENTER***

adminImage.setBounds(910,293,100,100);

frame.add(adminImage);

JLabel adlabel =**new** JLabel("ADMINISTRATOR LOGIN

Font font=**new** Font("LucidaGrande",Font.***PLAIN*** + Font.***BOLD***,24);

adlabel.setFont(font);

adlabel.setBounds(820,395,350,35);

frame.add(adlabel);

JTextField username=**new** JTextField();

username.setText("Username");

Font fontTF=**new** Font("LucidaGrande",Font.***PLAIN*** ,17);

username.setFont(fontTF);

username.setBorder(BorderFactory.*createLineBorder*(**new** Color(224,224,224 username.setForeground(Color.***gray***); username.setBounds(770,455,380,50);

username.addKeyListener(**new** KeyAdapter() {

@Override

**public** **void** keyTyped(KeyEvent e) {

**if**(username.getText().equals("Username"))

username.setText(""); }

});

username.addMouseListener(**new** MouseAdapter(){

@Override

**public** **void** mouseClicked(MouseEvent e){

username.setText("");

}

});

username.addFocusListener(**new** FocusListener() {

@Override

**public** **void** focusGained(FocusEvent e) {}

@Override

**public** **void** focusLost(FocusEvent e) {

**if**(username.getText().length()==0) {

username.setText("Username"); }

}

});

frame.add(username);

JPasswordField password = **new** JPasswordField(); password.setFont(fontTF); password.setBorder(BorderFactory.*createDashedBorder*(**new** Color(224,224,224))); password.setText("Password"); password.setForeground(Color.***gray***);

password.setBounds(770,520,380,50); password.setEchoChar((**char**)0 );

password.addKeyListener(**new** KeyAdapter() {

@Override

**public** **void** keyTyped(KeyEvent e) {

**if**(password.getText().equals("Password")) {

password.setText("");

password.setEchoChar('\*'); }

}

});

password.addFocusListener(**new** FocusListener() {

@Override

**public** **void** focusGained(FocusEvent e) { }

@Override

**public** **void** focusLost(FocusEvent e) { **if**(password.getText().length()==0) {

password.setEchoChar((**char**)0);

password.setText("Password"); }

}

});

frame.add(password);

JCheckBox shwPas=**new** JCheckBox("Show Password"); shwPas.setForeground(Color.***black***); shwPas.setBackground(**new** Color(245,245,250)); shwPas.setFont(**new** Font("LucidaGrande",Font.***BOLD***,13));

shwPas.setBounds(775,575,145,20); frame.add(shwPas);

shwPas.addActionListener(**new** ActionListener() {

**public** **void** actionPerformed(ActionEvent e) {

**if**(shwPas.isSelected()) {

password.setEchoChar((**char**)0); }

**else**

{password.setEchoChar('\*');}

}

});

JButton LoginB=**new** JButton("Login");

LoginB.setForeground(Color.***white***);

LoginB.setFont(**new** Font("LucidaGrande",Font.***BOLD***,20) ); LoginB.setBorder(BorderFactory.*createDashedBorder*(**new** Color(0,128,255)));

LoginB.setBounds(810,620,300,50); LoginB.setBackground(**new** Color(0,128,255)); frame.add(LoginB);

LoginB.addActionListener(**new** ActionListener() {

@Override

**public** **void** actionPerformed(ActionEvent e) {

**if**(e.getSource()==LoginB) {

**if**((username.getText().equals("") || username.getText().equals("Username") )&& (password.~~getText~~().equals("") || password.~~getText~~().equals("Password")))

{

ImageIcon ig1=**new** ImageIcon("C:\\Users\\AHsan Usmani\\Desktop\\Programs\\Project GMS\\src\\error.png");

JOptionPane.*showMessageDialog*(**null**,"Please enter usernamepassword","Warrning",JOptionPane.***WARNING\_MESSAGE***,ig1);

}

else if (password. getText(). equals("") || password. getText(). equals("Password")) {

ImageIcon ig1 = new ImageIcon(

"C: \\Users\\AHsan Usmani\\Desktop\\Programs\\Project GMS\\src\\error. png");

JOptionPane. showMessageDialog(null, "Please enter a password", "Warrning",

JOptionPane. WARNING\_MESSAGE, ig1);

}

else {

try {

Connection con = DriverManager. getConnection(

"jdbc: ucanaccess: //C: \\Users\\AHsan Usmani\\Desktop\\Programs\\Project GMS\\src\\sms. accdb");

Statement st = con. createStatement();

ResultSet rst = st. executeQuery("Select \* from userTable");

if (rst. next()) {

if (username. getText(). equals(rst. getString(1))

&& password. getText(). equals(rst. getString(2))) {

WelcomePage obj = new WelcomePage();

}

else {

ImageIcon ig1 = new ImageIcon(

"C: \\Users\\AHsan Usmani\\Desktop\\Programs\\Project GMS\\src\\error. png");

JOptionPane. showMessageDialog(null, "Wrong Username or password", "Warrning",

JOptionPane. WARNING\_MESSAGE, ig1);

}

}

}

catch (Exception evt) {

System. out. println(evt);

}

}

}

}

});

// Forget Password Button

JButton Forget = new JButton("Forget Password? "); // Forget Password object

Forget. setForeground(new Color(0, 125, 255)); // change Forget button foreground color

Forget. setBackground(new Color(245, 245, 250)); // change Forget button background panel color

Forget. setFont(new Font("LucidaGrande", Font. BOLD, 13)); // change forget Button font

Forget. setBorder(BorderFactory. createDashedBorder(new Color(245, 245, 250))); // change Forget button border

Forget. setBounds(770, 685, 120, 17); // set Forget button location

frame. add(Forget); // Attached button with frame

Forget. addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

if (e. getSource() == Forget) {

Forget obj = new Forget();

}

}

});

JButton EmployeeLogin = new JButton("Employee Login"); // Forget Password object

EmployeeLogin. setForeground(new Color(0, 125, 255)); // change Forget button foreground color

EmployeeLogin. setBackground(new Color(245, 245, 250)); // change Forget button background panel color

EmployeeLogin. setFont(new Font("LucidaGrande", Font. BOLD, 13)); // change forget Button font

EmployeeLogin. setBorder(BorderFactory. createDashedBorder(new Color(245, 245, 250))); // change Forget button

// border

EmployeeLogin. setBounds(770, 710, 105, 17); // set Forget button location

frame. add(EmployeeLogin);

EmployeeLogin. addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

EmployeeLogin obj = new EmployeeLogin();

frame. dispose();

}

});

// Change Password Button

JButton ChangePas = new JButton("Change Password"); // Password changed Button object with text

ChangePas. setForeground(new Color(0, 128, 255)); // Password changed Button foreground color

ChangePas. setBackground(new Color(245, 245, 250)); // Password changed Button Background panel color

ChangePas. setFont(new Font("LucidaGrande", Font. BOLD, 13)); // Password changed Button font

ChangePas. setBorder(BorderFactory. createDashedBorder(new Color(245, 245, 250))); // Password changed Button

// border

ChangePas. setBounds(1022, 685, 120, 17); // Password changed Button x, y, width , height

frame. add(ChangePas);

ChangePas. addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

if (e. getSource() == ChangePas) {

ChangePassword obj = new ChangePassword();

}

}

});

// Security Question Button

JButton SecQue = new JButton("Change Security Question"); // Change Security Question Button object with text

SecQue. setForeground(new Color(0, 128, 255)); // Change Security Question Button foreground color

SecQue. setBackground(new Color(245, 245, 250)); // Change Security Question Button background panel color

SecQue. setFont(new Font("LucidaGrande", Font. BOLD, 13)); // Change Security Question Button font

SecQue. setBorder(BorderFactory. createDashedBorder(new Color(245, 245, 250))); // Change Security Question Button

// Border panel color

SecQue. setBounds(990, 710, 170, 17); // Change Security Question Button x, y, width , height

frame. add(SecQue);

SecQue. addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

if (e. getSource() == SecQue) {

ChangeSecurityQuestion obj = new ChangeSecurityQuestion();

}

}

});

ImageIcon fb = new ImageIcon("C: \\Users\\AHsan Usmani\\Desktop\\Programs\\Project GMS\\src\\facebook. png");

JButton facebook = new JButton(fb);

facebook. setBorder(BorderFactory. createDashedBorder(new Color(245, 245, 250)));

facebook. setBounds(890, 756, 28, 28);

frame. add(facebook);

ImageIcon gm = new ImageIcon("C: \\Users\\AHsan Usmani\\Desktop\\Programs\\Project GMS\\src\\Gmail. png");

JButton gmail = new JButton(gm);

gmail. setBorder(BorderFactory. createDashedBorder(new Color(245, 245, 250)));

gmail. setBounds(940, 756, 28, 27);

frame. add(gmail);

ImageIcon inst = new ImageIcon("C: \\Users\\AHsan Usmani\\Desktop\\Programs\\Project GMS\\src\\instagram. png");

JButton instagram = new JButton(inst);

instagram. setBorder(BorderFactory. createDashedBorder(new Color(245, 245, 250)));

instagram. setBounds(990, 756, 28, 28);

frame. add(instagram);

JButton CreatAcc = new JButton("Create an account"); // Create account Button object

CreatAcc. setForeground(new Color(0, 128, 255)); // Create account Button foreground color

CreatAcc. setBackground(new Color(238, 238, 238)); // Create account Button Background

CreatAcc. setFont(new Font("LocidaGrande", Font. BOLD, 19)); // Create account Button Font

CreatAcc. setBorder(BorderFactory. createDashedBorder(new Color(238, 238, 238))); // Create account Button Border

CreatAcc. setBounds(868, 819, 175, 17); // Create account Button x, y, width, height

} }

// ActionListener for Create an account Button

CreatAcc. addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

if (e. getSource() == CreatAcc) {

SignUp obj = new SignUp();

// frame. hide();

}

}

});

frame. add(CreatAcc);

// Implements Panel in the JFrame work

panel = new JPanel(); // Main Login Panel object : panel

int P\_width = 420, P\_height = 510;

panel. setBounds(width / 2 - P\_width / 2, height / 2 - P\_height / 2, P\_width, P\_height); // width/2 -P\_width/2

// height/2 - P\_height/2

panel. setBackground(new Color(245, 245, 250)); // suppose the Width & height of any Screen is 1920 & 1080.

// 1920/2 - 420/2 1080/2 - 550/2

frame. add(panel);

panel. addFocusListener(new FocusListener() {

@Override

public void focusGained(FocusEvent e) {

if (username. getText(). equals("")) {

username. setText(" Username");

}

}

@Override

public void focusLost(FocusEvent e) {

// TODO Auto-generated method stub

}

});

// Background1

ImageIcon background\_image = new ImageIcon(

"C: \\Users\\AHsan Usmani\\Desktop\\Programs\\Project GMS\\src\\background. jpg");

Image img = background\_image. getImage();

Image temp\_img = img. getScaledInstance(width, height, Image. SCALE\_SMOOTH);

background\_image = new ImageIcon(temp\_img);

JLabel background = new JLabel("", background\_image, JLabel. LEFT);

background. setLayout(new FlowLayout());

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

frame. add(background);

// Frame

frame. setSize(width, height); // Frame size

frame. setLayout(null);

frame. setDefaultCloseOperation(frame. EXIT\_ON\_CLOSE);

frame. setResizable(false);

frame. setVisible(true);

frame. setLayout(null);

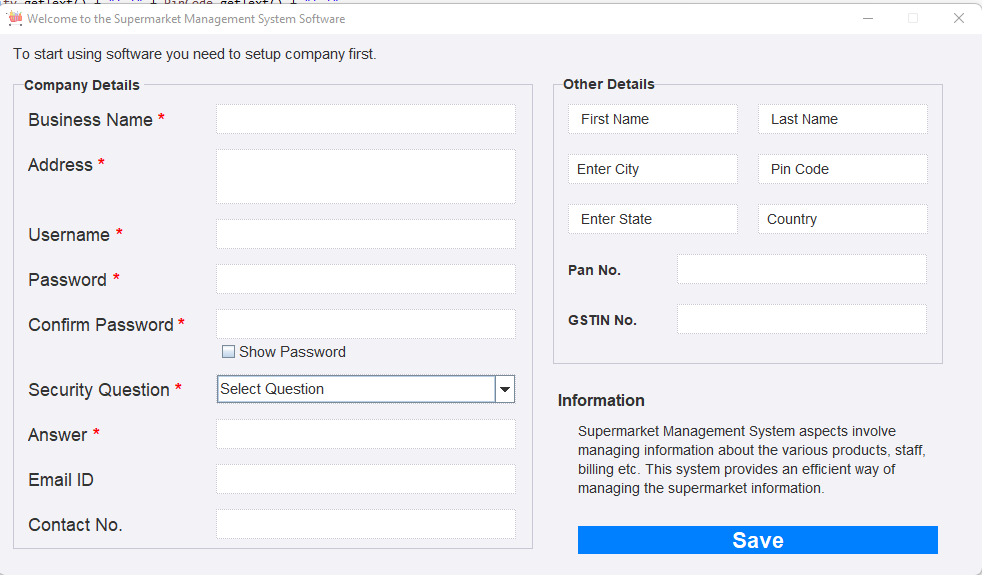
frame. setLocationRelativeTo(null);

}

public static void main(String[] args) {

Login L1 = new Login();

CREARE AN ACCOUNT FOR ADMINISTATOR WINDOW SCREEN



**Output form information**

This form is used to create an administrator account in super market management system software.

**Coding of Create Account Form**

import javax. swing. border. Border;

import javax. swing. \*;

import java. awt. event. \*;

import java. awt. \*;

import java. sql. \*;

public class SignUp {

JFrame frame;

JTextField Name;

JLabel error1, error2, error3, error4, error5, error6, error7;

SignUp() {

frame = new JFrame("Welcome to the Supermarket Management System Software"); // SignUp page Tital

frame. setSize(1000, 580); // frame size

frame. setLayout(null); // Frame layout

frame. getContentPane(). setBackground(new Color(242, 242, 247)); // Frame Background color new Color

// (242,242,247)

Image icon = Toolkit. getDefaultToolkit(). getImage("C: \\Users\\AHsan Usmani\\Desktop\\Programs\\Project GMS\\src\\su1. png");

frame. setIconImage(icon);

Border PanelBorder = BorderFactory. createLineBorder(new Color(199, 199, 214)); // Make a border for panel

// initialize on the top because

// of use many other panel

// Starting test Label

JLabel Text = new JLabel("To start using software you need to setup company first. "); // Create a label object

// for show a text top

// on the frame

Text. setFont(new Font("LucidaGrande", Font. PLAIN, 15)); // set text font

Text. setBounds(15, 10, 450, 18); // set text x, y, width , height

frame. add(Text); // add into the frame

// Company Details Text

JLabel ComText = new JLabel("Company Details"); // Create a label object ComText for a panel title

ComText. setFont(new Font("LucidaGrande", Font. BOLD, 14)); // ComText font

ComText. setBackground(Color. green); // ComText background color

ComText. setBounds(26, 41, 150, 18); // Comtext x, y, width , height

frame. add(ComText); // add ComText into the frame

JPanel ComTextP = new JPanel(); // Create a small panel object because background of title Company Details

ComTextP. setBounds(26, 41, 120, 18); // Same x, y, width , height because for make a design for Company details

ComTextP. setBackground(new Color(242, 242, 247)); // set background color

frame. add(ComTextP); // add into the frma

// Label for Business Name

JLabel BusNam = new JLabel("Business Name");

BusNam. setFont(new Font("LucidaGrande", Font. PLAIN, 18));

BusNam. setBounds(30, 70, 200, 30);

frame. add(BusNam);

// Label for \*

JLabel fix1 = new JLabel("\*");

fix1. setBounds(160, 70, 20, 30);

fix1. setFont(new Font("LucidaGrande", Font. BOLD, 18));

fix1. setForeground(Color. red);

frame. add(fix1);

// TextField for Business Name

JTextField BusName = new JTextField();

BusName. setFont(new Font("LucidaGrande", Font. PLAIN, 18));

BusName. setBorder(BorderFactory. createDashedBorder(new Color(199, 199, 214)));

BusName. setBounds(218, 70, 300, 30);

frame. add(BusName);

BusName. setToolTipText("Only alphabets");

// Label for show Address

JLabel Add = new JLabel("Address");

Add. setFont(new Font("LucidaGrande", Font. PLAIN, 18));

Add. setBounds(30, 115, 200, 30);

frame. add(Add);

// Label for \*

JLabel fix2 = new JLabel("\*");

fix2. setBounds(100, 115, 20, 30);

fix2. setFont(new Font("LucidaGrande", Font. BOLD, 18));

fix2. setForeground(Color. red);

frame. add(fix2);

// TextField for make in front of Address

JTextField Address = new JTextField();

Address. setFont(new Font("LucidaGrande", Font. PLAIN, 15));

Address. setBorder(BorderFactory. createDashedBorder(new Color(199, 199, 214)));

Address. setBounds(218, 115, 300, 55);

frame. add(Address);

// Label for username

JLabel userN = new JLabel("Username");

userN. setFont(new Font("LucidaGrande", Font. PLAIN, 18));

userN. setBounds(30, 185, 200, 30);

frame. add(userN);

// Label for \*

JLabel fix3 = new JLabel("\*");

fix3. setBounds(118, 185, 20, 30);

fix3. setFont(new Font("LucidaGrande", Font. BOLD, 18));

fix3. setForeground(Color. red);

frame. add(fix3);

// TextField for UserName

JTextField username = new JTextField();

username. setFont(new Font("LucidaGrande", Font. PLAIN, 18));

username. setBorder(BorderFactory. createDashedBorder(new Color(199, 199, 214)));

username. setBounds(218, 185, 300, 30);

frame. add(username);

username. setToolTipText("Only alphabets & numbers");

// Label for Password

JLabel passwordL = new JLabel("Password");

passwordL. setFont(new Font("Lucidardrande", Font. PLAIN, 18));

passwordL. setBounds(30, 230, 200, 30);

frame. add(passwordL);

// Label for \*

JLabel fix4 = new JLabel("\*");

fix4. setBounds(115, 230, 20, 30);

fix4. setFont(new Font("LucidaGrande", Font. BOLD, 18));

fix4. setForeground(Color. red);

frame. add(fix4);

// TextFied for JPassword

JPasswordField password = new JPasswordField();

password. setFont(new Font("LucidaGrande", Font. PLAIN, 18));

password. setBorder(BorderFactory. createDashedBorder(new Color(199, 199, 214)));

password. setBounds(218, 230, 300, 30);

password. setEchoChar('\*');

frame. add(password);

password. setToolTipText("Use alphabets, numbers & special symbols. ");

// Label for confirm password

JLabel ConPass = new JLabel("Confirm Password");

ConPass. setFont(new Font("LucidaGrande", Font. PLAIN, 18));

ConPass. setBounds(30, 275, 200, 30);

frame. add(ConPass);

// Label for \*

JLabel fix5 = new JLabel("\*");

fix5. setBounds(180, 275, 20, 30);

fix5. setFont(new Font("LucidaGrande", Font. BOLD, 18));

fix5. setForeground(Color. red);

frame. add(fix5);

// JPassword field for confim password

JPasswordField ConfirmPassword = new JPasswordField();

ConfirmPassword. setFont(new Font("LucidaGrande", Font. PLAIN, 18));

ConfirmPassword. setBorder(BorderFactory. createDashedBorder(new Color(199, 199, 214)));

ConfirmPassword. setBounds(218, 275, 300, 30);

ConfirmPassword. setEchoChar('\*');

frame. add(ConfirmPassword);

ConfirmPassword. setToolTipText("Cofirm password must be same as password");

// Check box for show password of both of field

JCheckBox showPas = new JCheckBox("Show Password");

showPas. setBackground(new Color(242, 242, 247));

showPas. setFont(new Font("LucidaGrande", Font. PLAIN, 15));

showPas. setBounds(220, 310, 140, 15);

frame. add(showPas);

showPas. addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

if (showPas. isSelected()) {

password. setEchoChar((char) 0);

ConfirmPassword. setEchoChar((char) 0);

} else {

password. setEchoChar('\*');

ConfirmPassword. setEchoChar('\*');

}

}

});

// Label for Security Question

JLabel SQ = new JLabel("Security Question");

SQ. setFont(new Font("LucidaGrande", Font. PLAIN, 18));

SQ. setBounds(30, 340, 200, 30);

frame. add(SQ);

// Label for \*

JLabel fix6 = new JLabel("\*");

fix6. setBounds(177, 340, 20, 30);

fix6. setFont(new Font("LucidaGrande", Font. BOLD, 18));

fix6. setForeground(Color. red);

frame. add(fix6);

// Make a ComboBox and initialize Question on in

String SecQ[] = { "Select Question", "In what city were you born? ", "What is the name of your favorite pet? ",

"What is your mother's maiden name? ", "What high school did you attend? ",

"What is the name of your first school? ", "What was the make of your first car? ",

"What was your favorite food as a child? ", "Where did you meet your spouse? " };

JComboBox SSQ = new JComboBox(SecQ);

SSQ. setFont(new Font("LucidaGrande", Font. PLAIN, 15));

SSQ. setBorder(BorderFactory. createDashedBorder(new Color(199, 199, 214)));

SSQ. setBackground(Color. white);

SSQ. setBounds(218, 340, 300, 30);

frame. add(SSQ);

// Label for answer the Question

JLabel Ans = new JLabel("Answer");

Ans. setFont(new Font("LucidaGrande", Font. PLAIN, 18));

Ans. setBounds(30, 385, 200, 30);

frame. add(Ans);

// Label for \*

JLabel fix7 = new JLabel("\*");

fix7. setBounds(95, 385, 20, 30);

fix7. setFont(new Font("LucidaGrande", Font. BOLD, 18));

fix7. setForeground(Color. red);

frame. add(fix7);

// TextField for Answer

JTextField answer = new JTextField();

answer. setFont(new Font("LucidaGrande", Font. PLAIN, 18));

answer. setBorder(BorderFactory. createDashedBorder(new Color(199, 199, 214)));

answer. setBounds(218, 385, 300, 30);

frame. add(answer);

// Email Id Label

JLabel Email = new JLabel("Email ID ");

Email. setFont(new Font("LucidaGrande", Font. PLAIN, 18));

Email. setBounds(30, 430, 200, 30);

frame. add(Email);

// TextField for EmailId

JTextField email = new JTextField();

email. setFont(new Font("LucidaGrande", Font. PLAIN, 18));

email. setBorder(BorderFactory. createDashedBorder(new Color(199, 199, 214)));

email. setBounds(218, 430, 300, 30);

frame. add(email);

// TextField for Phone Number

JLabel numbero = new JLabel("Contact No. ");

numbero. setFont(new Font("LucidaGrande", Font. PLAIN, 18));

numbero. setBounds(30, 475, 200, 30);

frame. add(numbero);

// TextFiedl for Phone Number

JTextField number = new JTextField();

number. setFont(new Font("LucidaGrande", Font. PLAIN, 18));

number. setBorder(BorderFactory. createDashedBorder(new Color(199, 199, 214)));

number. setBounds(218, 475, 300, 30);

frame. add(number);

// Label Other Details

JLabel OD = new JLabel("Other Details");

OD. setFont(new Font("LucidaGrande", Font. BOLD, 14));

OD. setBounds(565, 34, 150, 30);

frame. add(OD);

JPanel ODPP = new JPanel();

ODPP. setBounds(565, 38, 90, 25);

ODPP. setBackground(new Color(242, 242, 247));

ODPP. setBorder(BorderFactory. createDashedBorder(new Color(242, 242, 247)));

frame. add(ODPP);

// Text field for name

JTextField FirstName = new JTextField(" First Name");

FirstName. setFont(new Font("LucidaGrande", Font. PLAIN, 14));

FirstName. setBorder(BorderFactory. createDashedBorder(new Color(199, 199, 214)));

FirstName. setBounds(570, 70, 170, 30);

frame. add(FirstName);

FirstName. addKeyListener(new KeyAdapter() {

@Override

public void keyPressed(KeyEvent e) {

if (FirstName. getText(). equals(" First Name")) {

FirstName. setText("");

}

}

});

FirstName. addFocusListener(new FocusListener() {

@Override

public void focusGained(FocusEvent e) {

}

@Override

public void focusLost(FocusEvent e) {

if (FirstName. getText(). length() == 0) {

FirstName. setText(" First Name");

}

}

});

// Text field for Last Name

JTextField LastName = new JTextField(" Last Name");

LastName. setFont(new Font("LucidaGrande", Font. PLAIN, 14));

LastName. setBorder(BorderFactory. createDashedBorder(new Color(199, 199, 214)));

LastName. setBounds(760, 70, 170, 30);

frame. add(LastName);

LastName. addKeyListener(new KeyAdapter() {

@Override

public void keyPressed(KeyEvent e) {

if (LastName. getText(). equals(" Last Name")) {

LastName. setText("");

}

}

});

LastName. addFocusListener(new FocusListener() {

@Override

public void focusGained(FocusEvent e) {

}

@Override

public void focusLost(FocusEvent e) {

if (LastName. getText(). length() == 0) {

LastName. setText(" Last Name");

}

}

});

// Text field City

JTextField City = new JTextField(" Enter City");

City. setFont(new Font("LucidaGrande", Font. PLAIN, 14));

City. setBorder(BorderFactory. createDashedBorder(new Color(199, 199, 214)));

City. setBounds(570, 120, 170, 30);

frame. add(City);

City. addKeyListener(new KeyAdapter() {

@Override

public void keyPressed(KeyEvent e) {

if (City. getText(). equals(" Enter City")) {

City. setText("");

}

}

});

City. addFocusListener(new FocusListener() {

@Override

public void focusGained(FocusEvent e) {

}

@Override

public void focusLost(FocusEvent e) {

if (City. getText(). length() == 0) {

City. setText(" Enter City");

}

}

});

// Text field for Pin Code

JTextField PinCode = new JTextField(" Pin Code");

PinCode. setFont(new Font("LucidaGrande", Font. PLAIN, 14));

PinCode. setBorder(BorderFactory. createDashedBorder(new Color(199, 199, 214)));

PinCode. setBounds(760, 120, 170, 30);

frame. add(PinCode);

PinCode. addKeyListener(new KeyAdapter() {

@Override

public void keyPressed(KeyEvent e) {

if (PinCode. getText(). equals(" Pin Code")) {

PinCode. setText("");

}

}

});

PinCode. addFocusListener(new FocusListener() {

@Override

public void focusGained(FocusEvent e) {

}

@Override

public void focusLost(FocusEvent e) {

if (PinCode. getText(). length() == 0) {

PinCode. setText(" Pin Code");

}

}

});

// Text field for State

JTextField State = new JTextField(" Enter State");

State. setFont(new Font("LucidaGrande", Font. PLAIN, 14));

State. setBorder(BorderFactory. createDashedBorder(new Color(199, 199, 214)));

State. setBounds(570, 170, 170, 30);

frame. add(State);

State. addKeyListener(new KeyAdapter() {

@Override

public void keyPressed(KeyEvent e) {

if (State. getText(). equals(" Enter State")) {

State. setText("");

}

}

});

State. addFocusListener(new FocusListener() {

@Override

public void focusGained(FocusEvent e) {

}

@Override

public void focusLost(FocusEvent e) {

if (State. getText(). length() == 0) {

State. setText(" Enter State");

}

}

});

// TextField for Country

JTextField Country = new JTextField(" Country");

Country. setFont(new Font("LucidaGrande", Font. PLAIN, 14));

Country. setBorder(BorderFactory. createDashedBorder(new Color(199, 199, 214)));

Country. setBounds(760, 170, 170, 30);

frame. add(Country);

Country. addKeyListener(new KeyAdapter() {

@Override

public void keyPressed(KeyEvent e) {

if (Country. getText(). equals(" Country")) {

Country. setText("");

}

}

});

Country. addFocusListener(new FocusListener() {

@Override

public void focusGained(FocusEvent e) {

}

@Override

public void focusLost(FocusEvent e) {

if (Country. getText(). length() == 0) {

Country. setText(" Country");

}

}

});

// Text Field for Pan card Number

JLabel Pan = new JLabel("Pan No. ");

Pan. setFont(new Font("LucidaGrande", Font. BOLD, 14));

Pan. setBounds(570, 220, 120, 30);

frame. add(Pan);

// TextField for pan card number

JTextField PanNo = new JTextField();

PanNo. setFont(new Font("LucidaGrande", Font. PLAIN, 18));

PanNo. setBorder(BorderFactory. createDashedBorder(new Color(199, 199, 214)));

PanNo. setBounds(679, 220, 250, 30);

frame. add(PanNo);

// Text Field for Pan card Number

JLabel GSTN = new JLabel("GSTIN No. ");

GSTN. setFont(new Font("LucidaGrande", Font. BOLD, 14));

GSTN. setBounds(570, 270, 120, 30);

frame. add(GSTN);

// TextField for GST Number

JTextField GST = new JTextField();

GST. setFont(new Font("LucidaGrande", Font. PLAIN, 15));

GST. setBorder(BorderFactory. createDashedBorder(new Color(199, 199, 214)));

GST. setBounds(679, 270, 250, 30);

frame. add(GST);

// Other Details main Panel

JPanel ODP = new JPanel();

ODP. setBorder(PanelBorder);

ODP. setBackground(new Color(242, 242, 247));

ODP. setBounds(555, 50, 390, 280);

frame. add(ODP);

// Label for show information title

JLabel infoT = new JLabel("Information");

infoT. setFont(new Font("LucidaGrande", Font. BOLD, 16));

infoT. setBounds(560, 350, 90, 30);

frame. add(infoT);

// Make a paragraph atteched with panel

JPanel infoP = new JPanel();

infoP. setBounds(570, 382, 370, 110);

infoP. setBackground(new Color(242, 242, 247));

frame. add(infoP);

// TextArea for information

JTextArea info = new JTextArea("Supermarket Management System aspects involve \n"

+ "managing information about the various products, staff, \n"

+ "billing etc. This system provides an efficient way of\n" + "managing the supermarket information. ");

info. setFont(new Font("LucidaGrande", Font. PLAIN, 14));

info. setBackground(new Color(242, 242, 247));

info. setEditable(false);

infoP. add(info);

// error1= not enter the Business Name

error1 = new JLabel(" \u274C Enter the Business name"); // Error label for not enter the Business name

error1. setForeground(Color. red);

error1. setFont(new Font("LucidaGrande", Font. PLAIN, 11));

error1. setBounds(218, 100, 300, 15);

error1. setVisible(false);

frame. add(error1);

// error2= not enter the Address

error2 = new JLabel(" \u274C Enter the Address"); // Error label for not enter the address

error2. setForeground(Color. red);

error2. setFont(new Font("LucidaGrande", Font. PLAIN, 11));

error2. setBounds(218, 169, 300, 15);

error2. setVisible(false);

frame. add(error2);

// error3= not enter the username

error3 = new JLabel(" \u274C Enter the username"); // Error label for not enter the username

error3. setForeground(Color. red);

error3. setFont(new Font("LucidaGrande", Font. PLAIN, 11));

error3. setBounds(218, 215, 300, 15);

error3. setVisible(false);

frame. add(error3);

// error4= not enter the Password

error4 = new JLabel(" \u274C Enter the Password"); // Error label for not enter the Password

error4. setForeground(Color. red);

error4. setFont(new Font("LucidaGrande", Font. PLAIN, 11));

error4. setBounds(218, 260, 300, 15);

error4. setVisible(false);

frame. add(error4);

// error5= not enter the confirm Password

error5 = new JLabel(" \u274C Enter the Password"); // Error label for not enter the confirm password

error5. setForeground(Color. red);

error5. setFont(new Font("LucidaGrande", Font. PLAIN, 11));

error5. setBounds(360, 310, 300, 15);

error5. setVisible(false);

frame. add(error5);

// error6= not enter the select question

error6 = new JLabel(" \u274C Select the question"); // Error label for not enter the select question

error6. setForeground(Color. red);

error6. setFont(new Font("LucidaGrande", Font. PLAIN, 11));

error6. setBounds(218, 370, 300, 15);

error6. setVisible(false);

frame. add(error6);

// error7= not enter the select question

error7 = new JLabel(" \u274C Select the question"); // Error label for not enter the select question

error7. setForeground(Color. red);

error7. setFont(new Font("LucidaGrande", Font. PLAIN, 11));

error7. setBounds(218, 415, 300, 15);

error7. setVisible(false);

frame. add(error7);

// JButton for save all information

JButton save = new JButton("Save");

save. setBackground(new Color(0, 128, 255));

save. setForeground(Color. white);

save. setFont(new Font("LucidaGrande", Font. BOLD, 22));

save. setBorder(BorderFactory. createDashedBorder(new Color(0, 128, 255)));

save. setBounds(580, 490, 360, 30);

frame. add(save);

save. addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

if (e. getSource() == save) {

if (BusName. getText(). length() == 0) {

BusName. setBorder(BorderFactory. createLineBorder(Color. red, 1));

error1. setVisible(true);

BusName. addMouseListener(new MouseAdapter() {

@Override

public void mouseClicked(MouseEvent e) {

BusName. setBorder(BorderFactory. createLineBorder(new Color(199, 199, 214)));

error1. setVisible(false);

}

});

} else if (! (password. getText(). equals(ConfirmPassword. getText()))) {

ImageIcon ig1 = new ImageIcon(

"C: \\Users\\AHsan Usmani\\Desktop\\Programs\\Project GMS\\src\\cross. png");

JOptionPane. showMessageDialog(null, "Password and Confirm password not Match", "Warrning",

JOptionPane. WARNING\_MESSAGE, ig1);

}

else if (Address. getText(). length() == 0) {

Address. setBorder(BorderFactory. createLineBorder(Color. red, 1));

error2. setVisible(true);

Address. addMouseListener(new MouseAdapter() {

@Override

public void mouseClicked(MouseEvent e) {

Address. setBorder(BorderFactory. createLineBorder(new Color(199, 199, 214)));

error2. setVisible(false);

}

});

} else if (username. getText(). length() == 0) {

username. setBorder(BorderFactory. createLineBorder(Color. red, 1));

error3. setVisible(true);

username. addMouseListener(new MouseAdapter() {

@Override

public void mouseClicked(MouseEvent e) {

username. setBorder(BorderFactory. createLineBorder(new Color(199, 199, 214)));

error3. setVisible(false);

}

});

}

else if (password. getText(). length() == 0) {

password. setBorder(BorderFactory. createLineBorder(Color. red, 1));

error4. setVisible(true);

password. addMouseListener(new MouseAdapter() {

@Override

public void mouseClicked(MouseEvent e) {

password. setBorder(BorderFactory. createLineBorder(new Color(199, 199, 214)));

error4. setVisible(false);

}

});

}

else if (ConfirmPassword. getText(). length() == 0) {

ConfirmPassword. setBorder(BorderFactory. createLineBorder(Color. red, 1));

error5. setVisible(true);

ConfirmPassword. addMouseListener(new MouseAdapter() {

@Override

public void mouseClicked(MouseEvent e) {

ConfirmPassword. setBorder(BorderFactory. createLineBorder(new Color(199, 199, 214)));

error5. setVisible(false);

}

});

}

else if (SSQ. getItemAt(SSQ. getSelectedIndex()) == SSQ. getItemAt(0)) {

SSQ. setBorder(BorderFactory. createLineBorder(Color. red, 1));

System. out. println(SSQ. getItemAt(0));

error6. setVisible(true);

SSQ. addMouseListener(new MouseAdapter() {

@Override

public void mouseClicked(MouseEvent e) {

SSQ. setBorder(BorderFactory. createLineBorder(new Color(199, 199, 214)));

error6. setVisible(false);

}

});

} else if (answer. getText(). length() == 0) {

answer. setBorder(BorderFactory. createLineBorder(Color. red, 1));

error7. setVisible(true);

answer. addMouseListener(new MouseAdapter() {

@Override

public void mouseClicked(MouseEvent e) {

answer. setBorder(BorderFactory. createLineBorder(new Color(199, 199, 214)));

error7. setVisible(false);

}

});

}

else {

try {

Connection con = DriverManager. getConnection(

"jdbc: ucanaccess: //C: \\Users\\AHsan Usmani\\Desktop\\Programs\\Project GMS\\src\\sms. accdb");

Statement st = con. createStatement();

int i = st. executeUpdate("insert into userTable values('" + username. getText() + "', '"

+ password. getText() + "', '" + BusName. getText() + "', '" + Address. getText() + "', '"

+ SSQ. getSelectedItem(). toString() + "', '" + answer. getText() + "', '"

+ email. getText() + "', '" + number. getText() + "', '" + FirstName. getText() + "', '"

+ LastName. getText() + "', '" + City. getText() + "', '" + PinCode. getText() + "', '"

+ State. getText() + "', '" + Country. getText() + "', '" + PanNo. getText() + "', '"

+ GSTN. getText() + "')");

if (i > 0) {

ImageIcon ig1 = new ImageIcon(

"C: \\Users\\AHsan Usmani\\Desktop\\Programs\\Project GMS\\src\\done. png");

JOptionPane. showMessageDialog(null, "Account create successfully. ", "Done",

JOptionPane. INFORMATION\_MESSAGE, ig1);

frame. dispose();

}

}

catch (Exception evt) {

System. out. println(evt);

}

}

}

}

});

// Company Details Panel

JPanel ComPanel = new JPanel();

ComPanel. setBorder(PanelBorder);

ComPanel. setBackground(new Color(242, 242, 247));

ComPanel. setBounds(15, 50, 520, 465);

frame. add(ComPanel);

frame. setVisible(true);

frame. setResizable(false);

frame. setLocationRelativeTo(null);

frame. addWindowListener(new WindowAdapter() {

public void windowClosing(WindowEvent e) {

frame. dispose();

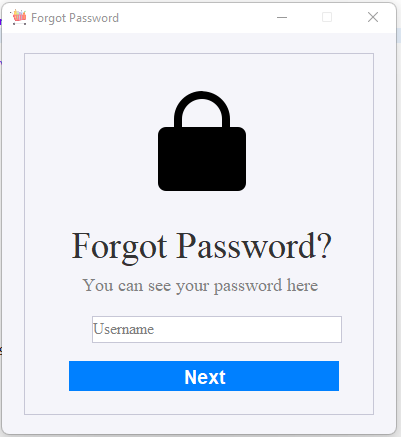
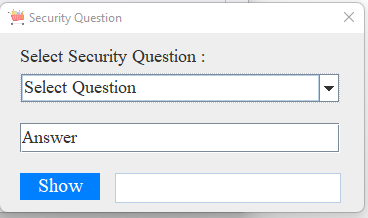
}

}); }

public static void main(String[] args) {

SignUp obj = new SignUp(); }

}

FORGET PASSWORD FOR ADMINISTATOR WINDOW SCREEN

**Output form information**

This form is use to show the password of administrator account in super market management system software.

**Coding of Forget Password Form**

import java. awt. event. \*;

import java. sql. \*;

import javax. swing. \*;

import java. awt. \*;

public class Forget {

JFrame frame;

JPanel panel;

JDialog SQD;

JButton nextB;

JLabel error1, error2, error3;

JTextField username;

Forget() {

frame= new JFrame("Forgot Password");

frame. setSize(410,440);

frame. setLayout(null);

Image icon= Toolkit. getDefaultToolkit(). getImage("C: \\Users\\AHsan Usmani\\Desktop\\Programs\\Project GMS\\src\\su1. png");

frame. setIconImage(icon);

ImageIcon img=new ImageIcon("C: \\Users\\AHsan Usmani\\Desktop\\Programs\\Project GMS\\src\\Lock. png");

JLabel lock=new JLabel(img);

lock. setBounds(100,10,200,200);

frame. add(lock);

JLabel forgotT=new JLabel("Forgot Password? ");

forgotT. setFont(new Font("Times New Roman", Font. PLAIN,37));

forgotT. setBounds(70,193,350,40);

frame. add(forgotT);

JLabel text=new JLabel("You can see your password here");

text. setFont(new Font("Times New Roman", Font. PLAIN,18));

text. setForeground(Color. gray);

text. setBounds(80,240,300,25);

frame. add(text);

//Username textField

username=new JTextField("Username");

username. setBorder(BorderFactory. createLineBorder(new Color(199,199,214)));

username. setFont(new Font("Times New Roman", Font. PLAIN,16));

username. setBounds(90,283,250,27);

username. setForeground(Color. gray);

username. addKeyListener(new KeyAdapter() {

@Override

public void keyTyped(KeyEvent e) {

if(username. getText(). equals("Username"))

username. setText(""); }

});

username. addMouseListener(new MouseAdapter(){

@Override

public void mouseClicked(MouseEvent e){

username. setText("");

}

});

username. addFocusListener(new FocusListener() {

@Override

public void focusGained(FocusEvent e) {

}

@Override

public void focusLost(FocusEvent e) {

if(username. getText(). length()==0) {

username. setText("Username");

}

}

});

frame. add(username);

//Label for username

ImageIcon img1=new ImageIcon("C: \\Users\\AHsan Usmani\\Desktop\\Programs\\Project GMS\\src\\mail1. jpg");

JLabel mail=new JLabel(img1);

mail. setBackground(Color. black);

mail. setForeground(Color. black);

mail. setBounds(90,283,30,25);

frame. add(mail);

//error1= not enter the username

JLabel error1 = new JLabel(" \u274C Enter the Username"); // Error label for not enter the username

error1. setForeground(Color. red);

error1. setFont(new Font("LucidaGrande", Font. PLAIN,12));

error1. setBounds(90,305,250,25);

error1. setVisible(false);

frame. add(error1);

//error2= not enter the current password

error2 = new JLabel(" \u274C Enter the current password"); // Error label for not enter the current password

error2. setForeground(Color. red);

error2. setFont(new Font("LucidaGrande", Font. PLAIN,12));

error2. setBounds(247,200,250,25);

error2. setVisible(false);

frame. add(error2);

//error3 = not enter the New password

error3 = new JLabel(" \u274C Enter the New password"); // Error label for not enter the New password

error3. setForeground(Color. red);

error3. setFont(new Font("LucidaGrande", Font. PLAIN,12));

error3. setBounds(247,242,250,25);

error3. setVisible(false);

frame. add(error3);

//Button for next

nextB=new JButton("Next");

nextB. setFont(new Font("LucidaGrande", Font. BOLD,19));

nextB. setBorder(BorderFactory. createLineBorder(new Color(0,128,255)));

nextB. setBackground(new Color(0,128,255));

nextB. setForeground(Color. white);

nextB. setBounds(67,328,270,30);

frame. add(nextB);

nextB. addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

if(username. getText(). equals("Username")) {

username. setBorder(BorderFactory. createLineBorder(Color. red,1));

error1. setVisible(true);

username. addMouseListener(new MouseAdapter(){

@Override

public void mouseClicked(MouseEvent e){

username. setBorder(BorderFactory. createLineBorder(new Color(199,199,214)));

error1. setVisible(false);

}

});

}

else {

try {

Connection con=DriverManager. getConnection("jdbc: ucanaccess: //C: \\Users\\AHsan Usmani\\Desktop\\Programs\\Project GMS\\src\\sms. accdb");

Statement st=con. createStatement();

ResultSet rs=st. executeQuery("Select \* from userTable");

if(rs. next()) {

if(username. getText(). equals(rs. getString(1))) {

dialog();

}

else {

ImageIcon ig1=new ImageIcon("C: \\Users\\AHsan Usmani\\Desktop\\Programs\\Project GMS\\src\\error. png");

JOptionPane. showMessageDialog(null, "Please enter a valid username", "Warrning", JOptionPane. WARNING\_MESSAGE, ig1);

}

}

}

catch(Exception evt) { System. out. println(evt); }

}

}

});

//Main Panel

panel=new JPanel();

panel. setBackground(new Color(245,245,250)); //new Color(242,242,247)

panel. setBorder(BorderFactory. createLineBorder(new Color(199,199,214)));

panel. setBounds(22,20,350,362);

frame. add(panel);

frame. setVisible(true);

frame. setLocationRelativeTo(null);

frame. getContentPane(). setBackground(new Color(245,245,250)); //new Color(242,242,247)

frame. setResizable(false);

//frame. dispose();

frame. addWindowListener(new WindowAdapter(){

public void windowClosing(WindowEvent e) {

frame. dispose();

}

});

// frame. setDefaultCloseOperation(JFrame. EXIT\_ON\_CLOSE);

}

public void dialog() {

SQD=new JDialog(frame, "Security Question", true);

SQD. setLocationRelativeTo(null);

SQD. setLayout(null);

JLabel SecQue=new JLabel("Select Security Question : ");

SecQue. setFont(new Font("Times New Roman", Font. PLAIN,18));

SecQue. setBounds(20,8,250,30);

SQD. add(SecQue);

//Make a ComboBox and initialize Question on in

String SecQ[]= {"Select Question",

"In what city were you born? ",

"What is the name of your favorite pet? ",

"What is your mother's maiden name? ",

"What high school did you attend? ",

"What is the name of your first school? ",

"What was the make of your first car? ",

"What was your favorite food as a child? ",

"Where did you meet your spouse? " };

JComboBox SSQ=new JComboBox(SecQ);

SSQ. setFont(new Font("Times New Roman", Font. PLAIN,18));

SSQ. setBorder(BorderFactory. createDashedBorder(new Color(199,199,214)));

SSQ. setBackground(Color. white);

SSQ. setBounds(20,40,320,30);

SQD. add(SSQ);

JTextField answer=new JTextField("Answer");

answer. setFont(new Font("Times New Roman", Font. PLAIN,18));

answer. setBounds(20,90,320,30);

SQD. add(answer);

answer. addKeyListener(new KeyAdapter() {

@Override

public void keyTyped(KeyEvent e) {

if(answer. getText(). equals("Answer"))

answer. setText(""); }

});

answer. addFocusListener(new FocusListener() {

@Override

public void focusGained(FocusEvent e) {

}

@Override

public void focusLost(FocusEvent e) {

if(answer. getText(). length()==0) {

answer. setText("Answer");

}

}

});

error2 = new JLabel(" \u274C Select the Security Question"); // Error label for not enter the current password

error2. setForeground(Color. red);

error2. setFont(new Font("LucidaGrande", Font. PLAIN,12));

error2. setBounds(20,65,320,30);

error2. setVisible(false);

SQD. add(error2);

//error3 = not enter the New password

error3 = new JLabel(" \u274C Enter the Answer"); // Error label for not enter the New password

error3. setForeground(Color. red);

error3. setFont(new Font("LucidaGrande", Font. PLAIN,12));

error3. setBounds(20,113,320,30);

error3. setVisible(false);

SQD. add(error3);

JTextField showPass=new JTextField();

showPass. setFont(new Font("Times New Roman", Font. PLAIN,19));

showPass. setBackground(Color. white);

showPass. setEditable(false);

showPass. setBounds(115,140,226,30);

SQD. add(showPass);

JButton show=new JButton("Show");

show. setFont(new Font("Times New Roman", Font. PLAIN,20));

show. setBorder(BorderFactory. createLineBorder(new Color(0,128,255)));

show. setBackground(new Color(0,128,255));

show. setForeground(Color. white);

show. setBounds(20,140,80,27);

SQD. add(show);

show. addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

if(SSQ. getItemAt(SSQ. getSelectedIndex())==SSQ. getItemAt(0)) {

SSQ. setBorder(BorderFactory. createLineBorder(Color. red,1));

error2. setVisible(true);

SSQ. addMouseListener(new MouseAdapter(){

@Override

public void mouseClicked(MouseEvent e){

SSQ. setBorder(BorderFactory. createLineBorder(new Color(199,199,214)));

error2. setVisible(false);

}

});

}

else if(answer. getText(). equals("Answer")) {

answer. setBorder(BorderFactory. createLineBorder(Color. red,1));

error3. setVisible(true);

answer. addMouseListener(new MouseAdapter(){

@Override

public void mouseClicked(MouseEvent e){

answer. setBorder(BorderFactory. createLineBorder(new Color(199,199,214)));

error3. setVisible(false);

}

});

}

else {

try {

Connection con=DriverManager. getConnection("jdbc: ucanaccess: //C: \\Users\\AHsan Usmani\\Desktop\\Programs\\Project GMS\\src\\sms. accdb");

Statement st=con. createStatement();

ResultSet rs=st. executeQuery("Select \* from userTable");

if(rs. next()) {

if(SSQ. getSelectedItem(). toString(). equals(rs. getString(5))) {

if(answer. getText(). equals(rs. getString(6))) {

showPass. setText(rs. getString(2));

}

else {

ImageIcon ig1=new ImageIcon("C: \\Users\\AHsan Usmani\\Desktop\\Programs\\Project GMS\\src\\error. png");

JOptionPane. showMessageDialog(null, "You enter wrong answer", "Warrning", JOptionPane. WARNING\_MESSAGE, ig1);

}

}

else{

ImageIcon ig1=new ImageIcon("C: \\Users\\AHsan Usmani\\Desktop\\Programs\\Project GMS\\src\\error. png");

JOptionPane. showMessageDialog(null, "Please select the right question", "Warrning", JOptionPane. WARNING\_MESSAGE, ig1);

}

}

}

catch(Exception evt) {

System. out. println(evt);

}

}

}

});

SQD. setSize(380,218);

SQD. setVisible(true);

}

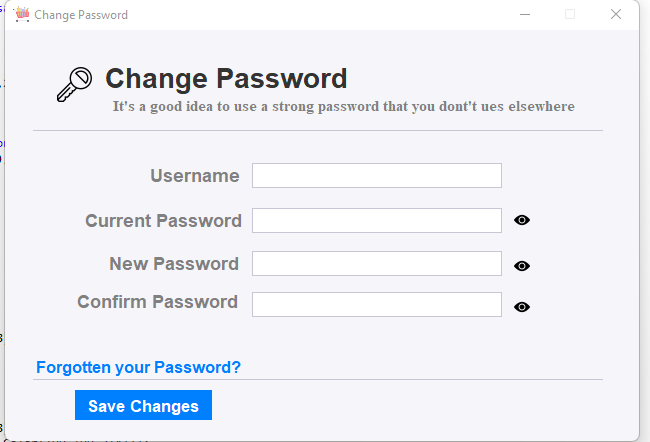
public static void main(String[] args) {

Forget obj=new Forget();

}

}

CHANGE PASSWORD FOR ADMINISTATOR WINDOW SCREEN



**Output form information**

This form is use to change the password of administrator account in super market management system software.

**Coding of Change Password Form**

import javax. swing. \*;

import java. sql. \*;

import java. awt. event. \*;

import java. awt. \*;

public class ChangePassword {

JFrame frame;

JTextField username;

JPasswordField current, ReType, NewPassword;

JLabel error1, error2, error3, error4;

ChangePassword() {

frame=new JFrame("Change Password");

frame. setLayout(null);

Image icon= Toolkit. getDefaultToolkit(). getImage("C: \\Users\\AHsan Usmani\\Desktop\\Programs\\Project GMS\\src\\su1. png");

frame. setIconImage(icon);

JLabel key=new JLabel(new ImageIcon("C: \\Users\\AHsan Usmani\\Desktop\\Programs\\Project GMS\\src\\key. png"));

key. setBounds(52,37,35,35);

frame. add(key);

JLabel CPLabel=new JLabel("Change Password");

CPLabel. setFont(new Font("LucidaGrande", Font. BOLD,28));

CPLabel. setBounds(100,30,250,34);

frame. add(CPLabel);

JLabel text=new JLabel("It's a good idea to use a strong password that you dont't ues elsewhere");

text. setFont(new Font("Times New Roman", Font. BOLD,15));

text. setForeground(Color. gray);

text. setBounds(108,64,500,25);

frame. add(text);

JSeparator sep1=new JSeparator();

sep1. setBounds(28,100,570,1);

sep1. setForeground(new Color(199,199,214));

frame. add(sep1);

JLabel usernameL=new JLabel("Username");

usernameL. setFont(new Font("LucidaGrande", Font. BOLD,18));

usernameL. setForeground(Color. gray);

usernameL. setBounds(145,130,100,30);

frame. add(usernameL);

username=new JTextField();

username. setFont(new Font("LucidaGrande", Font. PLAIN,18));

username. setBorder(BorderFactory. createLineBorder(new Color(199,199,214)));

username. setBounds(247,133,250,25);

frame. add(username);

JLabel CurrentL=new JLabel("Current Password");

CurrentL. setFont(new Font("LucidaGrande", Font. BOLD,18));

CurrentL. setForeground(Color. gray);

CurrentL. setBounds(80,175,250,30);

frame. add(CurrentL);

current=new JPasswordField();

current. setFont(new Font("LucidaGrande", Font. PLAIN,18));

current. setEchoChar('\*');

current. setBorder(BorderFactory. createLineBorder(new Color(199,199,214)));

current. setBounds(247,178,250,25);

frame. add(current);

JLabel CshowIcon=new JLabel(new ImageIcon("C: \\Users\\AHsan Usmani\\Desktop\\Programs\\Project GMS\\src\\view. png"));

CshowIcon. setBounds(502,175,30,30);

JLabel ChideIcon= new JLabel(new ImageIcon("C: \\Users\\AHsan Usmani\\Desktop\\Programs\\Project GMS\\src\\hide. png"));

ChideIcon. setVisible(false);

ChideIcon. setBounds(502,175,30,30);

frame. add(CshowIcon);

frame. add(ChideIcon);

CshowIcon. addMouseListener(new MouseAdapter(){

@Override

public void mousePressed(MouseEvent e){

CshowIcon. setVisible(false);

ChideIcon. setVisible(true);

current. setEchoChar((char)0);

}

@Override

public void mouseReleased(MouseEvent e){

ChideIcon. setVisible(false);

CshowIcon. setVisible(true);

current. setEchoChar('\*');

}

});

JLabel NewPass=new JLabel("New Password");

NewPass. setFont(new Font("LucidaGrande", Font. BOLD,18));

NewPass. setForeground(Color. gray);

NewPass. setBounds(104,218,250,30);

frame. add(NewPass);

NewPassword=new JPasswordField();

NewPassword. setFont(new Font("LucidaGrande", Font. PLAIN,18));

NewPassword. setEchoChar('\*');

NewPassword. setBorder(BorderFactory. createLineBorder(new Color(199,199,214)));

NewPassword. setBounds(247,221,250,25);

frame. add(NewPassword);

JLabel NshowIcon=new JLabel(new ImageIcon("C: \\Users\\AHsan Usmani\\Desktop\\Programs\\Project GMS\\src\\view. png"));

NshowIcon. setBounds(502,221,30,30);

JLabel NhideIcon= new JLabel(new ImageIcon("C: \\Users\\AHsan Usmani\\Desktop\\Programs\\Project GMS\\src\\hide. png"));

NhideIcon. setVisible(false);

NhideIcon. setBounds(502,221,30,30);

frame. add(NshowIcon);

frame. add(NhideIcon);

NshowIcon. addMouseListener(new MouseAdapter(){

@Override

public void mousePressed(MouseEvent e){

NshowIcon. setVisible(false);

NhideIcon. setVisible(true);

NewPassword. setEchoChar((char)0);

}

@Override

public void mouseReleased(MouseEvent e){

NhideIcon. setVisible(false);

NshowIcon. setVisible(true);

NewPassword. setEchoChar('\*');

}

});

JLabel REPass=new JLabel("Confirm Password");

REPass. setFont(new Font("LucidaGrande", Font. BOLD,18));

REPass. setForeground(Color. gray);

REPass. setBounds(72,256,250,30);

frame. add(REPass);

ReType=new JPasswordField();

ReType. setFont(new Font("LucidaGrande", Font. PLAIN,18));

ReType. setEchoChar('\*');

ReType. setBorder(BorderFactory. createLineBorder(new Color(199,199,214)));

ReType. setBounds(247,262,250,25);

frame. add(ReType);

JLabel RshowIcon=new JLabel(new ImageIcon("C: \\Users\\AHsan Usmani\\Desktop\\Programs\\Project GMS\\src\\view. png"));

RshowIcon. setBounds(502,262,30,30);

JLabel RhideIcon= new JLabel(new ImageIcon("C: \\Users\\AHsan Usmani\\Desktop\\Programs\\Project GMS\\src\\hide. png"));

RhideIcon. setVisible(false);

RhideIcon. setBounds(502,262,30,30);

frame. add(RshowIcon);

frame. add(RhideIcon);

RshowIcon. addMouseListener(new MouseAdapter(){

@Override

public void mousePressed(MouseEvent e){

RshowIcon. setVisible(false);

RhideIcon. setVisible(true);

ReType. setEchoChar((char)0);

}

@Override

public void mouseReleased(MouseEvent e){

RhideIcon. setVisible(false);

RshowIcon. setVisible(true);

ReType. setEchoChar('\*');

}

});

JButton Forget=new JButton("Forgotten your Password? ");

Forget. setForeground(new Color(0,128,255)); // change Forget button foreground color

Forget. setBackground(new Color(245,245,250)); // change Forget button background panel color

Forget. setFont(new Font("LucidaGrande", Font. BOLD,16)); // change forget Button font

Forget. setBorder(BorderFactory. createDashedBorder(new Color(245,245,250))); //change Forget button border

Forget. setBounds(28,326,210,21); // set Forget button location

frame. add(Forget);

Forget. addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

if(e. getSource()==Forget) {

Forget obj=new Forget();

} } });

//error1= not enter the username

error1 = new JLabel(" \u274C Enter the username"); // Error label for not enter the username

error1. setForeground(Color. red);

error1. setFont(new Font("LucidaGrande", Font. PLAIN,12));

error1. setBounds(247,155,250,25);

error1. setVisible(false);

frame. add(error1);

//error2= not enter the current password

error2 = new JLabel(" \u274C Enter the current password"); // Error label for not enter the current password

error2. setForeground(Color. red);

error2. setFont(new Font("LucidaGrande", Font. PLAIN,12));

error2. setBounds(247,200,250,25);

error2. setVisible(false);

frame. add(error2);

//error3 = not enter the New password

error3 = new JLabel(" \u274C Enter the New password"); // Error label for not enter the New password

error3. setForeground(Color. red);

error3. setFont(new Font("LucidaGrande", Font. PLAIN,12));

error3. setBounds(247,242,250,25);

error3. setVisible(false);

frame. add(error3);

//error4 = not enter the confirm password

error4 = new JLabel(" \u274C Enter the Confirm password"); // Error label for not enter the confirm password

error4. setForeground(Color. red);

error4. setFont(new Font("LucidaGrande", Font. PLAIN,12));

error4. setBounds(247,286,250,25);

error4. setVisible(false);

frame. add(error4);

JSeparator sep2=new JSeparator();

sep2. setBounds(28,349,570,1);

sep2. setForeground(new Color(199,199,214));

frame. add(sep2);

JButton Save=new JButton("Save Changes");

Save. setBackground(new Color(0,128,255));

Save. setFont(new Font("LucidaGrande", Font. BOLD,17));

Save. setBorder(BorderFactory. createDashedBorder(new Color(0,128,255)));

Save. setForeground(Color. white);

Save. setBounds(70,360,137,30);

Save. addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

if((username. getText(). length()==0)&&(current. getText(). length()==0)&&(NewPassword. getText(). length()==0)&&(ReType. getText(). length()==0)) {

ImageIcon ig1=new ImageIcon("C: \\Users\\AHsan Usmani\\Desktop\\Programs\\Project GMS\\src\\error. png");

JOptionPane. showMessageDialog(null, "Enter all the details. ", "Warrning", JOptionPane. WARNING\_MESSAGE, ig1); }

else if(username. getText(). length()==0) {

username. setBorder(BorderFactory. createLineBorder(Color. red,1));

error1. setVisible(true);

username. addMouseListener(new MouseAdapter(){

@Override

public void mouseClicked(MouseEvent e){

username. setBorder(BorderFactory. createLineBorder(new Color(199,199,214)));

error1. setVisible(false);

} }); }

else if(current. getText(). length()==0) {

current. setBorder(BorderFactory. createLineBorder(Color. red,1));

error2. setVisible(true);

current. addMouseListener(new MouseAdapter(){

@Override

public void mouseClicked(MouseEvent e){

current. setBorder(BorderFactory. createLineBorder(new Color(199,199,214)));

error2. setVisible(false);

} }); }

else if(NewPassword. getText(). length()==0) {

NewPassword. setBorder(BorderFactory. createLineBorder(Color. red,1));

error3. setVisible(true);

NewPassword. addMouseListener(new MouseAdapter(){

@Override

public void mouseClicked(MouseEvent e){

NewPassword. setBorder(BorderFactory. createLineBorder(new Color(199,199,214)));

error3. setVisible(false);

} }); }

else if(ReType. getText(). length()==0) {

ReType. setBorder(BorderFactory. createLineBorder(Color. red,1));

error4. setVisible(true);

ReType. addMouseListener(new MouseAdapter(){

@Override

public void mouseClicked(MouseEvent e){

ReType. setBorder(BorderFactory. createLineBorder(new Color(199,199,214)));

error4. setVisible(false);

} }); }

else if(! (NewPassword. getText(). equals(ReType. getText()))) {

ImageIcon ig1=new ImageIcon("C: \\Users\\AHsan Usmani\\Desktop\\Programs\\Project GMS\\src\\cross. png");

JOptionPane. showMessageDialog(null, "New password doesn't match. \nPlease re-enter a new password again", "Warrning", JOptionPane. WARNING\_MESSAGE, ig1); }

else if((NewPassword. getText(). equals(current. getText()))) {

ImageIcon ig1=new ImageIcon("C: \\Users\\AHsan Usmani\\Desktop\\Programs\\Project GMS\\src\\cross. png");

JOptionPane. showMessageDialog(null, "New password and confirm password is same", "Warrning", JOptionPane. WARNING\_MESSAGE, ig1);

}

else {

try {

//Connection con=DriverManager. getConnection("jdbc. ucanaccess: //C: \\Users\\AHsan Usmani\\Desktop\\Programs\\Project GMS\\src\\sms. accdb");

Connection con=DriverManager. getConnection("jdbc: ucanaccess: //C: \\Users\\AHsan Usmani\\Desktop\\Programs\\Project GMS\\src\\sms. accdb");

Statement st=con. createStatement();

ResultSet rs=st. executeQuery("Select \* from userTable");

if(rs. next()) {

if(username. getText(). equals(rs. getString(1))) {

if(current. getText(). equals(rs. getString(2))) {

int i=st. executeUpdate("update userTable set password='"+NewPassword. getText()+"'where username='"+username. getText()+"'");

if(i>0) {

ImageIcon ig1=new ImageIcon("C: \\Users\\AHsan Usmani\\Desktop\\Programs\\Project GMS\\src\\done. png");

JOptionPane. showMessageDialog(null, "Password Changed. ", "Warrning", JOptionPane. WARNING\_MESSAGE, ig1);

frame. dispose(); }

else {

ImageIcon ig1=new ImageIcon("C: \\Users\\AHsan Usmani\\Desktop\\Programs\\Project GMS\\src\\cross. png");

JOptionPane. showMessageDialog(null, "Password is not Changed\nPlease try-ag", "Warrning", JOptionPane. WARNING\_MESSAGE, ig1); } }

else {

ImageIcon ig1=new ImageIcon("C: \\Users\\AHsan Usmani\\Desktop\\Programs\\Project GMS\\src\\cross. png");

JOptionPane. showMessageDialog(null, "You enter wrong password", "Warrning", JOptionPane. WARNING\_MESSAGE, ig1); } }

else {

ImageIcon ig1=new ImageIcon("C: \\Users\\AHsan Usmani\\Desktop\\Programs\\Project GMS\\src\\cross. png");

JOptionPane. showMessageDialog(null, "Please Enter a valid username", "Warrning", JOptionPane. WARNING\_MESSAGE, ig1); } } }

catch(Exception evt) { System. out. println(evt); }

} }); }

frame. add(Save);

frame. getContentPane(). setBackground(new Color(245,245,250)); //new Color(242,242,247)

frame. setVisible(true);

frame. setSize(650,450);

frame. setLocationRelativeTo(null);

frame. setResizable(false);

frame. addWindowListener(new WindowAdapter(){

public void windowClosing(WindowEvent e) {

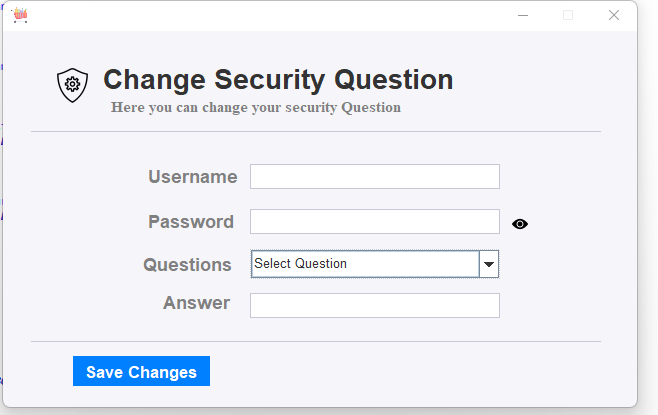
frame. dispose();

} }); }

public static void main(String[] args) {

ChangePassword obj=new ChangePassword();

} }

CHANGE SECURITY QUESTION FOR ADMINISTATOR WINDOW SCREEN

**Output form information**

This form is use to change the Security Question of administrator account in super market management system software.

**Coding of Security Question Form**

import java. awt. event. \*;

import java. awt. \*;

import javax. swing. \*;

import java. sql. \*;

public class ChangeSecurityQuestion {

JButton Save;

JFrame frame;

JLabel error1, error2, error3, error4;

ChangeSecurityQuestion() {

frame = new JFrame();

frame. setLayout(null);

Image icon = Toolkit. getDefaultToolkit(). getImage(

"C: \\Users\\AHsan Usmani\\Desktop\\Programs\\Project GMS\\src\\su1. png");

frame. setIconImage(icon);

JLabel key = new JLabel(new ImageIcon(

"C: \\Users\\AHsan Usmani\\Desktop\\Programs\\Project GMS\\src\\img. png"));

key. setBounds(52, 37, 35, 35);

frame. add(key);

JLabel CPLabel = new JLabel("Change Security Question");

CPLabel. setFont(new Font("LucidaGrande", Font. BOLD, 28));

CPLabel. setBounds(100, 30, 400, 34);

frame. add(CPLabel);

JLabel text = new JLabel("Here you can change your security Question");

text. setFont(new Font("Times New Roman", Font. BOLD, 15));

text. setForeground(Color. gray);

text. setBounds(108, 64, 500, 25);

frame. add(text);

JSeparator sep1 = new JSeparator();

sep1. setBounds(28, 100, 570, 1);

sep1. setForeground(new Color(199, 199, 214));

frame. add(sep1);

JLabel usernameL = new JLabel("Username");

usernameL. setFont(new Font("LucidaGrande", Font. BOLD, 18));

usernameL. setForeground(Color. gray);

usernameL. setBounds(145, 130, 100, 30);

frame. add(usernameL);

JTextField username = new JTextField();

username. setFont(new Font("LucidaGrande", Font. PLAIN, 18));

username. setBorder(

BorderFactory. createLineBorder(new Color(199, 199, 214)));

username. setBounds(247, 133, 250, 25);

frame. add(username);

JLabel CurrentL = new JLabel("Password");

CurrentL. setFont(new Font("LucidaGrande", Font. BOLD, 18));

CurrentL. setForeground(Color. gray);

CurrentL. setBounds(145, 175, 250, 30);

frame. add(CurrentL);

JPasswordField current = new JPasswordField();

current. setFont(new Font("LucidaGrande", Font. PLAIN, 18));

current. setBorder(

BorderFactory. createLineBorder(new Color(199, 199, 214)));

current. setBounds(247, 178, 250, 25);

JLabel NshowIcon = new JLabel(new ImageIcon(

"C: \\Users\\AHsan Usmani\\Desktop\\Programs\\Project GMS\\src\\view. png"));

NshowIcon. setBounds(502, 178, 30, 30);

JLabel NhideIcon = new JLabel(new ImageIcon(

"C: \\Users\\AHsan Usmani\\Desktop\\Programs\\Project GMS\\src\\hide. png"));

NhideIcon. setVisible(false);

NhideIcon. setBounds(502, 178, 30, 30);

frame. add(NshowIcon);

frame. add(NhideIcon);

NshowIcon. addMouseListener(new MouseAdapter() {

@Override

public void mousePressed(MouseEvent e) {

NshowIcon. setVisible(false);

NhideIcon. setVisible(true);

current. setEchoChar((char) 0); }

@Override

public void mouseReleased(MouseEvent e) {

NhideIcon. setVisible(false);

NshowIcon. setVisible(true);

current. setEchoChar('\*'); }

});

frame. add(current);

JLabel password = new JLabel("Questions");

password. setFont(new Font("LucidaGrande", Font. BOLD, 18));

password. setForeground(Color. gray);

password. setBounds(140, 218, 250, 30);

frame. add(password);

String SecQ[] = {"Select Question", "In what city were you born? ",

"What is the name of your favorite pet? ",

"What is your mother's maiden name? ",

"What high school did you attend? ",

"What is the name of your first school? ",

"What was the make of your first car? ",

"What was your favorite food as a child? ",

"Where did you meet your spouse? "};

JComboBox SSQ = new JComboBox(SecQ);

SSQ. setFont(new Font("LucidaGrande", Font. PLAIN, 13));

SSQ. setBorder(

BorderFactory. createDashedBorder(new Color(199, 199, 214)));

SSQ. setBackground(Color. white);

SSQ. setBounds(247, 218, 250, 30);

frame. add(SSQ);

JLabel REPass = new JLabel("Answer");

REPass. setFont(new Font("LucidaGrande", Font. BOLD, 18));

REPass. setForeground(Color. gray);

REPass. setBounds(160, 256, 250, 30);

frame. add(REPass);

JTextField Answer = new JTextField();

Answer. setFont(new Font("LucidaGrande", Font. PLAIN, 18));

Answer. setBorder(

BorderFactory. createLineBorder(new Color(199, 199, 214)));

Answer. setBounds(247, 262, 250, 25);

frame. add(Answer);

JSeparator sep2 = new JSeparator();

sep2. setBounds(28, 310, 570, 1);

sep2. setForeground(new Color(199, 199, 214));

frame. add(sep2);

// error1= not enter the username

error1 = new JLabel(" \u274C Enter the username"); // Error label for

error1. setForeground(Color. red);

error1. setFont(new Font("LucidaGrande", Font. PLAIN, 12));

error1. setBounds(247, 154, 250, 25);

error1. setVisible(false);

frame. add(error1);

// error2= not enter the current passwor

error2 = new JLabel(" \u274C Enter the current password"); // Error

error2. setForeground(Color. red);

error2. setFont(new Font("LucidaGrande", Font. PLAIN, 12));

error2. setBounds(247, 198, 250, 25);

error2. setVisible(false);

frame. add(error2);

// error3 = not enter the New password

error3 = new JLabel(" \u274C Select the question"); // Error label for

error3. setForeground(Color. red);

error3. setFont(new Font("LucidaGrande", Font. PLAIN, 12));

error3. setBounds(247, 239, 250, 30);

error3. setVisible(false);

frame. add(error3);

// error4 = not enter the confirm password

error4 = new JLabel(" \u274C Enter the Confirm password"); // Error

error4. setForeground(Color. red);

error4. setFont(new Font("LucidaGrande", Font. PLAIN, 12));

error4. setBounds(247, 284, 250, 25);

error4. setVisible(false);

frame. add(error4);

Save = new JButton("Save Changes");

Save. setBackground(new Color(0, 128, 255));

Save. setFont(new Font("LucidaGrande", Font. BOLD, 17));

Save. setBorder(

BorderFactory. createDashedBorder(new Color(0, 128, 255)));

Save. setForeground(Color. white);

Save. setBounds(70, 325, 137, 30);

Save. addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

if ((username. getText(). length() == 0)&& (current. getText(). length() == 0)

&& (SSQ. getItemAt(SSQ. getSelectedIndex()) == SSQ. getItemAt(0))

&& (Answer. getText(). length() == 0)) {

ImageIcon ig1 = new ImageIcon(

"C: \\Users\\AHsan Usmani\\Desktop\\Programs\\Project GMS\\src\\error. png");

JOptionPane. showMessageDialog(null,"Enter all the details. ", "Warrning",

JOptionPane. WARNING\_MESSAGE, ig1); }

else if (username. getText(). length() == 0) {

username. setBorder(

BorderFactory. createLineBorder(Color. red, 1));

error1. setVisible(true);

username. addMouseListener(new MouseAdapter() {

@Override

public void mouseClicked(MouseEvent e) {

username. setBorder(BorderFactory. createLineBorder(

new Color(199, 199, 214)));

error1. setVisible(false); } }); }

else if (current. getText(). length() == 0) {

current. setBorder(BorderFactory. createLineBorder(Color. red, 1));

error2. setVisible(true);

current. addMouseListener(new MouseAdapter() {

@Override

public void mouseClicked(MouseEvent e) {

current. setBorder(BorderFactory. createLineBorder(

new Color(199, 199, 214)));

error2. setVisible(false); } }); }

else if (SSQ. getItemAt(SSQ. getSelectedIndex()) == SSQ. getItemAt(0)) {

SSQ. setBorder(BorderFactory. createLineBorder(Color. red, 1));

error3. setVisible(true);

SSQ. addMouseListener(new MouseAdapter() {

@Override

public void mouseClicked(MouseEvent e) {

SSQ. setBorder(BorderFactory. createLineBorder(

new Color(199, 199, 214)));

error3. setVisible(false); } }); }

else if (Answer. getText(). length() == 0) {

Answer. setBorder(

BorderFactory. createLineBorder(Color. red, 1));

error4. setVisible(true);

Answer. addMouseListener(new MouseAdapter() {

@Override

public void mouseClicked(MouseEvent e) {

Answer. setBorder(BorderFactory. createLineBorder(

new Color(199, 199, 214)));

error4. setVisible(false); } }); }

else {

try {

Connection con = DriverManager. getConnection(

"jdbc: ucanaccess: //C: \\Users\\AHsan Usmani\\Desktop\\Programs\\Project GMS\\src\\sms. accdb");

Statement st = con. createStatement();

ResultSet rs = st. executeQuery("select \* from userTable");

if (rs. next()) {

if (username. getText(). equals(rs. getString(1))) {

if (current. getText(). equals(rs. getString(2))) {

st. executeUpdate(

"update userTable set Security\_Question='"+ SSQ. getSelectedItem()+ "', answer='"

+ Answer. getText()+ "' where username='"+ username. getText() + "'");

ImageIcon ig1 = new ImageIcon(

"C: \\Users\\AHsan Usmani\\Desktop\\Programs\\Project GMS\\src\\done. png");

JOptionPane. showMessageDialog(null,"Security Question Changed. ","Warrning",

JOptionPane. WARNING\_MESSAGE, ig1);

frame. dispose(); }

else {

ImageIcon ig1 = new ImageIcon(

"C: \\Users\\AHsan Usmani\\Desktop\\Programs\\Project GMS\\src\\cross. png");

JOptionPane. showMessageDialog(null,"You Enter wrong password","Warrning",

JOptionPane. WARNING\_MESSAGE, ig1); } }

else {

ImageIcon ig1 = new ImageIcon(

"C: \\Users\\AHsan Usmani\\Desktop\\Programs\\Project GMS\\src\\cross. png");

JOptionPane. showMessageDialog(null,"Please Enter a valid username",

"Warrning", JOptionPane. WARNING\_MESSAGE,ig1);

} } } catch (Exception evt) {

System. out. println(evt); } } } });

frame. add(Save);

frame. getContentPane(). setBackground(new Color(245, 245, 250));

frame. setVisible(true);

frame. setSize(650, 415);

frame. setLocationRelativeTo(null);

frame. setResizable(false);

frame. addWindowListener(new WindowAdapter() {

public void windowClosing(WindowEvent e) {

frame. dispose(); } }); }

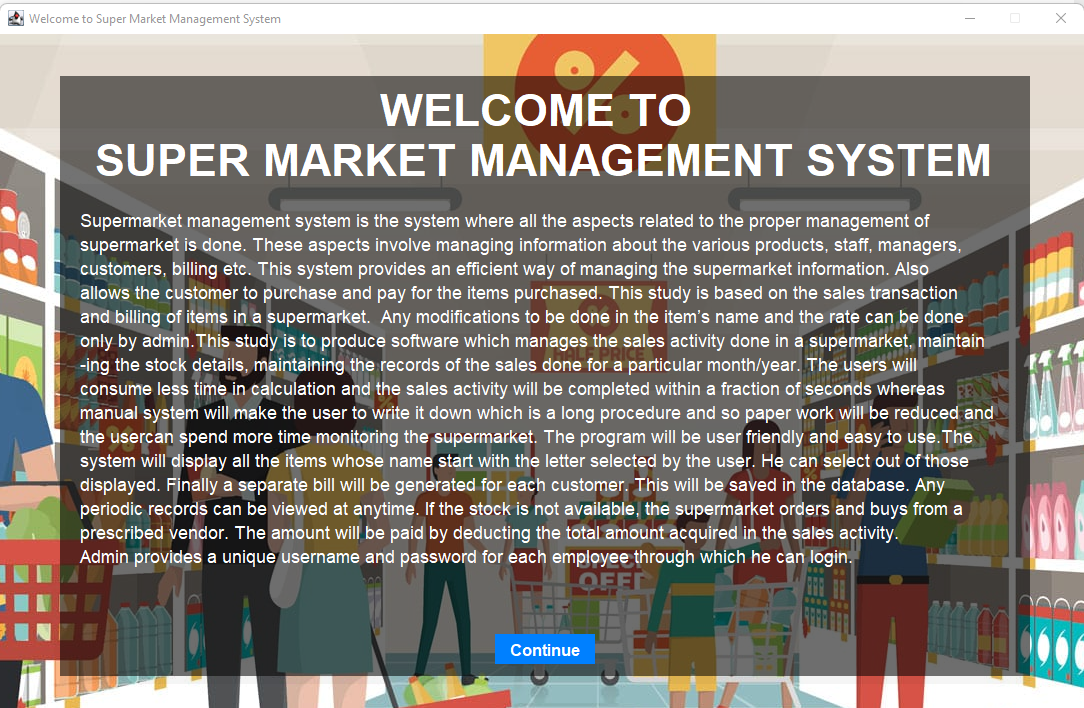
public static void main(String[] args) {

ChangeSecurityQuestion obj = new ChangeSecurityQuestion();

}

}

WELCOME PAGE WINDOW SCREEN



**Output form information**

This page is used for entering the main page of super market management system software.

**Coding of Welcome Page Form**

import java. awt. \*;

import javax. swing. \*;

import java. awt. event. \*;

public class WelcomePage {

JPanel mainPanel;

static JProgressBar PB;

JFrame frame = new JFrame("Welcome to Super Market Management System ");

WelcomePage() {

frame. setSize(1100, 720);

JButton cont = new JButton("Continue");

cont. setBounds(495, 600, 100, 30);

cont. setFont(new Font("LucidaGrande", Font. BOLD, 16));

cont. setBorder(

BorderFactory. createDashedBorder(new Color(0, 128, 255)));

cont. setBackground(new Color(0, 128, 255));

cont. setForeground(Color. white);

cont. setEnabled(true);

frame. add(cont);

cont. addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

MainPage obj = new MainPage();

frame. hide();

} });

JLabel Heading1 = new JLabel("WELCOME TO");

Heading1. setFont(new Font("LucidaGrande", Font. BOLD, 45));

Heading1. setBounds(380, 50, 350, 50);

Heading1. setForeground(Color. white);

frame. add(Heading1);

JLabel Heading2 = new JLabel("SUPER MARKET MANAGEMENT SYSTEM");

Heading2. setFont(new Font("LucidaGrande", Font. BOLD, 45));

Heading2. setBounds(95, 100, 900, 50);

Heading2. setForeground(Color. white);

frame. add(Heading2);

JTextArea text = new JTextArea(

"\n Supermarket management system is the system where all the aspects related to the proper management of "+

"\n supermarket is done. These aspects involve managing information about the various products, staff, managers, "

+ "\n customers, billing etc. This system provides an efficient way of managing the supermarket information. Also "+ "\n allows the customer to purchase and pay for the items purchased. This study is based on the sales transaction "

+ "\n and billing of items in a supermarket. Any modifications to be done in the items name and the rate can be done "

+ "\n only by admin. This study is to produce software which manages the sales activity done in a supermarket, maintain "

+ "\n -ing the stock details, maintaining the records of the sales done for a particular month/year. The users will "

+ "\n consume less time in calculation and the sales activity will be completed within a fraction of seconds whereas "

+ "\n manual system will make the user to write it down which is a long procedure and so paper work will be reduced and "

+ "\n the usercan spend more time monitoring the supermarket. The program will be user friendly and easy to use. The "

+ "\n system will display all the items whose name start with the letter selected by the user. He can select out of those"

+ "\n displayed. Finally a separate bill will be generated for each customer. This will be saved in the database. Any "

+ "\n periodic records can be viewed at anytime. If the stock is not available, the supermarket orders and buys from a "

+ "\n prescribed vendor. The amount will be paid by deducting the total amount acquired in the sales activity. "

+ "\n Admin provides a unique username and password for each employee through which he can login. ");

text. setBounds(60, 150, 970, 500);

text. setFont(new Font("Calibri (Body)", Font. PLAIN, 18));

text. setEditable(false);

text. setForeground(Color. white);

text. setBackground(new Color(0, 0, 0, 10));

frame. add(text);

mainPanel = new JPanel();

mainPanel. setBounds(60, 42, 970, 600);

mainPanel. setBackground(new Color(0, 0, 0, 140));

frame. add(mainPanel);

ImageIcon adImg = new ImageIcon(

"C: \\Users\\AHsan Usmani\\Desktop\\Programs\\Project GMS\\src\\WPBack. jpg");

JLabel adminImage = new JLabel(adImg, JLabel. CENTER);

adminImage. setBounds(0, 0, 1200, 850);

frame. add(adminImage);

frame. setLocationRelativeTo(null);

frame. setLayout(null);

frame. setVisible(true);

frame. setResizable(false);

frame. setDefaultCloseOperation(frame. EXIT\_ON\_CLOSE);

frame. addWindowListener(new WindowAdapter() {

public void addWindowClosed(WindowEvent e) {

frame. dispose();

}

});

// frame. setDefaultCloseOperation(frame. EXIT\_ON\_CLOSE);

frame. setLocationRelativeTo(null);

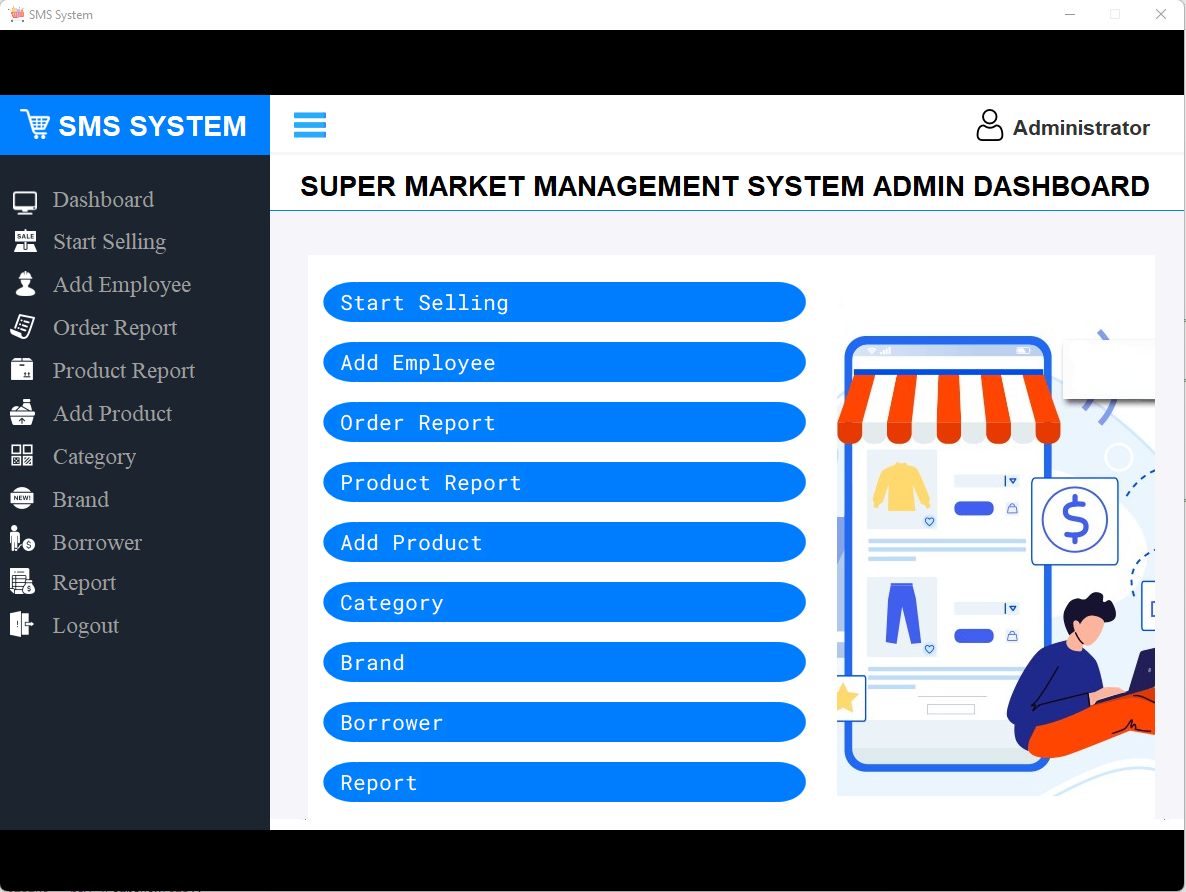
}

public static void main(String[] args) {

WelcomePage obj = new WelcomePage();

}

}

MAIN PAGE WINDOW SCREEN

**Output form information**

This is the main page of super market management system in which all the modules are used. This page in divide in several panels.

* Start Selling
* Add Employee
* Order Report
* Product Report
* Add Products
* Category
* Brand
* Borrower
* Logout.

**Coding of Main Page**

import java.text.SimpleDateFormat;

import java.awt.print.\*;

import java.util.Calendar;

import javax.swing.event.ChangeEvent;

import javax.swing.event.ChangeListener;

import javax.swing.table.DefaultTableModel;

import javax.swing.table.JTableHeader;

import javax.swing.table.TableColumn;

import javax.swing.table.TableColumnModel;

import java.util.\*;

import javax.swing.\*;

import java.awt.\*;

import java.sql.\*;

public class MainPage {

JPanel dashboardP;

JTabbedPane TP = new JTabbedPane();

JPanel upperline, buttomline;

JFrame frame = new JFrame("SMS System");

JLabel smsText;

JLabel error1, error2, error3, error4, error5, error9;

JTextField costumerName, costumerNum;

int am=0;

int Amounttemp=0;

JDialog printBill;

MainPage() {

frame.setLayout(null);

Image icon = Toolkit.getDefaultToolkit().getImage("C:\\Users\\AHsan Usmani\\Desktop\\Programs\\Project GMS\\src\\su1.png");

frame.setIconImage(icon);

JLabel MobileNum = new JLabel();

JLabel CostumerName = new JLabel();

JLabel OrderDate = new JLabel();

JLabel OrderId = new JLabel();

JLabel OrderId1 = new JLabel();

JLabel OrderDate1 = new JLabel();

JLabel costumerName1 = new JLabel();

JLabel ContactNumber1 = new JLabel();

JLabel OrderStats = new JLabel();

JTextField SearchProduct = new JTextField();

JLabel TotalAmount = new JLabel();

JLabel TotalAmount1 = new JLabel();

JLabel TotalAmount2 = new JLabel();

String O2col[] = {"ID", "Product Name", "Total units", "Price", "Total Cost", "Action" };

DefaultTableModel O2rdmodel = new DefaultTableModel();

JTable O2t = new JTable(O2rdmodel);

JTableHeader o2th = O2t.getTableHeader();

DefaultTableModel Ordmodel = new DefaultTableModel();

JTable Ot = new JTable(Ordmodel); // Ot=order table

JTableHeader oth = Ot.getTableHeader(); //oth= order table header

String sta[] = { "Payment method", "Cash", "Borrow", "Credit card", "Debit card", "Online Payment","Other Method" };

JComboBox status = new JComboBox(sta);

JLabel TA1 = new JLabel("Total Amount"); // TA1=total amount 1

JLabel text14 = new JLabel("Total Amount:");

JSeparator sp3 = new JSeparator();// Below the main button separator

sp3.setBounds(270, 181, 1200, 44);

sp3.setBorder(BorderFactory.createLineBorder(new Color(245, 245, 250), 44));

frame.add(sp3);

JSeparator sp4 = new JSeparator();// Right side Separator

sp4.setBounds(270, 181, 38, 608);

sp4.setBorder(BorderFactory.createLineBorder(new Color(245, 245, 250), 38));

sp4.setOrientation(SwingConstants.VERTICAL);

frame.add(sp4);

JSeparator sp5 = new JSeparator();// Left side Separator

sp5.setBounds(1155, 181, 36, 608);

sp5.setBorder(BorderFactory.createLineBorder(new Color(245, 245, 250), 30));

sp5.setOrientation(SwingConstants.VERTICAL);

frame.add(sp5);

JSeparator sp6 = new JSeparator();// JTabbedPane Buttum line separator

sp6.setBounds(300, 790, 870, 10);

sp6.setBorder(BorderFactory.createLineBorder(new Color(255, 255, 255), 10));

frame.add(sp6);

// TP=new JTabbedPane();

TP.setBounds(300, 197, 870, 600); // TP=TabbedPane

TP.setBackground(Color.white);

TP.setForeground(Color.black);

TP.setBorder(BorderFactory.createLineBorder(new Color(245, 245, 250), 5));

UIManager.put("TabbedPane.contentAreaColor", Color.white);

UIManager.put("TP.selected", Color.black);

frame.add(TP);

upperline = new JPanel();

upperline.setBounds(0, 0, 1200, 65);

upperline.setBackground(Color.black);

frame.add(upperline);

buttomline = new JPanel();

buttomline.setBounds(0, 800, 1200, 65);

buttomline.setBackground(Color.black);

frame.add(buttomline);

JLabel menu = new JLabel(new ImageIcon("C:\\Users\\AHsanUsmani\\Desktop\\Programs\\Project GMS\\src\\menu.png"));

menu.setBounds(290, 75, 40, 40);

frame.add(menu);

JLabel adminicon = new JLabel(

new ImageIcon("C:\\Users\\AHsan Usmani\\Desktop\\Programs\\ProjectGMS\\src\\user.png"));

adminicon.setBounds(970, 75, 40, 40);

frame.add(adminicon);

JLabel username = new JLabel("Administrator");

username.setFont(new Font("LucidaGrande", Font.BOLD, 21));

username.setBounds(1013, 77, 200, 40);

frame.add(username);

JSeparator sp1 = new JSeparator();// SP in Separator below of Black

sp1.setBounds(270, 122, 1200, 3);

sp1.setBorder(BorderFactory.createLineBorder(new Color(245, 245, 250), 3));

frame.add(sp1);

JPanel BPanel = new JPanel();

JLabel dashIcon2 = new JLabel(new ImageIcon("C:\\Users\\AHsan

Usmani\\Desktop\\Programs\\Project GMS\\src\\dash2.png"));

dashIcon2.setBounds(10, 158, 30, 30);

frame.add(dashIcon2);

JButton dashText = new JButton("Dashboard");

dashText.setBounds(50, 158, 107, 23);

dashText.setForeground(new Color(160, 160, 160));

dashText.setBackground(new Color(28, 37, 47));

dashText.setBorder(BorderFactory.createLineBorder(new Color(28, 37, 47), 3));

dashText.setFont(new Font("Times New Roman", Font.PLAIN, 23));

frame.add(dashText);

dashText.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

smsText.setText("SUPER MARKET MANAGEMENT SYSTEM ADMIN DASHBOARD ");

TP.setSelectedIndex(0); }

});

JLabel saleIcon = new JLabel(

new ImageIcon("C:\\Users\\AHsan Usmani\\Desktop\\Programs\\ProjectGMS\\src\\sale.png"));

saleIcon.setBounds(10, 195, 30, 30);

frame.add(saleIcon);

JButton sale = new JButton("Start Selling");

sale.setBounds(50, 195, 119, 32);

sale.setForeground(new Color(160, 160, 160));

sale.setBackground(new Color(28, 37, 47));

sale.setBorder(BorderFactory.createLineBorder(new Color(28, 37, 47), 3));

sale.setFont(new Font("Times New Roman", Font.PLAIN, 23));

frame.add(sale);

sale.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

smsText.setText("SELL DASHBOARD");

costumerName.setText(" Enter Costumer Name");

costumerNum.setText(" Enter Costumer Number");

TP.setSelectedIndex(1); }

});

JLabel EmployeeIcon = new JLabel(new ImageIcon("C:\\Users\\AHsan

Usmani\\Desktop\\Programs\\ProjectGMS\\src\\employee.png"));

EmployeeIcon.setBounds(10, 238, 30, 30);

frame.add(EmployeeIcon);

JButton employee = new JButton("Add Employee");

employee.setBounds(50, 238, 144, 32);

employee.setForeground(new Color(160, 160, 160));

employee.setBackground(new Color(28, 37, 47));

employee.setBorder(BorderFactory.createLineBorder(new Color(28, 37, 47), 3));

employee.setFont(new Font("Times New Roman", Font.PLAIN, 23));

frame.add(employee);

employee.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

smsText.setText("EMPLOYEE DASHBOARD");

TP.setSelectedIndex(4); }

});

JLabel orderIcon = new JLabel(new ImageIcon("C:\\Users\\AHsan

Usmani\\Desktop\\Programs\\Project GMS\\src\\order.png"));

orderIcon.setBounds(10, 281, 25, 30);

frame.add(orderIcon);

JButton order = new JButton("Order Report");

order.setBounds(50, 281, 130, 32);

order.setForeground(new Color(160, 160, 160));

order.setBackground(new Color(28, 37, 47));

order.setBorder(BorderFactory.createLineBorder(new Color(28, 37, 47), 3));

order.setFont(new Font("Times New Roman", Font.PLAIN, 23));

frame.add(order);

order.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

smsText.setText("ALL ORDER REPORT");

TP.setSelectedIndex(5); }

});

JLabel productIcon = new JLabel( new ImageIcon("C:\\Users\\AHsan

Usmani\\Desktop\\Programs\\Project GMS\\src\\product.png"));

productIcon.setBounds(10, 324, 25, 30);

frame.add(productIcon);

JButton product = new JButton("Product Report");

product.setBounds(50, 324, 148, 32);

product.setForeground(new Color(160, 160, 160));

product.setBackground(new Color(28, 37, 47));

product.setBorder(BorderFactory.createLineBorder(new Color(28, 37, 47), 3));

product.setFont(new Font("Times New Roman", Font.PLAIN, 23));

frame.add(product);

product.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

smsText.setText("ALL PRODUCT REPORT");

TP.setSelectedIndex(6); }

});

JLabel addproducticon = new JLabel( new ImageIcon("C:\\Users\\AHsan

Usmani\\Desktop\\Programs\\Project GMS\\src\\addproduct.png"));

addproducticon.setBounds(10, 367, 25, 30);

frame.add(addproducticon);

JButton addproduct = new JButton("Add Product");

addproduct.setBounds(50, 367, 125, 32);

addproduct.setForeground(new Color(160, 160, 160));

addproduct.setBackground(new Color(28, 37, 47));

addproduct.setBorder(BorderFactory.createLineBorder(new Color(28, 37, 47), 3));

addproduct.setFont(new Font("Times New Roman", Font.PLAIN, 23));

frame.add(addproduct);

addproduct.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

smsText.setText("ADD NEW PRODUCT");

TP.setSelectedIndex(7); }

});

JLabel categIcon = new JLabel( new ImageIcon("C:\\Users\\AHsan Usmani\\Desktop\\Programs\\Project GMS\\src\\category.png"));

categIcon.setBounds(10, 410, 25, 30);

frame.add(categIcon);

JButton Bcategory = new JButton("Category");

Bcategory.setBounds(50, 410, 89, 32);

Bcategory.setForeground(new Color(160, 160, 160));

Bcategory.setBackground(new Color(28, 37, 47));

Bcategory.setBorder(BorderFactory.createLineBorder(new Color(28, 37, 47), 3));

Bcategory.setFont(new Font("Times New Roman", Font.PLAIN, 23));

frame.add(Bcategory);

Bcategory.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

smsText.setText("ALL CATEGORY REPORT");

TP.setSelectedIndex(8);}

});

JLabel brandIcon = new JLabel(new ImageIcon("C:\\Users\\AHsan

Usmani\\Desktop\\Programs\\Project GMS\\src\\brand.png"));

brandIcon.setBounds(10, 453, 25, 30);

frame.add(brandIcon);

JButton Bbrand = new JButton("Brand");

Bbrand.setBounds(50, 453, 62, 32);

Bbrand.setForeground(new Color(160, 160, 160));

Bbrand.setBackground(new Color(28, 37, 47));

Bbrand.setBorder(BorderFactory.createLineBorder(new Color(28, 37, 47), 3));

Bbrand.setFont(new Font("Times New Roman", Font.PLAIN, 23));

frame.add(Bbrand);

Bbrand.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

smsText.setText("ALL BRAND REPORT");

TP.setSelectedIndex(9); }

});

JLabel borrowerIcon = new JLabel(new ImageIcon("C:\\Users\\AHsan

Usmani\\Desktop\\Programs\\Project GMS\\src\\borrower.png"));

borrowerIcon.setBounds(10, 493, 25, 30);

frame.add(borrowerIcon);

JButton borrow = new JButton("Borrower");

borrow.setBounds(50, 496, 95, 32);

borrow.setForeground(new Color(160, 160, 160));

borrow.setBackground(new Color(28, 37, 47));

borrow.setBorder(BorderFactory.createLineBorder(new Color(28, 37, 47), 3));

borrow.setFont(new Font("Times New Roman", Font.PLAIN, 23));

frame.add(borrow);

borrow.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

smsText.setText("BORROW CUSTOMERS");

TP.setSelectedIndex(10); }

});

JLabel reportIcon = new JLabel(new ImageIcon("C:\\Users\\AHsan

Usmani\\Desktop\\Programs\\Project GMS\\src\\report.png"));

reportIcon.setBounds(10, 536, 25, 30);

frame.add(reportIcon);

JButton report = new JButton("Report");

report.setBounds(50, 536, 69, 32);

report.setForeground(new Color(160, 160, 160));

report.setBackground(new Color(28, 37, 47));

report.setBorder(BorderFactory.createLineBorder(new Color(28, 37, 47), 3));

report.setFont(new Font("Times New Roman", Font.PLAIN, 23));

frame.add(report);

report.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) { smsText.setText("ALL REPORT OF SUPER MARKET MANAGEMENT SYSTEM"); }

});

JLabel logoutIcon = new JLabel(new ImageIcon("C:\\Users\\AHsan

Usmani\\Desktop\\Programs\\Project GMS\\src\\logout.png"));

logoutIcon.setBounds(10, 579, 25, 30);

frame.add(logoutIcon);

JButton logout = new JButton("Logout");

logout.setBounds(50, 579, 72, 32);

logout.setForeground(new Color(160, 160, 160));

logout.setBackground(new Color(28, 37, 47));

logout.setBorder(BorderFactory.createLineBorder(new Color(28, 37, 47), 3));

logout.setFont(new Font("Times New Roman", Font.PLAIN, 23));

frame.add(logout);

logout.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

Login obj = new Login();

frame.dispose(); }

});

BPanel.setBackground(new Color(28, 37, 47));

BPanel.setBounds(0, 125, 270, 835);

frame.add(BPanel);

smsText = new JLabel("SUPER MARKET MANAGEMENT SYSTEM ADMIN DASHBOARD");

smsText.setFont(new Font("LucidaGrande", Font.BOLD, 28));

smsText.setForeground(Color.black);

smsText.setBounds(300, 120, 900, 70);

frame.add(smsText);

JSeparator sp2 = new JSeparator();// SP in Separator below of Black

sp2.setBounds(270, 180, 1200, 1);

sp2.setBorder(BorderFactory.createLineBorder(new Color(0, 128, 255), 1));

frame.add(sp2);

JLabel basket = new JLabel(new ImageIcon("C:\\Users\\AHsan

Usmani\\Desktop\\Programs\\Project GMS\\src\\basket.png"));

basket.setBounds(13, 72, 45, 45);

frame.add(basket);

JLabel sms = new JLabel("SMS SYSTEM");

sms.setFont(new Font("LucidaGrande", Font.BOLD, 28));

sms.setForeground(Color.white);

sms.setBounds(58, 60, 300, 70);

frame.add(sms);

JPanel sidetext = new JPanel();

sidetext.setBounds(0, 65, 270, 60);

sidetext.setBackground(new Color(0, 128, 255));

frame.add(sidetext);

frame.setSize(1200, 900);

frame.setVisible(true);

frame.setResizable(false);

frame.setLocationRelativeTo(null);

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

frame.getContentPane().setBackground(Color.white);

}

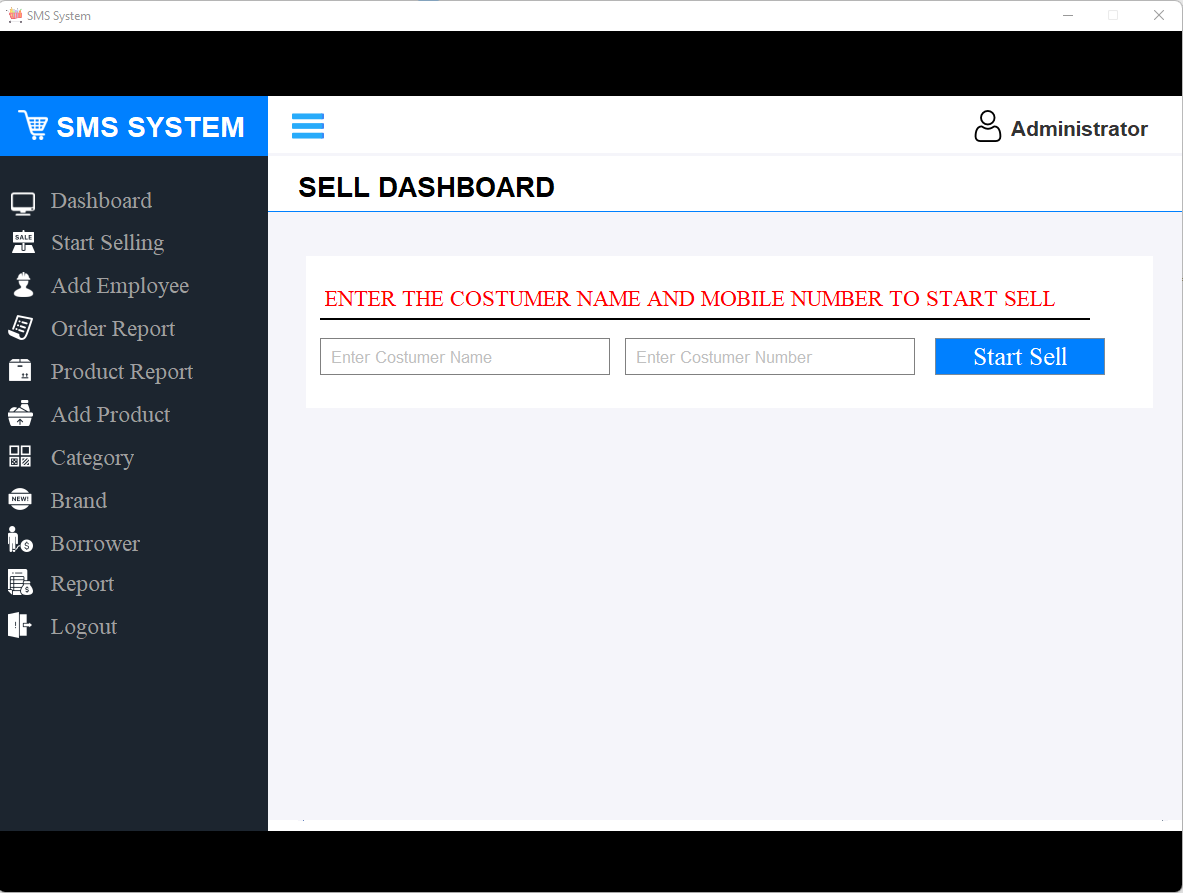
public static void main(String[] args) {

MainPage obj = new MainPage();

}

}

START SELLING WINDOW SCREEN

****

**Output form information**

Selling page of super market management system. It’s hava 5 panels. These panels are working together.This panal are used for entering the customer details.

**Coding of Customer Details Panel**

JPanel selling = new JPanel();

selling = new JPanel();

selling.setBackground(Color.white);

selling.setLayout(null);

TP.add(selling);

JLabel text1 = new JLabel("ENTER THE COSTUMER NAME AND MOBILE NUMBER TO START SELL");

text1.setBounds(20, 0, 750, 80);

text1.setForeground(Color.red);

text1.setFont(new Font("Times New Roman", Font.PLAIN, 23));

selling.add(text1);

JSeparator sellSep = new JSeparator();

sellSep.setBounds(15, 60, 770, 2);

sellSep.setBorder(BorderFactory.createLineBorder(new Color(0, 0, 0), 2));

selling.add(sellSep);

JLabel Cerror = new JLabel(" \u274C Enter only characters");

Cerror.setForeground(Color.red);

Cerror.setFont(new Font("LucidaGrande", Font.PLAIN, 12));

Cerror.setBounds(20, 118, 250, 25);

Cerror.setVisible(false);

selling.add(Cerror);

JTextField costumerName = new JTextField(" Enter Costumer Name");

costumerName.setBounds(15, 80, 290, 37);

costumerName.setBorder(BorderFactory.createLineBorder(new Color(128, 128, 128)));

costumerName.setForeground(new Color(192, 192, 192));

costumerName.setFont(new Font("LucidaGrande", Font.PLAIN, 17));

selling.add(costumerName);

costumerName.addKeyListener(new KeyAdapter() {

public void keyPressed(KeyEvent e) {

if(e.getKeyChar()>='a' && e.getKeyChar()<='z' || e.getKeyChar()>= 'A' && e.getKeyChar() <= 'Z' || e.getKeyChar()==8 || e.getKeyChar()==127 || e.getKeyChar()=='\u0020' ) {

costumerName.setEditable(true);

Cerror.setVisible(false); }

else {

costumerName.setEditable(false);

Cerror.setVisible(true); }

}

});

JLabel error6 = new JLabel(" \u274C Enter the Costumer Name");

error6.setForeground(Color.red);

error6.setFont(new Font("LucidaGrande", Font.PLAIN, 12));

error6.setBounds(20, 118, 250, 25);

error6.setVisible(false);

selling.add(error6);

costumerName.addKeyListener(new KeyAdapter() {

public void keyTyped(KeyEvent e) {

if (costumerName.getText().equals(" Enter Costumer Name")) {

costumerName.setText(""); }

}

});

costumerName.addFocusListener(new FocusListener() {

public void focusGained(FocusEvent e) {}

public void focusLost(FocusEvent e) {

if (costumerName.getText().length() == 0) {

costumerName.setText(" Enter Costumer Name"); }

}

});

costumerName.addMouseListener(new MouseAdapter() {

public void mouseClicked(MouseEvent e) {

costumerName.setBorder(BorderFactory.createLineBorder(new Color(128, 128, 128)));

error6.setVisible(false); }

});

JLabel Nerror = new JLabel(" \u274C Enter only numbers (0 - 9)");

Nerror.setForeground(Color.red);

Nerror.setFont(new Font("LucidaGrande", Font.PLAIN, 12));

Nerror.setBounds(325, 118, 250, 25);

Nerror.setVisible(false);

selling.add(Nerror);

JTextField costumerNum = new JTextField(" Enter Costumer Number");

costumerNum.setBounds(320, 80, 290, 37);

costumerNum.setBorder(BorderFactory.createLineBorder(new Color(128, 128, 128)));

costumerNum.setForeground(new Color(192, 192, 192));

costumerNum.setFont(new Font("LucidaGrande", Font.PLAIN, 17));

selling.add(costumerNum);

costumerNum.addKeyListener(new KeyAdapter() {

public void keyPressed(KeyEvent e) {

if(e.getKeyChar()>='0' && e.getKeyChar()<='9' || e.getKeyChar()==8 || e.getKeyChar()==127 || e.getKeyChar()=='\u0020' ) {

costumerNum.setEditable(true);

Nerror.setVisible(false); }

else {

costumerNum.setEditable(false);

Nerror.setVisible(true); }

}

});

JLabel error7 = new JLabel(" \u274C Enter contact number");

error7.setForeground(Color.red);

error7.setFont(new Font("LucidaGrande", Font.PLAIN, 12));

error7.setBounds(325, 118, 250, 25);

error7.setVisible(false);

selling.add(error7);

costumerNum.addKeyListener(new KeyAdapter() {

public void keyTyped(KeyEvent e) {

if (costumerNum.getText().equals(" Enter Costumer Number")) {

costumerNum.setText(""); }

}

});

costumerNum.addFocusListener(new FocusListener() {

public void focusGained(FocusEvent e) {}

public void focusLost(FocusEvent e) {

if (costumerNum.getText().length() == 0) {

costumerNum.setText(" Enter Costumer Number"); }

}

});

costumerNum.addMouseListener(new MouseAdapter() {

public void mouseClicked(MouseEvent e) {

costumerNum.setBorder(BorderFactory.createLineBorder(new Color(128, 128, 128)));

error7.setVisible(false); }

});

JButton startsell = new JButton("Start Sell");

startsell.setBounds(630, 80, 170, 37);

startsell.setBackground(new Color(0, 128, 255));

startsell.setForeground(Color.white);

startsell.setFont(new Font("Times New Roman", Font.PLAIN, 26));

selling.add(startsell);

startsell.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

SearchProduct.setText("");

Ordmodel.setRowCount(0);

if (costumerName.getText().equals(" Enter Costumer Name")) {

costumerName.setBorder(BorderFactory.createLineBorder(Color.red, 1));

error6.setVisible(true); }

else if (costumerNum.getText().equals(" Enter Costumer Number")) { costumerNum.setBorder(BorderFactory.createLineBorder(Color.red, 1));

error7.setVisible(true);}

else if(!(costumerNum.getText().length()==10)) {

JOptionPane.showMessageDialog(null, "Please enter a valid number", "WARNING", JOptionPane.ERROR\_MESSAGE); }

else

try {

Connection con = DriverManager.getConnection("jdbc:ucanaccess://C:\\Users\\AHsan

Usmani\\Desktop\\Programs\\Project GMS\\src\\sms.accdb");

Statement st = con.createStatement();

ResultSet res = st.executeQuery("Select max(Order\_Id) from Costumers");

if (res.next()) {

int id = res.getInt(1);

++id;

Calendar calendar = Calendar.getInstance();

SimpleDateFormat simpledate = new SimpleDateFormat("E,dd MMM,YY hh:mm aa");

String date = simpledate.format(calendar.getTime());

System.out.println("Date & Time is " + date);

int i = st.executeUpdate("insert into Costumers(Order\_Id,Cos\_Name,Cos\_number,Ord\_Date) values('" + id + "', '" + costumerName.getText() + "', '" + costumerNum.getText() + "', '" + date+ "')");

ResultSet rus = st.executeQuery("select \* from costumers where Order\_Id=(select max(Order\_Id) from Costumers)");

if (rus.next()) {

OrderId.setText(rus.getString(2));

CostumerName.setText(rus.getString(3));

MobileNum.setText(rus.getString(4));

OrderDate.setText(rus.getString(5)); }

TP.setSelectedIndex(2);

TotalAmount.setText("");

status.setSelectedIndex(0); }

} catch (Exception evt) { System.out.println(evt); }

}

});

JSeparator sellSep2 = new JSeparator();

sellSep2.setBounds(0, 150, 870, 625);

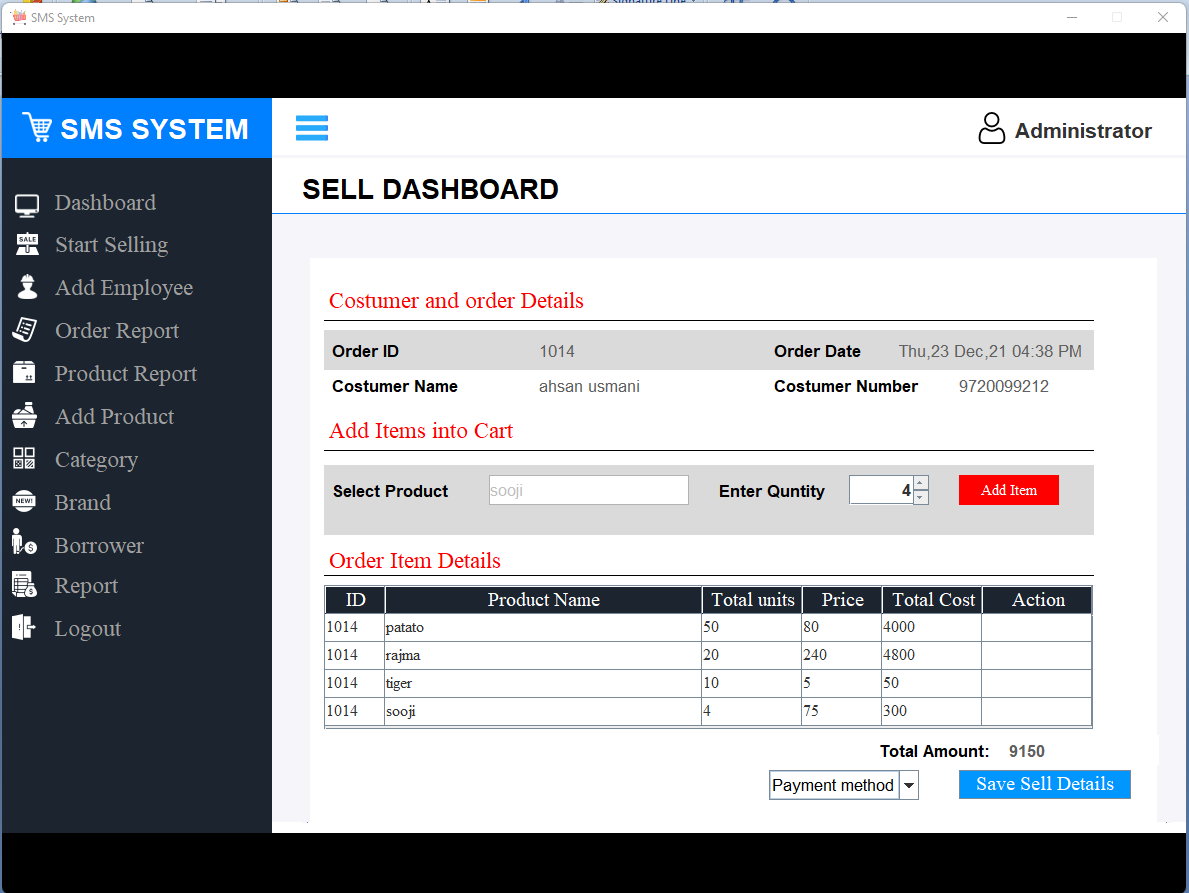
sellSep2.setBorder(BorderFactory.createLineBorder(new Color(245, 245, 250), 625));

selling.add(sellSep2);

TP.add(selling);

// Ending start selling panel

MAIN SELLING WINDOW SCREEN



**Output form information**

Main selling page of super market management system in which operator search products and add them into the table and select the payment method and click on save sell details.

**Coding of Main Selling Panel**

JPanel Mainselling = new JPanel();

Mainselling.setBackground(Color.white);

Mainselling.setLayout(null);

TP.add(Mainselling);

am=0;

JLabel text2 = new JLabel("Costumer and order Details");

text2.setBounds(20, 0, 750, 80);

text2.setForeground(Color.red);

text2.setFont(new Font("Times New Roman", Font.PLAIN, 23));

Mainselling.add(text2);

JSeparator sellSep3 = new JSeparator();

sellSep3.setBounds(15, 60, 770, 1);

sellSep3.setBorder(BorderFactory.createLineBorder(new Color(0, 0, 0), 1));

Mainselling.add(sellSep3);

JLabel OID = new JLabel("Order ID"); // OID=Oder ID

OID.setBounds(23, 75, 80, 30);

OID.setForeground(Color.BLACK);

OID.setFont(new Font("LucidaGrande", Font.BOLD, 17));

Mainselling.add(OID);

OrderId.setBounds(230, 75, 150, 30);

OrderId.setForeground(new Color(96, 96, 96));

OrderId.setFont(new Font("LucidaGrande", Font.PLAIN, 17));

Mainselling.add(OrderId);

JLabel OD = new JLabel("Order Date"); // od=Order Date

OD.setBounds(465, 75, 110, 30);

OD.setForeground(Color.BLACK);

OD.setFont(new Font("LucidaGrande", Font.BOLD, 17));

Mainselling.add(OD);

OrderDate.setBounds(590, 75, 300, 30);

OrderDate.setForeground(new Color(96, 96, 96));

OrderDate.setFont(new Font("LucidaGrande", Font.PLAIN, 17));

Mainselling.add(OrderDate);

JLabel CosName = new JLabel("Costumer Name"); // od=Order Date

CosName.setBounds(23, 110, 130, 30);

CosName.setForeground(Color.BLACK);

CosName.setFont(new Font("LucidaGrande", Font.BOLD, 17));

Mainselling.add(CosName);

CostumerName.setBounds(230, 110, 150, 30);

CostumerName.setForeground(new Color(96, 96, 96));

CostumerName.setFont(new Font("LucidaGrande", Font.PLAIN, 17));

Mainselling.add(CostumerName);

JLabel MN = new JLabel("Costumer Number"); // Mobile Number

MN.setBounds(465, 110, 150, 30);

MN.setForeground(Color.BLACK);

MN.setFont(new Font("LucidaGrande", Font.BOLD, 17));

Mainselling.add(MN);

MobileNum.setBounds(650, 110, 150, 30);

MobileNum.setForeground(new Color(96, 96, 96));

MobileNum.setFont(new Font("LucidaGrande", Font.PLAIN, 17));

Mainselling.add(MobileNum);

JSeparator sellSep6 = new JSeparator();

sellSep6.setBounds(15, 70, 770, 40);

sellSep6.setBorder(BorderFactory.createLineBorder(new Color(218, 218, 218), 40));

Mainselling.add(sellSep6);

JLabel text3 = new JLabel("Add Items into Cart");

text3.setBounds(20, 130, 750, 80);

text3.setForeground(Color.red);

text3.setFont(new Font("Times New Roman", Font.PLAIN, 23));

Mainselling.add(text3);

JSeparator sellSep4 = new JSeparator();

sellSep4.setBounds(15, 190, 770, 1);

sellSep4.setBorder(BorderFactory.createLineBorder(new Color(0, 0, 0), 1));

Mainselling.add(sellSep4);

JLabel SP = new JLabel("Select Product"); // SP= Select Product

SP.setBounds(23, 215, 122, 30);

SP.setForeground(Color.BLACK);

SP.setFont(new Font("LucidaGrande", Font.BOLD, 17));

Mainselling.add(SP);

SearchProduct.setBounds(180, 215, 200, 30);

SearchProduct.setBorder(BorderFactory.createLineBorder(new Color(179, 179, 179)));

SearchProduct.setForeground(new Color(192, 192, 192));

SearchProduct.setFont(new Font("LucidaGrande", Font.PLAIN, 17));

Mainselling.add(SearchProduct);

try {

Set <String>s;

s=new TreeSet<String>();

Connection con=DriverManager.getConnection("jdbc:ucanaccess://C:\\Users\\AHsan

Usmani\\Desktop\\Programs\\Project GMS\\src\\sms.accdb");

Statement st=con.createStatement();

ResultSet res=st.executeQuery("select \* from Product");

while(res.next()) {

s.add(res.getString("Product\_Name")); }

SearchProduct.addKeyListener(new KeyAdapter() {

public void keyReleased(KeyEvent e) {

if(e.getKeyCode()==KeyEvent.VK\_BACK\_SPACE||e.getKeyCode()==KeyEvent.VK\_DELETE) {}

else {

String check=SearchProduct.getText();

int String\_length=check.length();

for(String data:s) {

String check\_from\_data="";

for(int i=0;i<String\_length;i++) {

if(String\_length<=data.length()) {

check\_from\_data=check\_from\_data+data.charAt(i); }

}

if(check\_from\_data.equals(check)) {

SearchProduct.setText(data);

SearchProduct.setSelectionStart(String\_length);

SearchProduct.setSelectionEnd(data.length());

break; }

}

}

}

});

}

catch(Exception e) {System.out.println(e); }

JLabel qun = new JLabel("Enter Quntity"); // qun=Quantity

qun.setBounds(410, 215, 122, 30);

qun.setForeground(Color.BLACK);

qun.setFont(new Font("LucidaGrande", Font.BOLD, 17));

Mainselling.add(qun);

JLabel spinnertempLabel=new JLabel();

SpinnerModel sp = new SpinnerNumberModel(1, 1, 1000000, 1);

JSpinner spinner = new JSpinner(sp);

spinner.setFont(new Font("LucidaGrande", Font.BOLD, 17));

spinner.setBounds(540, 215, 80, 30);

Mainselling.add(spinner);

spinnertempLabel.setText(spinner.getValue().toString());

spinner.addChangeListener(new ChangeListener() {

public void stateChanged(ChangeEvent e) {

String tempSp=spinner.getValue().toString();

spinnertempLabel.setText(tempSp); }

});

JButton AddItemB = new JButton("Add Item");

AddItemB.setBounds(650, 215, 100, 30);

AddItemB.setBackground(Color.red);

AddItemB.setBorder(BorderFactory.createLineBorder(Color.red));

AddItemB.setForeground(Color.white);

AddItemB.setFont(new Font("Times New Roman", Font.PLAIN, 15));

Mainselling.add(AddItemB);

TotalAmount.setBounds(700, 475, 150, 30);

TotalAmount.setForeground(new Color(96, 96, 96));

TotalAmount.setFont(new Font("LucidaGrande", Font.BOLD, 17));

Mainselling.add(TotalAmount);

JSeparator sellSep5 = new JSeparator();

sellSep5.setBounds(15, 205, 770, 70);

sellSep5.setBorder(BorderFactory.createLineBorder(new Color(218, 218, 218), 70));

Mainselling.add(sellSep5);

JLabel text4 = new JLabel("Order Item Details");

text4.setBounds(20, 260, 400, 80);

text4.setForeground(Color.red);

text4.setFont(new Font("Times New Roman", Font.PLAIN, 23));

Mainselling.add(text4);

JSeparator sellSep7 = new JSeparator();

sellSep7.setBounds(15, 315, 770, 1);

sellSep7.setBorder(BorderFactory.createLineBorder(new Color(0, 0, 0), 1));

Mainselling.add(sellSep7);

String Ocol[] = {"ID", "Product Name", "Total units", "Price", "Total Cost", "Action" };

Ordmodel.setColumnIdentifiers(Ocol);

Ot.setRowHeight(28);

Ot.setFont(new Font("Times New Roman", Font.PLAIN, 16));

oth.setFont(new Font("Times New Roman", Font.PLAIN, 20));

oth.setForeground(Color.white);

oth.setBackground(new Color(28, 37, 47));

oth.getColumnModel().getColumn(0).setMaxWidth(60); oth.getColumnModel().getColumn(2).setMaxWidth(100);

oth.getColumnModel().getColumn(2).setMinWidth(100);

oth.getColumnModel().getColumn(3).setMaxWidth(80); oth.getColumnModel().getColumn(3).setMinWidth(80); oth.getColumnModel().getColumn(4).setMaxWidth(100); oth.getColumnModel().getColumn(4).setMinWidth(100); oth.getColumnModel().getColumn(5).setMaxWidth(110); oth.getColumnModel().getColumn(5).setMinWidth(110); Ot.getTableHeader().setReorderingAllowed(false);

JScrollPane Ordpane = new JScrollPane();

Ordpane.setViewportView(Ot);

Ordpane.setBounds(15, 325, 770, 145);

Mainselling.add(Ordpane);

AddItemB.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

try {

Connection con = DriverManager.getConnection("jdbc:ucanaccess://C:\\Users\\AHsan

Usmani\\Desktop\\Programs\\Project GMS\\src\\sms.accdb");

Statement st=con.createStatement();

ResultSet res3=st.executeQuery("select Product\_Name from Product");

JLabel tempLabel1=new JLabel();

l1: while(res3.next()) {

if(res3.getString("Product\_Name").equals(SearchProduct.getText())) {

tempLabel1.setText("false");

break l1; }

else

tempLabel1.setText("true"); }

boolean flag=Boolean.parseBoolean(tempLabel1.getText());

JLabel tempLabel2=new JLabel();

JLabel tempLabel3=new JLabel();

JLabel tempLabel4=new JLabel();

ResultSet res4=st.executeQuery("select Total\_units from Product where Product\_Name= '"+SearchProduct.getText()+"' ");

if(res4.next()) {

String tempUnit1=res4.getString("Total\_units");

int EU=Integer.parseInt(spinnertempLabel.getText());

int TU=Integer.parseInt(tempUnit1);

if(TU==0) { tempLabel2.setText("true"); }

else

tempLabel2.setText("false");

if(EU>TU) {

tempLabel3.setText("true");

tempLabel4.setText(String.valueOf(TU)); }

else

tempLabel3.setText("false");

}

boolean flag1=Boolean.parseBoolean(tempLabel2.getText());

boolean flag2=Boolean.parseBoolean(tempLabel3.getText());

if(SearchProduct.getText().equals("")) {

ImageIcon ig1 = new ImageIcon("C:\\Users\\AHsan Usmani\\Desktop\\Programs\\Project GMS\\src\\error.png");

JOptionPane.showMessageDialog(null, "Select the Product ", "Warrning", JOptionPane.WARNING\_MESSAGE,ig1); }

else if(flag) {

JOptionPane.showMessageDialog(null, "Product not found", "Error",

JOptionPane.ERROR\_MESSAGE); }

else if(flag1) {

JOptionPane.showMessageDialog(null, SearchProduct.getText()+" is out of stock",

"WARNING", JOptionPane.ERROR\_MESSAGE); }

else if(flag2) {

JOptionPane.showMessageDialog(null,"Only "+ tempLabel4.getText()+" units are

available", "WARNING", JOptionPane.WARNING\_MESSAGE); }

else

{

JLabel tempvar2= new JLabel();

ResultSet res=st.executeQuery("select \* from Product where Product\_Name ='"+SearchProduct.getText().toLowerCase()+"'");

while(res.next()) { tempvar2.setText(res.getString("Price\_Per\_Unit")); }

int ordId=Integer.parseInt(OrderId.getText());

String ProName=SearchProduct.getText().toLowerCase();

int unit=Integer.parseInt(spinnertempLabel.getText());

int ProRate=Integer.parseInt(tempvar2.getText());

int oneProductcost=unit\*ProRate;

String action = "";

String OrderDet[] = {OrderId.getText(),ProName,spinnertempLabel.getText(),tempvar2.getText(),String.valueOf(

oneProductcost)};

Ordmodel.addRow(OrderDet);

for(int i = 0;i<Ot.getModel().getRowCount();i++) {

String Tamount = Ot.getModel().getValueAt(i, 4).toString();

int am=Integer.parseInt(Tamount);

if(i==Ot.getModel().getRowCount()-1) {

Amounttemp= Amounttemp+am;

TotalAmount.setText(String.valueOf(Amounttemp)); } } } }

catch(Exception ev) {System.out.println(ev);}

}

});

JLabel TA = new JLabel("Total Amount:"); // TA= total amount label

TA.setBounds(570, 475, 125, 30);

TA.setForeground(Color.BLACK);

TA.setFont(new Font("LucidaGrande", Font.BOLD, 17));

Mainselling.add(TA);

status.setBounds(460, 510, 150, 30);

status.setForeground(new Color(0, 0, 0));

status.setBackground(new Color(255, 255, 255));

status.setFont(new Font("LucidaGrande", Font.PLAIN, 17));

Mainselling.add(status);

JButton SSD = new JButton("Save Sell Details"); // SSD Save sell Details

SSD.setBounds(650, 510, 172, 29);

SSD.setBackground(new Color(0, 150, 255));

SSD.setForeground(Color.white);

SSD.setFont(new Font("Times New Roman", Font.PLAIN, 20));

Mainselling.add(SSD);

JLabel tempLabel1=new JLabel();

SSD.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

if (SearchProduct.getText().length() == 0) {

ImageIcon ig1 = new ImageIcon("C:\\Users\\AHsan Usmani\\Desktop\\Programs\\Project GMS\\src\\error.png");

JOptionPane.showMessageDialog(null, "Select the Product ", "Warrning", JOptionPane.WARNING\_MESSAGE,ig1); }

else if (status.getItemAt(status.getSelectedIndex()) == status.getItemAt(0)) {

ImageIcon ig1 = new ImageIcon("C:\\Users\\AHsan Usmani\\Desktop\\Programs\\Project GMS\\src\\error.png");

JOptionPane.showMessageDialog(null, "Please select the payment method", "Warrning",JOptionPane.WARNING\_MESSAGE, ig1); }

else if(Ordmodel.getRowCount()==0) {

JOptionPane.showMessageDialog(null, "Please add the product", "Warrning",JOptionPane.ERROR\_MESSAGE); }

else {

try {

tempLabel1=new JLabel();

JLabel tempLabel2=new JLabel();

Connection con = DriverManager.getConnection("jdbc:ucanaccess://C:\\Users\\AHsan Usmani\\Desktop\\Programs\\Project GMS\\src\\sms.accdb");

Statement st = con.createStatement();

if(status.getSelectedItem().toString().equals("Borrow")) {

ResultSet res4=st.executeQuery("Select \* from Borrower");

boolean flag=true;

l1:while(res4.next()) {

tempLabel1.setText(res4.getString("Bor\_Name")); tempLabel2.setText(res4.getString("Contact\_No"));

if(res4.getString("Bor\_Name").equals(costumerName.getText())&&res4.getString("Contact\_No").equals(costumerNum.getText())) {

flag=true;

break l1; }

else

flag=false; }

if(flag) {

ImageIcon img=new ImageIcon("C:\\Users\\AHsan Usmani\\Desktop\\Programs\\Project GMS\\src\\br.png");

int ta=Integer.parseInt(TotalAmount.getText());

String pa=JOptionPane.showInputDialog(null," Please enter the pay amount ","Borrow" ,JOptionPane.INFORMATION\_MESSAGE);

int PA=Integer.parseInt(pa);

int due= ta-PA;

TA1.setText("Due Amount");

text14.setText("Due Amount");

TotalAmount2.setText(String.valueOf(due));

TotalAmount1.setText(String.valueOf(due));

int b=st.executeUpdate("update Borrower set DueAmount='"+String.valueOf(due)+"' where Bor\_Name= '"+tempLabel1.getText()+"' AND Contact\_No='"+tempLabel2.getText()+"' ");

ResultSet res = st.executeQuery("select max(Order\_Id) from Costumers");

if (res.next()) {

smsText.setText("Order Item of Order ID: " + res.getInt(1));

int i = st.executeUpdate("update costumers set Payment\_Status= '" + status.getSelectedItem().toString()+ "' where Order\_Id=(select max(Order\_Id) from Costumers)");

ResultSet res1 = st.executeQuery("select \* from Costumers where Order\_Id=(select max(Order\_Id) from Costumers)");

if (res1.next()) { OrderId1.setText(res1.getString(2)); OrderDate1.setText(res1.getString(5)); costumerName1.setText(res1.getString(3)); ContactNumber1.setText(res1.getString(4));

OrderStats.setText(res1.getString(6));

TP.setSelectedIndex(3);

for(int k=0;k<Ordmodel.getRowCount();k++) {

String Oid=Ordmodel.getValueAt(k, 0).toString();

String Pname=Ordmodel.getValueAt(k, 1).toString(); String Ounits=Ordmodel.getValueAt(k, 2).toString();

String ORate=Ordmodel.getValueAt(k, 3).toString();

String OrRate=Ordmodel.getValueAt(k, 4).toString();

String ac="";

String[] o2rdetails = {Oid,Pname,Ounits,ORate,OrRate,ac};

O2rdmodel.addRow(o2rdetails);

int l=st.executeUpdate("insert into Order(Order\_ID,Product\_Name,Total\_units,Price,Total\_Cost) values('"+Oid+"', '"+Pname+"', '"+Ounits+"', '"+ORate+"', '"+OrRate+"') ");

int j=st.executeUpdate("update Costumers set Amount='"+TotalAmount.getText()+"' where Order\_Id= '"+OrderId.getText()+"' "); }

} }

for(int a=0;a<Ordmodel.getRowCount();a++) {

int EU=Integer.parseInt(Ordmodel.getValueAt(a, 2).toString());

ResultSet res3=st.executeQuery("select \* from Product where Product\_Name='"+Ordmodel

.getValueAt(a, 1).toString()+"'");

if(res3.next()) {

int TU=Integer.parseInt(res3.getString("Total\_units"));

TU=TU-EU;

int m=st.executeUpdate("update Product set Total\_units='"+String.valueOf(TU)+"' where Product\_Name= '"+Ordmodel.getValueAt(a, 1).toString()+"' "); } } }

else

JOptionPane.showMessageDialog(null, "This costumer in not borrower\nPlease select cash or other payment method.", "Warrning",JOptionPane.WARNING\_MESSAGE);

}

else {

ResultSet res = st.executeQuery("select max(Order\_Id) from Costumers");

if (res.next()) {

smsText.setText("Order Item of Order ID: " + res.getInt(1));

int i = st.executeUpdate("update costumers set Payment\_Status= '" + status.getSelectedItem().toString()+ "' where Order\_Id=(select max(Order\_Id) from Costumers)");

ResultSet res1 = st.executeQuery("select \* from Costumers where Order\_Id=(select max(Order\_Id) from Costumers)");

if (res1.next()) { OrderId1.setText(res1.getString(2)); OrderDate1.setText(res1.getString(5)); costumerName1.setText(res1.getString(3)); ContactNumber1.setText(res1.getString(4)); OrderStats.setText(res1.getString(6));

TP.setSelectedIndex(3);

for(int k=0;k<Ordmodel.getRowCount();k++) {

String Oid=Ordmodel.getValueAt(k, 0).toString();

String Pname=Ordmodel.getValueAt(k, 1).toString();

String Ounits=Ordmodel.getValueAt(k, 2).toString();

String ORate=Ordmodel.getValueAt(k, 3).toString();

String OrRate=Ordmodel.getValueAt(k, 4).toString();

String ac="";

String[] o2rdetails = {Oid,Pname,Ounits,ORate,OrRate,ac};

O2rdmodel.addRow(o2rdetails);

int l=st.executeUpdate("insert into Order(Order\_ID,Product\_Name,Total\_units,Price, Total\_Cost) values('"+Oid+"', '"+Pname+"', '"+Ounits+"', '"+ORate+"', '"+OrRate+"') ");

int j=st.executeUpdate("update Costumers set Amount='"+TotalAmount.getText()+"'

where Order\_Id= '"+OrderId.getText()+"' "); TotalAmount2.setText(TotalAmount.getText());

TotalAmount1.setText(TotalAmount.getText()); } } }

for(int a=0;a<Ordmodel.getRowCount();a++) {

int EU=Integer.parseInt(Ordmodel.getValueAt(a, 2).toString());

ResultSet res3=st.executeQuery("select \* from Product where Product\_Name='"+Ordmodel

.getValueAt(a, 1).toString()+"'");

if(res3.next()) {

int TU=Integer.parseInt(res3.getString("Total\_units"));

TU=TU-EU;

int m=st.executeUpdate("update Product set Total\_units='"+String.valueOf(TU)

+"' where Product\_Name= '"+Ordmodel.getValueAt(a, 1).toString()+"' ");

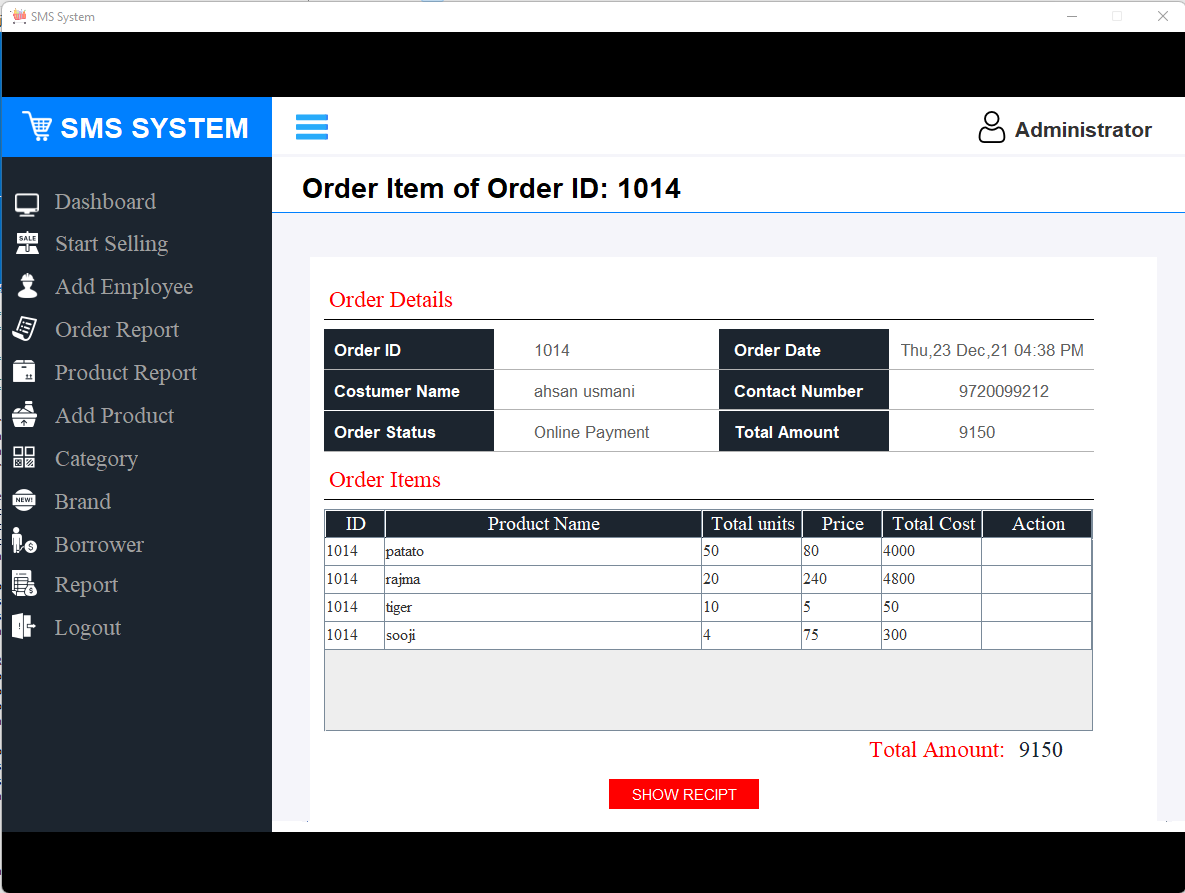
} } }

} catch (Exception evt) { System.out.println(evt); }

} } });

// Ending of selling main Panel

ORDER DETAILS WINDOW SCREEN



**Output form information**

Order details panel of super market management system. It’s show all the list of products purchased by customers as well as operator will print the recipt.

**Coding of Order Details Panel**

JPanel orderDetails = new JPanel();

orderDetails.setBackground(Color.white);

orderDetails.setLayout(null);

TP.add(orderDetails);

JLabel text5 = new JLabel("Order Details");

text5.setBounds(20, 0, 750, 80);

text5.setForeground(Color.red);

text5.setFont(new Font("Times New Roman", Font.PLAIN, 23));

orderDetails.add(text5);

JSeparator ordSep1 = new JSeparator();

ordSep1.setBounds(15, 60, 770, 1);

ordSep1.setBorder(BorderFactory.createLineBorder(new Color(0, 0, 0), 1));

orderDetails.add(ordSep1);

JLabel ORD = new JLabel("Order ID"); // ProaName= Product Name

ORD.setBounds(25, 75, 122, 30);

ORD.setForeground(Color.white);

ORD.setFont(new Font("LucidaGrande", Font.BOLD, 17));

orderDetails.add(ORD);

JSeparator ordSep2 = new JSeparator();

ordSep2.setBounds(15, 70, 170, 40);

ordSep2.setBorder(BorderFactory.createLineBorder(new Color(28, 37, 47), 40));

orderDetails.add(ordSep2);

OrderId1.setBounds(225, 75, 150, 30);

OrderId1.setForeground(new Color(96, 96, 96));

OrderId1.setFont(new Font("LucidaGrande", Font.PLAIN, 17));

orderDetails.add(OrderId1);

JLabel OD1 = new JLabel("Order Date"); // OD1= Order Date

OD1.setBounds(425, 75, 122, 30);

OD1.setForeground(Color.white);

OD1.setFont(new Font("LucidaGrande", Font.BOLD, 17));

orderDetails.add(OD1);

JSeparator ordSep3 = new JSeparator();

ordSep3.setBounds(410, 70, 170, 40);

ordSep3.setBorder(BorderFactory.createLineBorder(new Color(28, 37, 47), 40));

orderDetails.add(ordSep3);

OrderDate1.setBounds(592, 75, 300, 30);

OrderDate1.setForeground(new Color(96, 96, 96));

OrderDate1.setFont(new Font("LucidaGrande", Font.PLAIN, 17));

orderDetails.add(OrderDate1);

JSeparator ordSep4 = new JSeparator();

ordSep4.setBounds(15, 110, 770, 1);

ordSep4.setBorder(BorderFactory.createLineBorder(new Color(179, 179, 179), 1));

orderDetails.add(ordSep4);

JLabel CN1 = new JLabel("Costumer Name"); // CN1= costumer name 1

CN1.setBounds(25, 116, 150, 30);

CN1.setForeground(Color.white);

CN1.setFont(new Font("LucidaGrande", Font.BOLD, 17));

orderDetails.add(CN1);

JSeparator ordSep5 = new JSeparator();

ordSep5.setBounds(15, 111, 170, 40);

ordSep5.setBorder(BorderFactory.createLineBorder(new Color(28, 37, 47), 40));

orderDetails.add(ordSep5);

costumerName1.setBounds(225, 116, 150, 30);

costumerName1.setForeground(new Color(96, 96, 96));

costumerName1.setFont(new Font("LucidaGrande", Font.PLAIN, 17));

orderDetails.add(costumerName1);

JLabel CN2 = new JLabel("Contact Number"); // CN2 =contach Number

CN2.setBounds(425, 116, 150, 30);

CN2.setForeground(Color.white);

CN2.setFont(new Font("LucidaGrande", Font.BOLD, 17));

orderDetails.add(CN2);

JSeparator ordSep6 = new JSeparator();

ordSep6.setBounds(410, 110, 170, 40);

ordSep6.setBorder(BorderFactory.createLineBorder(new Color(28, 37, 47), 40));

orderDetails.add(ordSep6);

ContactNumber1.setBounds(650, 116, 150, 30);

ContactNumber1.setForeground(new Color(96, 96, 96));

ContactNumber1.setFont(new Font("LucidaGrande", Font.PLAIN, 17));

orderDetails.add(ContactNumber1);

JSeparator ordSep7 = new JSeparator();

ordSep7.setBounds(15, 150, 770, 1);

ordSep7.setBorder(BorderFactory.createLineBorder(new Color(179, 179, 179), 1));

orderDetails.add(ordSep7);

JLabel ORS = new JLabel("Order Status"); // ORS= Order status

ORS.setBounds(25, 157, 122, 30);

ORS.setForeground(Color.white);

ORS.setFont(new Font("LucidaGrande", Font.BOLD, 17));

orderDetails.add(ORS);

JSeparator ordSep8 = new JSeparator();

ordSep8.setBounds(15, 152, 170, 40);

ordSep8.setBorder(BorderFactory.createLineBorder(new Color(28, 37, 47), 40));

orderDetails.add(ordSep8);

OrderStats.setBounds(225, 157, 150, 30);

OrderStats.setForeground(new Color(96, 96, 96));

OrderStats.setFont(new Font("LucidaGrande", Font.PLAIN, 17));

orderDetails.add(OrderStats);

TA1.setBounds(425, 157, 150, 30);

TA1.setForeground(Color.white);

TA1.setFont(new Font("LucidaGrande", Font.BOLD, 17));

orderDetails.add(TA1);

JSeparator ordSep10 = new JSeparator();

ordSep10.setBounds(410, 152, 170, 40);

ordSep10.setBorder(BorderFactory.createLineBorder(new Color(28, 37, 47), 40));

orderDetails.add(ordSep10);

TotalAmount1.setBounds(650, 157, 150, 30);

TotalAmount1.setForeground(new Color(96, 96, 96));

TotalAmount1.setFont(new Font("LucidaGrande", Font.PLAIN, 17));

orderDetails.add(TotalAmount1);

JSeparator ordSep11 = new JSeparator();

ordSep11.setBounds(15, 192, 770, 1);

ordSep11.setBorder(BorderFactory.createLineBorder(new Color(179, 179, 179), 1));

orderDetails.add(ordSep11);

JLabel text6 = new JLabel("Order Items");

text6.setBounds(20, 180, 750, 80);

text6.setForeground(Color.red);

text6.setFont(new Font("Times New Roman", Font.PLAIN, 23));

orderDetails.add(text6);

JSeparator ordSep12 = new JSeparator();

ordSep12.setBounds(15, 240, 770, 1);

ordSep12.setBorder(BorderFactory.createLineBorder(new Color(0, 0, 0), 1));

orderDetails.add(ordSep12);

text14.setBounds(560, 450, 150, 80);

text14.setForeground(Color.red);

text14.setFont(new Font("Times New Roman", Font.PLAIN, 23));

orderDetails.add(text14);

O2rdmodel.setColumnIdentifiers(O2col);

O2t.setRowHeight(28);

O2t.setFont(new Font("Times New Roman", Font.PLAIN, 16));

o2th.setFont(new Font("Times New Roman", Font.PLAIN, 20));

o2th.setForeground(Color.white);

o2th.setBackground(new Color(28, 37, 47));

o2th.getColumnModel().getColumn(0).setMaxWidth(60);

o2th.getColumnModel().getColumn(2).setMaxWidth(100);

o2th.getColumnModel().getColumn(2).setMinWidth(100);

o2th.getColumnModel().getColumn(3).setMaxWidth(80);

o2th.getColumnModel().getColumn(3).setMinWidth(80);

o2th.getColumnModel().getColumn(4).setMaxWidth(100);

o2th.getColumnModel().getColumn(4).setMinWidth(100);

o2th.getColumnModel().getColumn(5).setMaxWidth(110);

o2th.getColumnModel().getColumn(5).setMinWidth(110);

O2t.getTableHeader().setReorderingAllowed(false);

JScrollPane O2rdpane = new JScrollPane();

O2rdpane.setViewportView(O2t);

O2rdpane.setBounds(15, 250, 770, 223);

orderDetails.add(O2rdpane);

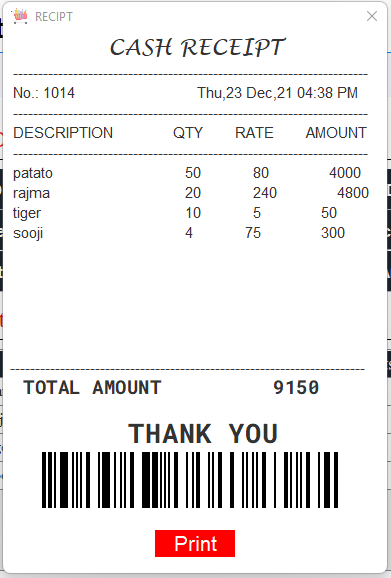
TotalAmount2.setBounds(710, 450, 750, 80);

TotalAmount2.setForeground(new Color(28, 37, 47));

TotalAmount2.setFont(new Font("Times New Roman", Font.PLAIN, 23));

orderDetails.add(TotalAmount2);

BILL RECIPT WINDOW SCREEN



**Output form information**

This panel show the bill of super market management system. And we can print the bill recipts.

**Coding of Print bill Panel**

JButton recipt = new JButton("SHOW RECIPT");

recipt.setBounds(300, 520, 150, 30);

recipt.setBackground(Color.red);

recipt.setBorder(BorderFactory.createLineBorder(Color.red));

recipt.setForeground(Color.white);

recipt.setFont(new Font("LucidaGrande", Font.PLAIN, 15));

orderDetails.add(recipt);

printBill=new JDialog(frame," RECIPT ",true);

JTextArea area=new JTextArea();

area.setFont(new Font("LucidaGrande", Font.PLAIN, 15));

area.setBounds(10,30,380,295);

printBill.add(area);

recipt.addActionListener(new ActionListener( ){

public void actionPerformed(ActionEvent ev) {

try {

String userN="ahsan";

Connection con= DriverManager.getConnection("jdbc:ucanaccess://C:\\Users\\AHsan Usmani\\Desktop\\Programs\\Project GMS\\src\\sms.accdb");

Statement st=con.createStatement();

ResultSet rs=st.executeQuery("select \* from userTable where username='"+userN+"'");

if(rs.next()) {

JLabel t1=new JLabel();

t1.setFont(new Font("Lucida Handwriting",Font.PLAIN,22));

t1.setText("CASH RECEIPT");

t1.setBounds(105,0,330,30);

printBill.add(t1);

String PName="Dal";

String Qty="5";

String rate="10";

String amount="50";

area.setText("-----------------------------------------------------------------------\n");

area.setText(area.getText()+"No.: "+OrderId1.getText()+"\t "+OrderDate1.getText()+"\n");

area.setText(area.getText()+"-----------------------------------------------------------------------\n");

area.setText(area.getText()+"DESCRIPTION\t QTY RATE AMOUNT\n");

area.setText(area.getText()+"-----------------------------------------------------------------------\n");

for(int k=0;k<Ordmodel.getRowCount();k++) {

String Oid=Ordmodel.getValueAt(k, 0).toString();

String Pname=Ordmodel.getValueAt(k, 1).toString();

String Ounits=Ordmodel.getValueAt(k, 2).toString();

String ORate=Ordmodel.getValueAt(k, 3).toString();

String OrRate=Ordmodel.getValueAt(k, 4).toString();

area.setText(area.getText()+Pname+"\t "+Ounits+" "+ORate+" "+OrRate+"\n"); }

JLabel t3=new JLabel();

t3.setFont(new Font("LucidaGrande",Font.PLAIN,15));

t3.setText("-----------------------------------------------------------------------");

t3.setBounds(7,320,400,30);

printBill.add(t3);

JLabel t2=new JLabel();

t2.setFont(new Font("Roboto Mono",Font.BOLD,19));

t2.setText("TOTAL AMOUNT ");

t2.setBounds(20,340,300,30);

printBill.add(t2);

JLabel Tamount=new JLabel();

Tamount.setFont(new Font("Roboto Mono",Font.BOLD,19));

Tamount.setText( TotalAmount2.getText());

Tamount.setBounds(270,340,120,30);

printBill.add(Tamount);

JLabel t4=new JLabel();

t4.setFont(new Font("Roboto Mono",Font.BOLD,27));

t4.setText("THANK YOU");

t4.setBounds(125,385,300,30);

printBill.add(t4);

ImageIcon im=new ImageIcon("C:\\Users\\AHsan Usmani\\Desktop\\Programs\\Project

GMS\\src\\BC.gif");

JLabel qr=new JLabel(im);

qr.setBounds(35,420,300,60);

printBill.add(qr); }

JButton printB=new JButton("Print");

printB.setBounds(152,498,80,27);

printB.setBackground(Color.red);

printB.setBorder(BorderFactory.createLineBorder(Color.red));

printB.setForeground(Color.white);

printB.setFont(new Font("LucidaGrande", Font.PLAIN, 21));

printBill.add(printB);

printB.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent ev) {

try {

area.print();

}catch(Exception e) {}

} }); }

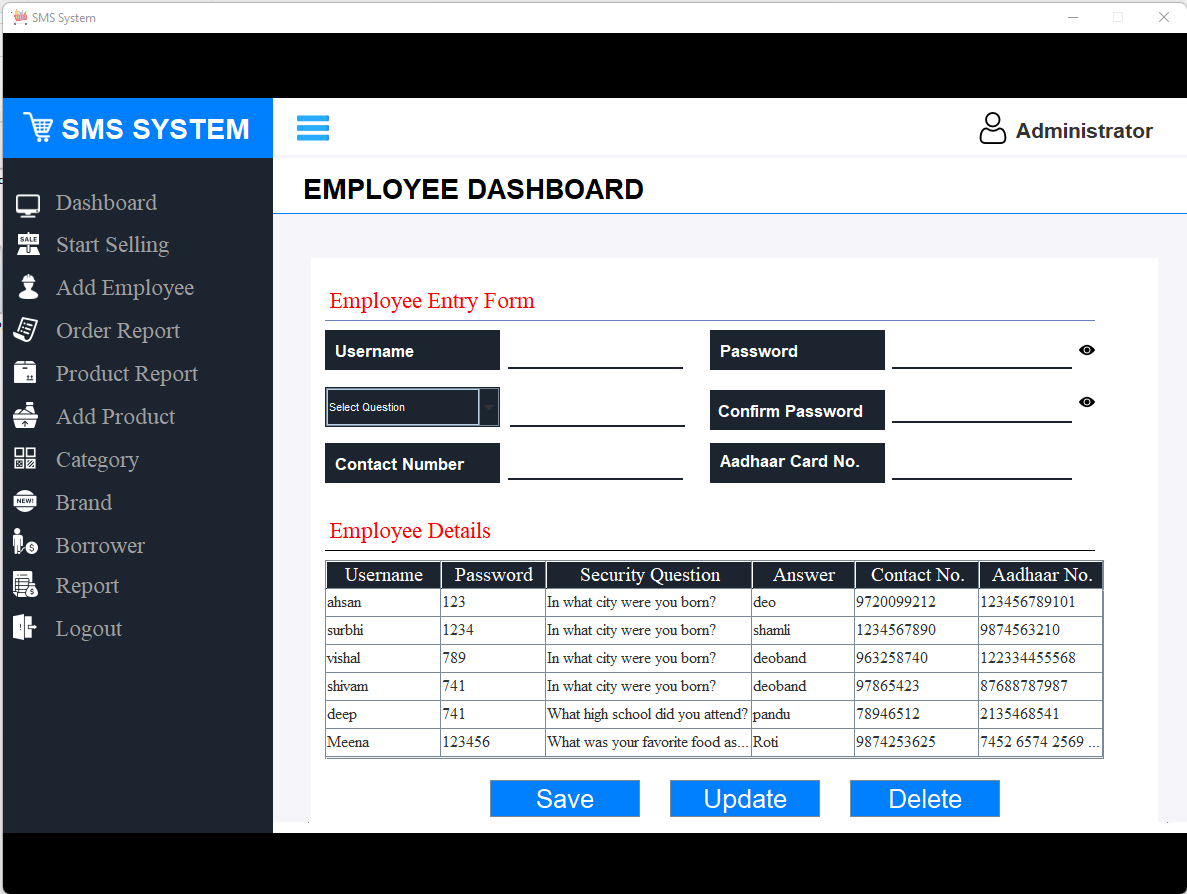
catch(Exception ex) { System.out.println(ex); }

printBill.getContentPane().setBackground(Color.WHITE);

printBill.setSize(400,580);

printBill.setLocationRelativeTo(null); printBill.setLayout(null);

printBill.setVisible(true); } });

EMPLOYEE PANEL WINDOW SCREEN

**Output form information**

Admin will add the employee and then employee have a username and password by which employee can be use the software.

**Coding of Employee Panel**

JPanel Employee = new JPanel();

Employee.setBackground(Color.white);

Employee.setLayout(null);

TP.add(Employee);

JLabel Emptext1 = new JLabel("Employee Entry Form");

Emptext1.setBounds(20, 0, 750, 80);

Emptext1.setForeground(Color.red);

Emptext1.setFont(new Font("Times New Roman", Font.PLAIN, 23));

Employee.add(Emptext1);

JSeparator EmpSeP1 = new JSeparator();

EmpSeP1.setBounds(15, 60, 770, 1);

ordSep1.setBorder(BorderFactory.createLineBorder(new Color(0, 0, 0), 1));

Employee.add(EmpSeP1);

JLabel EORD = new JLabel("Username"); // EORD= Employee order ID Label

EORD.setBounds(25, 75, 122, 30);

EORD.setForeground(Color.white);

EORD.setFont(new Font("LucidaGrande", Font.BOLD, 17));

Employee.add(EORD);

JSeparator EmpSeP2 = new JSeparator();

EmpSeP2.setBounds(15, 70, 175, 40);

EmpSeP2.setBorder(BorderFactory.createLineBorder(new Color(28, 37, 47), 40));

Employee.add(EmpSeP2);

JTextField EmpUsername = new JTextField();

EmpUsername.setBounds(200, 75, 180, 25);

EmpUsername.setBorder(BorderFactory.createLineBorder(new Color(255, 255, 255)));

EmpUsername.setForeground(new Color(192, 192, 192));

EmpUsername.setFont(new Font("LucidaGrande", Font.PLAIN, 17));

Employee.add(EmpUsername);

JSeparator EmpSeP3 = new JSeparator();

EmpSeP3.setBounds(198, 107, 175, 2);

EmpSeP3.setBorder(BorderFactory.createLineBorder(new Color(28, 37, 47), 2));

Employee.add(EmpSeP3);

EmpUsername.addMouseListener(new MouseAdapter() {

public void mouseClicked(MouseEvent e) {

EmpSeP3.setBorder(BorderFactory.createLineBorder(new Color(28, 37, 47), 2));

} });

JLabel EPD = new JLabel("Password"); // EPD= Employee Password Label

EPD.setBounds(410, 75, 122, 30);

EPD.setForeground(Color.white);

EPD.setFont(new Font("LucidaGrande", Font.BOLD, 17));

Employee.add(EPD);

JSeparator EmpSeP4 = new JSeparator();

EmpSeP4.setBounds(400, 70, 175, 40);

EmpSeP4.setBorder(BorderFactory.createLineBorder(new Color(28, 37, 47), 40));

Employee.add(EmpSeP4);

JPasswordField EmpPassword = new JPasswordField();

EmpPassword.setBounds(585, 75, 180, 25);

EmpPassword.setBorder(BorderFactory.createLineBorder(new Color(255, 255, 255)));

EmpPassword.setForeground(new Color(192, 192, 192));

EmpPassword.setEchoChar('\*');

EmpPassword.setFont(new Font("LucidaGrande", Font.PLAIN, 17));

Employee.add(EmpPassword);

JSeparator EmpSeP5 = new JSeparator();

EmpSeP5.setBounds(582, 107, 180, 2);

EmpSeP5.setBorder(BorderFactory.createLineBorder(new Color(28, 37, 47), 2));

Employee.add(EmpSeP5);

EmpPassword.addMouseListener(new MouseAdapter() {

public void mouseClicked(MouseEvent e) {

EmpSeP5.setBorder(BorderFactory.createLineBorder(new Color(28, 37, 47), 2));

} });

JLabel RshowIcon = new JLabel(

new ImageIcon("C:\\Users\\AHsan Usmani\\Desktop\\Programs\\ProjectGMS\\src\\view.png"));

RshowIcon.setBounds(762, 75, 30, 30);

JLabel RhideIcon = new JLabel(new ImageIcon("C:\\Users\\AHsanUsmani\\Desktop\\Programs\\Project GMS\\src\\hide.png"));

RhideIcon.setVisible(false);

RhideIcon.setBounds(762, 75, 30, 30);

Employee.add(RshowIcon);

Employee.add(RhideIcon);

RshowIcon.addMouseListener(new MouseAdapter() {

public void mousePressed(MouseEvent e) {

RshowIcon.setVisible(false);

RhideIcon.setVisible(true);

EmpPassword.setEchoChar((char) 0); }

public void mouseReleased(MouseEvent e) {

RhideIcon.setVisible(false);

RshowIcon.setVisible(true);

EmpPassword.setEchoChar('\*'); } });

String SecQ[] = { "Select Question", "In what city were you born?",

"What is the name of your favorite pet?",

"What is your mother's maiden name?", "What high school did you attend?",

"What is the name of your first school?", "What was the make of your first car?",

"What was your favorite food as a child?", "Where did you meet your spouse?" };

JComboBox EmpSecQuestion = new JComboBox(SecQ);

EmpSecQuestion.setBounds(15, 127, 175, 40);

EmpSecQuestion.setBorder(BorderFactory.createLineBorder(new Color(28, 37, 47)));

EmpSecQuestion.setForeground(new Color(255, 255, 255));

EmpSecQuestion.setBackground(new Color(28, 37, 47));

EmpSecQuestion.setFont(new Font("LucidaGrande", Font.PLAIN, 11));

Employee.add(EmpSecQuestion);

JTextField EmpAnswer = new JTextField("Answer");

EmpAnswer.setBounds(205, 135, 180, 25);

EmpAnswer.setBorder(BorderFactory.createLineBorder(new Color(255, 255, 255)));

EmpAnswer.setForeground(new Color(192, 192, 192));

EmpAnswer.setFont(new Font("LucidaGrande", Font.PLAIN, 17));

Employee.add(EmpAnswer);

JSeparator EmpSeP6 = new JSeparator();

EmpSeP6.setBounds(200, 165, 175, 2);

EmpSeP6.setBorder(BorderFactory.createLineBorder(new Color(28, 37, 47), 2));

Employee.add(EmpSeP6);

EmpAnswer.addMouseListener(new MouseAdapter() {

public void mouseClicked(MouseEvent e) {

EmpSeP6.setBorder(BorderFactory.createLineBorder(new Color(28, 37, 47), 2));

} });

EmpAnswer.addKeyListener(new KeyAdapter() {

public void keyTyped(KeyEvent e) {

if (EmpAnswer.getText().equals("Answer")) {

EmpAnswer.setText("");

} } });

EmpAnswer.addFocusListener(new FocusListener() {

public void focusGained(FocusEvent e) {}

public void focusLost(FocusEvent e) {

if (EmpAnswer.getText().length() == 0) {

EmpAnswer.setText("Answer");

} } });

JLabel ECPD = new JLabel("Confirm Password"); // ECPD= Employee Confirm ECPD.setBounds(408, 135, 160, 30);

ECPD.setForeground(Color.white);

ECPD.setFont(new Font("LucidaGrande", Font.BOLD, 17));

Employee.add(ECPD);

JSeparator EmpSeP8 = new JSeparator();

EmpSeP8.setBounds(400, 130, 175, 40);

EmpSeP8.setBorder(BorderFactory.createLineBorder(new Color(28, 37, 47), 40));

Employee.add(EmpSeP8);

JPasswordField EmpConfirmPassword = new JPasswordField();

EmpConfirmPassword.setBounds(585, 135, 180, 25);

EmpConfirmPassword.setBorder(BorderFactory.createLineBorder(new Color(255, 255, 255)));

EmpConfirmPassword.setForeground(new Color(192, 192, 192));

EmpConfirmPassword.setEchoChar('\*');

EmpConfirmPassword.setFont(new Font("LucidaGrande", Font.PLAIN, 17));

Employee.add(EmpConfirmPassword);

JSeparator EmpSeP9 = new JSeparator();

EmpSeP9.setBounds(582, 161, 180, 2);

EmpSeP9.setBorder(BorderFactory.createLineBorder(new Color(28, 37, 47), 2));

Employee.add(EmpSeP9);

EmpConfirmPassword.addMouseListener(new MouseAdapter() {

public void mouseClicked(MouseEvent e) {

EmpSeP9.setBorder(BorderFactory.createLineBorder(new Color(28, 37, 47), 2));

} });

JLabel RshowIcon1 = new JLabel(new ImageIcon("C:\\Users\\AHsan Usmani\\Desktop\\

Programs\\Project GMS\\src\\view.png"));

RshowIcon1.setBounds(762, 127, 30, 30);

JLabel RhideIcon1 = new JLabel( new ImageIcon("C:\\Users\\AHsan Usmani

\\Desktop\\Programs\\Project GMS\\src\\hide.png"));

RhideIcon1.setVisible(false);

RhideIcon1.setBounds(762, 127, 30, 30);

Employee.add(RshowIcon1);

Employee.add(RhideIcon1);

RshowIcon1.addMouseListener(new MouseAdapter() {

public void mousePressed(MouseEvent e) {

RshowIcon1.setVisible(false);

RhideIcon1.setVisible(true);

EmpConfirmPassword.setEchoChar((char) 0); }

public void mouseReleased(MouseEvent e) {

RhideIcon1.setVisible(false);

RshowIcon1.setVisible(true);

EmpConfirmPassword.setEchoChar('\*'); } });

JLabel EPHN = new JLabel("Contact Number"); // EPHN= Employee Phone

EPHN.setBounds(25, 188, 140, 30);

EPHN.setForeground(Color.white);

EPHN.setFont(new Font("LucidaGrande", Font.BOLD, 17));

Employee.add(EPHN);

JSeparator EmpSeP10 = new JSeparator();

EmpSeP10.setBounds(15, 183, 175, 40);

EmpSeP10.setBorder(BorderFactory.createLineBorder(new Color(28, 37, 47), 40));

Employee.add(EmpSeP10);

JTextField EmpPhoneNo = new JTextField();

EmpPhoneNo.setBounds(200, 190, 180, 25);

EmpPhoneNo.setBorder(BorderFactory.createLineBorder(new Color(255, 255, 255)));

EmpPhoneNo.setForeground(new Color(192, 192, 192));

EmpPhoneNo.setFont(new Font("LucidaGrande", Font.PLAIN, 17));

Employee.add(EmpPhoneNo);

JSeparator EmpSeP11 = new JSeparator();

EmpSeP11.setBounds(198, 218, 175, 2);

EmpSeP11.setBorder(BorderFactory.createLineBorder(new Color(28, 37, 47), 2));

Employee.add(EmpSeP11);

EmpPhoneNo.addMouseListener(new MouseAdapter() {

public void mouseClicked(MouseEvent e) {

EmpSeP11.setBorder(BorderFactory.createLineBorder(new Color(28, 37, 47), 2));

} });

JLabel EACN = new JLabel("Aadhaar Card No."); // ECN= Employee EACN.setBounds(410, 185, 150, 30);

EACN.setForeground(Color.white);

EACN.setFont(new Font("LucidaGrande", Font.BOLD, 17));

Employee.add(EACN);

JSeparator EmpSeP12 = new JSeparator();

EmpSeP12.setBounds(400, 183, 175, 40);

EmpSeP12.setBorder(BorderFactory.createLineBorder(new Color(28, 37, 47), 40));

Employee.add(EmpSeP12);

JTextField EmpAadhaarNo = new JTextField();

EmpAadhaarNo.setBounds(585, 192, 180, 25);

EmpAadhaarNo.setBorder(BorderFactory.createLineBorder(new Color(255, 255, 255)));

EmpAadhaarNo.setForeground(new Color(192, 192, 192));

EmpAadhaarNo.setFont(new Font("LucidaGrande", Font.PLAIN, 17));

Employee.add(EmpAadhaarNo);

JSeparator EmpSeP13 = new JSeparator();

EmpSeP13.setBounds(582, 218, 180, 2);

EmpSeP13.setBorder(BorderFactory.createLineBorder(new Color(28, 37, 47), 2));

Employee.add(EmpSeP13);

EmpAadhaarNo.addMouseListener(new MouseAdapter() {

public void mouseClicked(MouseEvent e) {

EmpSeP13.setBorder(BorderFactory.createLineBorder(new Color(28, 37, 47), 2));

} });

JLabel Emptext2 = new JLabel("Employee Details");

Emptext2.setBounds(20, 230, 750, 80);

Emptext2.setForeground(Color.red);

Emptext2.setFont(new Font("Times New Roman", Font.PLAIN, 23));

Employee.add(Emptext2);

JSeparator EmpSeP14 = new JSeparator();

EmpSeP14.setBounds(15, 290, 770, 1);

EmpSeP14.setBorder(BorderFactory.createLineBorder(new Color(0, 0, 0), 1));

Employee.add(EmpSeP14);

String col[] = {"Username", "Password", "Security Question", "Answer",

"Contact No.", "Aadhaar No." };

DefaultTableModel model = new DefaultTableModel();

JTable et = new JTable(model);

model.setColumnIdentifiers(col);

JScrollPane emppane = new JScrollPane();

et.setRowHeight(28);

et.setFont(new Font("Times New Roman", Font.PLAIN, 16));

JTableHeader eth = et.getTableHeader();

eth.setFont(new Font("Times New Roman", Font.PLAIN, 20));

eth.setForeground(Color.white);

eth.setBackground(new Color(28, 37, 47));

eth.getColumnModel().getColumn(2).setPreferredWidth(40);

eth.getColumnModel().getColumn(4).setPreferredWidth(40);

eth.getColumnModel().getColumn(1).setPreferredWidth(50);

et.getTableHeader().setReorderingAllowed(false);

emppane.setViewportView(et);

emppane.setBounds(15, 300, 780, 200);

Employee.add(emppane);

try {

Connection con = DriverManager.getConnection("jdbc:ucanaccess://C:\\Users\\AHsan

Usmani\\Desktop\\Programs\\Project GMS\\src\\sms.accdb");

Statement st1 = con.createStatement();

ResultSet res = st1.executeQuery("Select \* from Employee");

while (res.next()) {

String Eusername = res.getString("username");

String EPassword = res.getString("password");

String ESQ = res.getString("security\_question");

String Eans = res.getString("answer");

String EConNo = res.getString("Contact\_number");

String EIDProf = res.getString("Aadhaar\_NO");

String Edetails[] = { Eusername, EPassword, ESQ, Eans, EConNo, EIDProf };

model.addRow(Edetails); }

} catch (Exception evt) { System.out.println(evt); }

// Employee JTable end

JButton EmpSave = new JButton("Save");

EmpSave.setBounds(180, 520, 150, 37);

EmpSave.setBackground(new Color(0, 128, 255));

EmpSave.setForeground(Color.white);

EmpSave.setFont(new Font("LucidaGrande", Font.PLAIN, 26));

Employee.add(EmpSave);

EmpSave.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

if (EmpUsername.getText().length() == 0) {

EmpSeP3.setBorder(BorderFactory.createLineBorder(Color.red, 1));}

else if (EmpPassword.getText().length() == 0) {

EmpSeP5.setBorder(BorderFactory.createLineBorder(Color.red, 1));}

else if (EmpConfirmPassword.getText().length() == 0) {

EmpSeP9.setBorder(BorderFactory.createLineBorder(Color.red, 1)); }

else if (!(EmpPassword.getText().equals(EmpConfirmPassword.getText()))) {

ImageIcon ig1 = new ImageIcon("C:\\Users\\AHsan Usmani\\Desktop\\Programs\\Project

GMS\\src\\error.png");

JOptionPane.showMessageDialog(null,"Your password and confirm password\ndidn't

match please try again!", "Warrning",JOptionPane.WARNING\_MESSAGE, ig1);

}

else if (EmpSecQuestion.getItemAt(EmpSecQuestion.getSelectedIndex()) ==

EmpSecQuestion.getItemAt(0)) {

ImageIcon ig1 = new ImageIcon("C:\\Users\\AHsan Usmani\\Desktop\\Programs\\Project

GMS\\src\\error.png");

JOptionPane.showMessageDialog(null, "Please select security question", "Warrning",

JOptionPane.WARNING\_MESSAGE, ig1); }

else if (EmpAnswer.getText().equals("Answer")) {

EmpSeP6.setBorder(BorderFactory.createLineBorder(Color.red, 1)); }

else if (EmpPhoneNo.getText().length() == 0) {

EmpSeP11.setBorder(BorderFactory.createLineBorder(Color.red, 1)); }

else if (EmpAadhaarNo.getText().length() == 0) { EmpSeP13.setBorder(BorderFactory.createLineBorder(Color.red, 1)); }

else {

try {

Connection con = DriverManager.getConnection( "jdbc:ucanaccess://C:\\Users\\AHsan Usmani\\Desktop\\Programs\\Project GMS\\src\\sms.accdb");

Statement st = con.createStatement();

int i = st.executeUpdate("insert into Employee(username,password,security\_question,

answer,Contact\_number,Aadhaar\_NO) values( '"+ EmpUsername.getText() + "', '" + EmpPassword.getText() + "', '"+ EmpSecQuestion.getSelectedItem().toString() + "','" + EmpAnswer.getText()+ "', '" + EmpPhoneNo.getText() + "', '" + EmpAadhaarNo.getText() + "')");

if (i > 0) {

ImageIcon ig1 = new ImageIcon("C:\\Users\\AHsan Usmani\\Desktop\\Programs\\Project GMS\\src\\done.png");

JOptionPane.showMessageDialog(null, "Employee account successfully created.", "Done",

JOptionPane.INFORMATION\_MESSAGE, ig1);

model.addRow(new Object[] {EmpUsername.getText(), EmpPassword.getText(), EmpSecQuestion.getSelectedItem().toString(), EmpAnswer.getText(),EmpPhoneNo.getText(), EmpAadhaarNo.getText() });

EmpUsername.setText(""); EmpPassword.setText("");

EmpConfirmPassword.setText("");

EmpSecQuestion.setSelectedIndex(0);

EmpAnswer.setText("");

EmpPhoneNo.setText("");

EmpAadhaarNo.setText(""); }

} catch (Exception evt) { System.out.println(evt); }

} } });

JButton EmpUpdate = new JButton("Update");

EmpUpdate.setBounds(360, 520, 150, 37);

EmpUpdate.setBackground(new Color(0, 128, 255));

EmpUpdate.setForeground(Color.white);

EmpUpdate.setFont(new Font("LucidaGrande", Font.PLAIN, 26));

Employee.add(EmpUpdate);

EmpUpdate.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

try {

Connection con = DriverManager.getConnection("jdbc:ucanaccess://C:\\Users\\AHsan

Usmani\\Desktop\\Programs\\Project GMS\\src\\sms.accdb");

Statement st1 = con.createStatement();

int i=st1.executeUpdate("update Employee set username='"+EmpUsername.getText()+"',

password='"+EmpPassword.getText()+"',answer='"+EmpAnswer.getText()+"',Contact\_numb

er='"+EmpPhoneNo.getText()+"', Aadhaar\_NO='"+EmpAadhaarNo.getText()+"' where Aadhaar\_NO ='"+EmpAadhaarNo.getText()+"' ");

if (i>0) {

ImageIcon ig1 = new ImageIcon("C:\\Users\\AHsan Usmani\\Desktop\\Programs\\Project

GMS\\src\\done.png");

JOptionPane.showMessageDialog(null, "Employee details update successfully.",

"Done",JOptionPane.INFORMATION\_MESSAGE, ig1);

int j=et.getSelectedRow();

if(j>=0) {

model.setValueAt(EmpUsername.getText(), j, 0);

model.setValueAt(EmpPassword.getText(),j,1); model.setValueAt(EmpSecQuestion.getSelectedItem().toString(), j, 2);

model.setValueAt(EmpAnswer.getText(), j, 3);

model.setValueAt(EmpPhoneNo.getText(), j, 4);

model.setValueAt(EmpAadhaarNo.getText(), j, 5); }

EmpUsername.setText("");

EmpPassword.setText("");

EmpConfirmPassword.setText("");

EmpSecQuestion.setSelectedIndex(0);

EmpAnswer.setText("");

EmpPhoneNo.setText("");

EmpAadhaarNo.setText("");

EmpSave.setEnabled(true); }

}catch(Exception evt) { System.out.println(evt); }

} });

JButton EmpDelete = new JButton("Delete");

EmpDelete.setBounds(540, 520, 150, 37);

EmpDelete.setBackground(new Color(0, 128, 255));

EmpDelete.setForeground(Color.white);

EmpDelete.setFont(new Font("LucidaGrande", Font.PLAIN, 26));

Employee.add(EmpDelete);

EmpDelete.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent evt) {

if(et.getSelectedRowCount()==1) {

try {

Connection con=DriverManager.getConnection("jdbc:ucanaccess://C:\\Users\\AHsan

Usmani\\Desktop\\Programs\\Project GMS\\src\\sms.accdb");

Statement st=con.createStatement();

int i=st.executeUpdate("Delete from Employee where username='"+EmpUsername.getText() +"'");

if(i>0) {

model.removeRow(et.getSelectedRow());

JOptionPane.showMessageDialog(null,"Row Deleted", "Warrning",JOptionPane.

INFORMATION\_MESSAGE); }

} catch(Exception e) {System.out.println(e);}

}

else {

if(et.getRowCount()==0) JOptionPane.showMessageDialog(null,"Empty row", "Warrning",JOptionPane .WARNING\_MESSAGE);

else{

JOptionPane.showMessageDialog(null,"Please select the row", "Warrning"

,JOptionPane.WARNING\_MESSAGE); }

} } });

et.addMouseListener(new MouseAdapter() {

public void mouseClicked(MouseEvent e) {

String EmUsername = model.getValueAt(et.getSelectedRow(), 0).toString();

String e\_password = model.getValueAt(et.getSelectedRow(), 1).toString();

String EsecQue = model.getValueAt(et.getSelectedRow(), 2).toString();

String EAns = model.getValueAt(et.getSelectedRow(), 3).toString();

String ECon = model.getValueAt(et.getSelectedRow(), 4).toString();

String E\_Id = model.getValueAt(et.getSelectedRow(), 5).toString();

EmpUsername.setText(EmUsername);

EmpPassword.setText(e\_password);

EmpConfirmPassword.setText("");

EmpAnswer.setText(EAns);

EmpPhoneNo.setText(ECon);

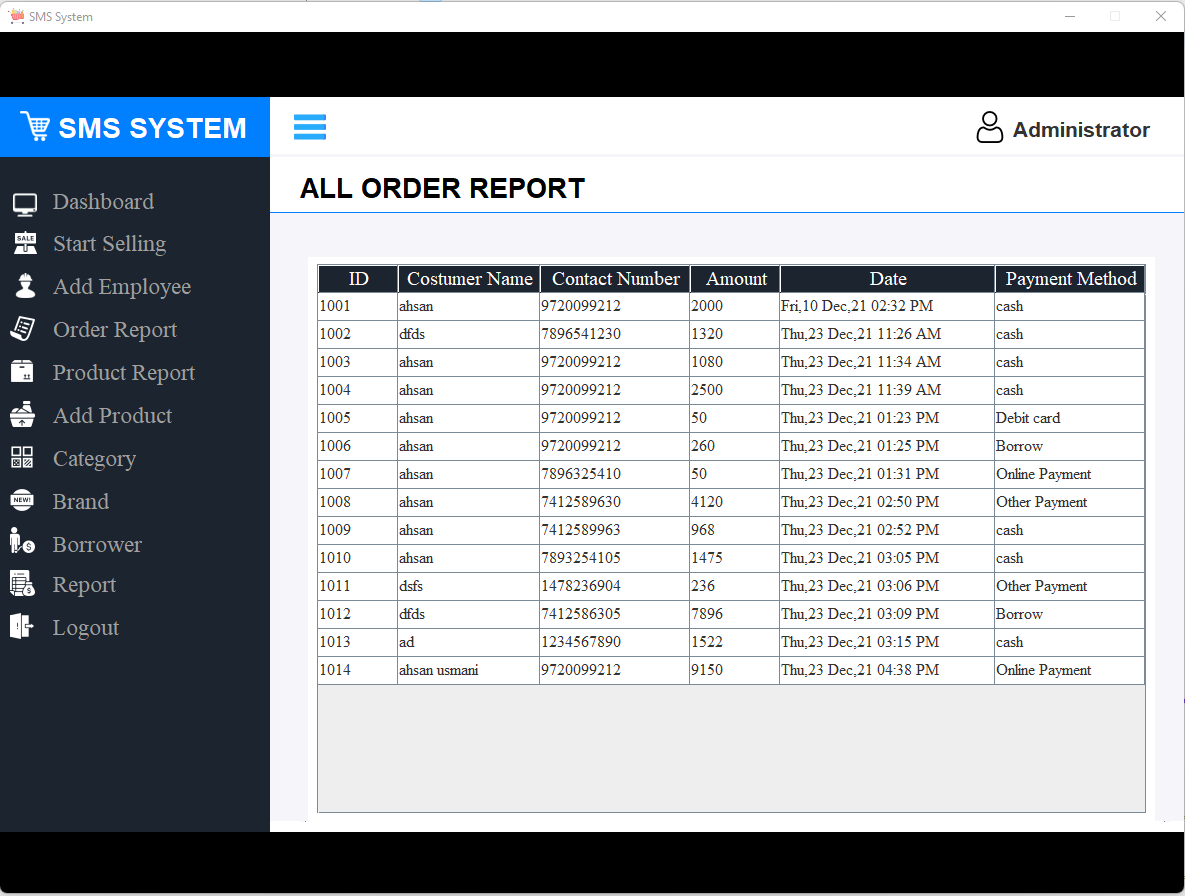
EmpAadhaarNo.setText(E\_Id);

EmpDelete.setEnabled(true);

EmpSave.setEnabled(false);

} });

ORDER REPORT WINDOW SCREEN



**Output form information**

Order Report panel show all the selling products recored of supermarket management system.

**Coding of Order Report Panel**

JPanel OrderReport = new JPanel();

OrderReport.setBackground(Color.white);

OrderReport.setLayout(null);

TP.add(OrderReport);

String Rcol[] = {"ID", "Costumer Name","Contact Number", "Amount", "Date",

"Payment Method" };

DefaultTableModel Rmodel = new DefaultTableModel();

JTable Rt = new JTable(Rmodel); // Ot=order table Rmodel.setColumnIdentifiers(Rcol);

Rt.setRowHeight(28);

Rt.setFont(new Font("Times New Roman", Font.PLAIN, 16));

JTableHeader rth = Rt.getTableHeader();

rth.setFont(new Font("Times New Roman", Font.PLAIN, 20));

rth.setForeground(Color.white);

rth.setBackground(new Color(28, 37, 47));

rth.getColumnModel().getColumn(0).setMaxWidth(80);

rth.getColumnModel().getColumn(0).setMaxWidth(80);

rth.getColumnModel().getColumn(1).setMaxWidth(150);

rth.getColumnModel().getColumn(1).setMaxWidth(150);

rth.getColumnModel().getColumn(2).setMaxWidth(150);

rth.getColumnModel().getColumn(2).setMinWidth(150);

rth.getColumnModel().getColumn(3).setMaxWidth(90);

rth.getColumnModel().getColumn(3).setMinWidth(90);

rth.getColumnModel().getColumn(4).setMaxWidth(215);

rth.getColumnModel().getColumn(4).setMinWidth(215);

rth.getColumnModel().getColumn(5).setMaxWidth(150);

rth.getColumnModel().getColumn(5).setMinWidth(150);

Rt.getTableHeader().setReorderingAllowed(false);

JScrollPane Rpane = new JScrollPane();

Rpane.setViewportView(Rt);

Rpane.setBounds(10, 5, 830, 550);

OrderReport.add(Rpane);

try {

Connection con=DriverManager.getConnection("jdbc:ucanaccess://C:\\Users\\AHsan

Usmani\\Desktop\\Programs\\Project GMS\\src\\sms.accdb");

Statement st1=con.createStatement();

ResultSet res=st1.executeQuery("select \* from Costumers");

while(res.next()) {

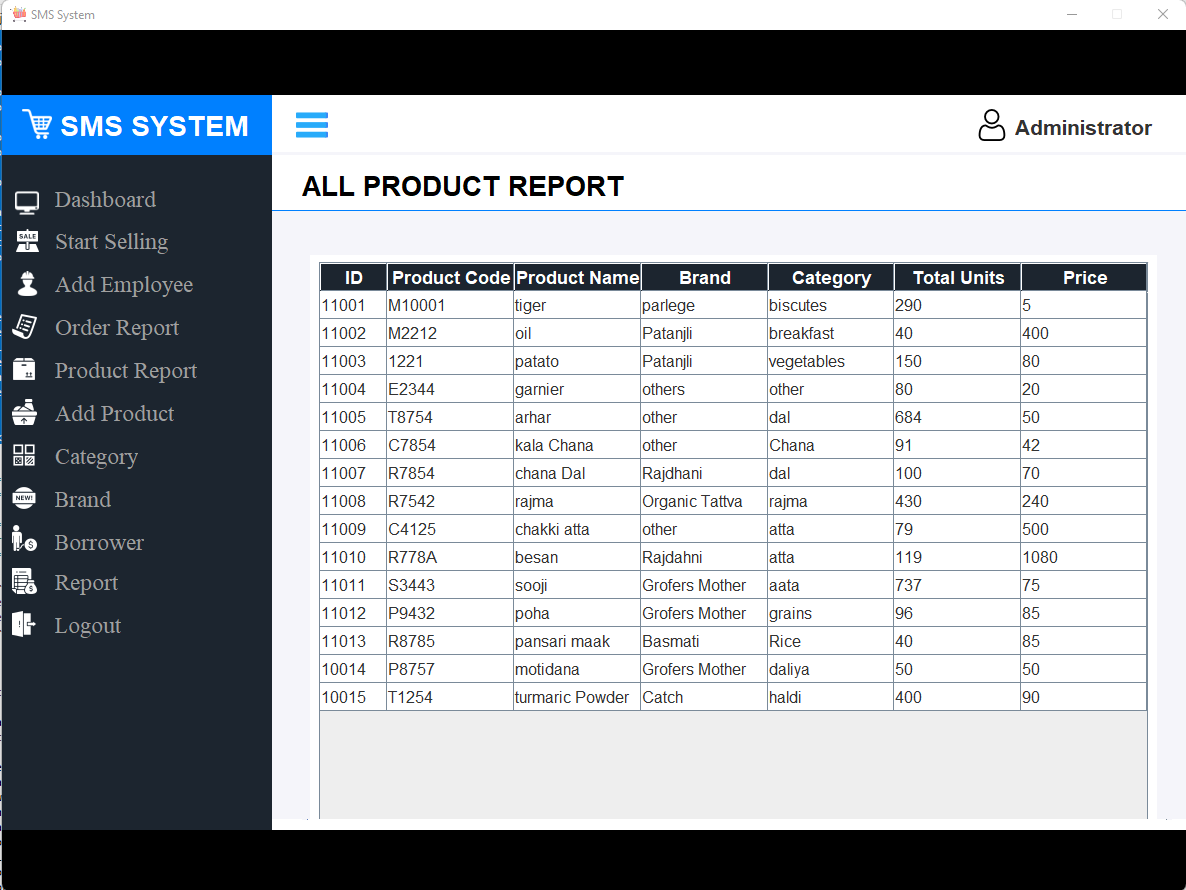
String Rdetails[] = {res.getString("Order\_Id"), res.getString("Cos\_Name"), res.getString("Cos\_number"), res.getString("Amount"), res.getString("Ord\_Date")

,res.getString ("Payment\_Status") };

Rmodel.addRow(Rdetails); }

}catch(Exception e) {System.out.println(e);}

PRODUCT REPORT WINDOW SCREEN



**Output form information**

Product Report panel show all the purchesed goods recored of supermarket management system.

**Coding of Product Report Panel**

JPanel ProductReport = new JPanel();

ProductReport.setBackground(Color.white);

ProductReport.setLayout(null);

TP.add(ProductReport);

String Pcol[] = { "ID", "Product Code", "Product Name", "Brand", "Category",

"Total Units", "Price" };

DefaultTableModel Pmodel = new DefaultTableModel();

JTable Pt = new JTable(Pmodel);

Pmodel.setColumnIdentifiers(Pcol);

JScrollPane Propane = new JScrollPane();

Pt.setRowHeight(28);

Pt.setFont(new Font("LucidaGrande", Font.PLAIN, 16));

JTableHeader Pth = Pt.getTableHeader();

Pth.getColumnModel().getColumn(0).setPreferredWidth(10);

Pt.getColumnModel().getColumn(0).setPreferredWidth(10);

Pth.setFont(new Font("LucidaGrande", Font.BOLD, 18));

Pth.setForeground(Color.white);

Pth.setBackground(new Color(28, 37, 47));

Pt.getTableHeader().setReorderingAllowed(false);

Propane.setViewportView(Pt);

Propane.setBounds(10, 5, 830, 600);

ProductReport.add(Propane);

try {

Connection con = DriverManager.getConnection("jdbc:ucanaccess://C:\\Users\\AHsan

Usmani\\Desktop\\Programs\\Project GMS\\src\\sms.accdb");

Statement st1 = con.createStatement();

ResultSet res = st1.executeQuery("Select \* from Product");

while (res.next()) {

String ProID = res.getString("Pro\_ID");

String Pro\_Code = res.getString("Product\_Code");

String Pro\_Name = res.getString("Product\_Name");

String Pro\_Brand = res.getString("Product\_Brand");

String Pro\_category = res.getString("Product\_Category");

String Units = res.getString("Total\_units");

String Price = res.getString("Price\_Per\_Unit");

String Pdetails[] = { ProID, Pro\_Code, Pro\_Name, Pro\_Brand,

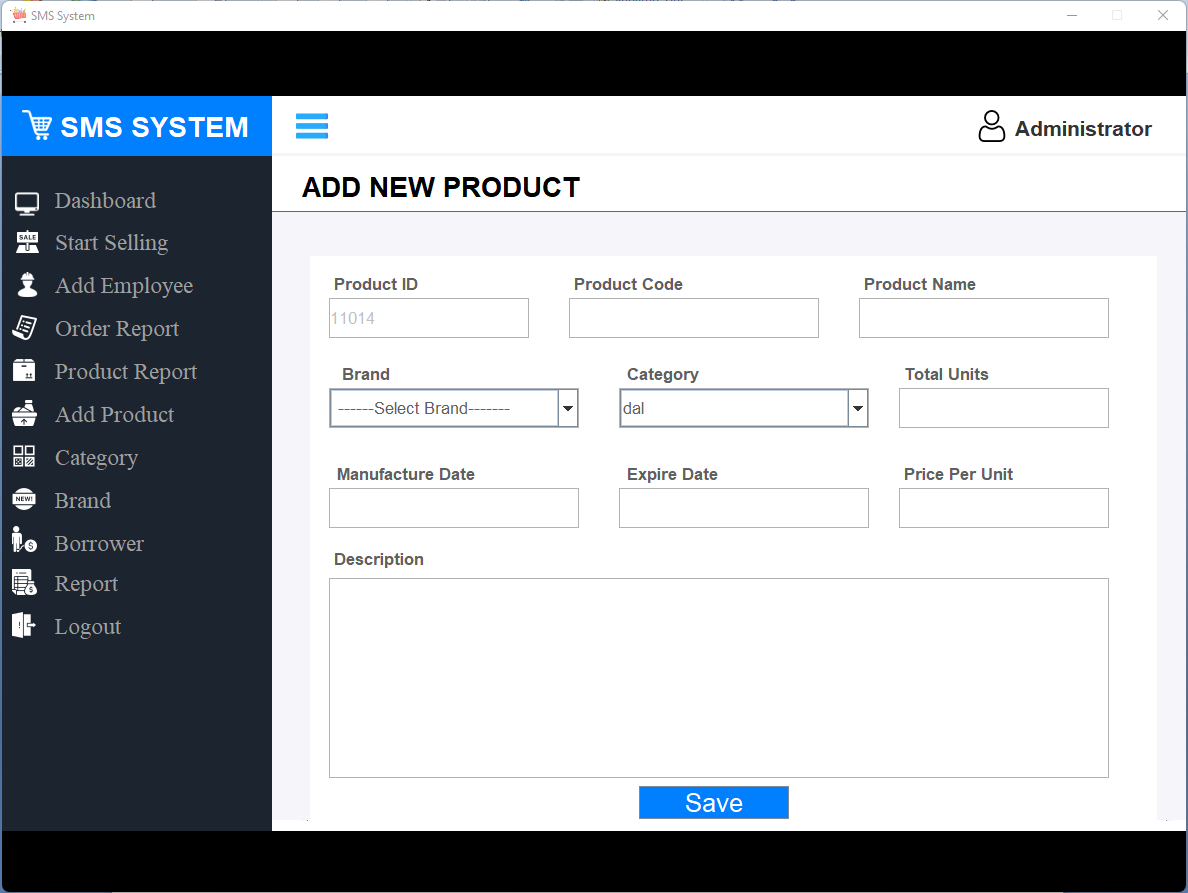
Pro\_category, Units, Price };

Pmodel.addRow(Pdetails);

}

} catch (Exception evt) { System.out.println(evt); }

ADD PRODUCT WINDOW SCREEN



**Output form information**

Add product panel is use for adding the purchased product by super market in the super market management system software.

**Coding of Add Product Panel**

JPanel AddProduct = new JPanel();

AddProduct.setBackground(Color.white);

AddProduct.setLayout(null);

TP.add(AddProduct);

JLabel PAL = new JLabel("Product ID"); // PAL=Product Add Label

PAL.setBounds(25, 10, 122, 30);

PAL.setForeground(new Color(96, 96, 96));

PAL.setFont(new Font("LucidaGrande", Font.BOLD, 17));

AddProduct.add(PAL);

JTextField ProID = new JTextField();

ProID.setBounds(20, 40, 200, 40);

ProID.setBorder(BorderFactory.createLineBorder(new Color(179, 179, 179)));

ProID.setForeground(new Color(192, 192, 192));

ProID.setBackground(Color.white);

ProID.setEditable(false);

ProID.setFont(new Font("LucidaGrande", Font.PLAIN, 17));

AddProduct.add(ProID);

try {

Connection con = DriverManager.getConnection("jdbc:ucanaccess://C:\\Users\\AHsan

Usmani\\Desktop\\Programs\\Project GMS\\src\\sms.accdb");

Statement st1 = con.createStatement();

ResultSet res = st1.executeQuery("Select max(Pro\_ID) from Product");

if (res.next()) {

int id = res.getInt(1);

++id;

ProID.setText(String.valueOf(id)); }

} catch (Exception evt) { System.out.println(evt); }

JLabel PC = new JLabel("Product Code"); // PC=Product code

PC.setBounds(265, 10, 122, 30);

PC.setForeground(new Color(96, 96, 96));

PC.setFont(new Font("LucidaGrande", Font.BOLD, 17));

AddProduct.add(PC);

JTextField ProductCode = new JTextField();

ProductCode.setBounds(260, 40, 250, 40);

ProductCode.setBorder(BorderFactory.createLineBorder(new Color(179, 179, 179)));

ProductCode.setForeground(new Color(192, 192, 192));

ProductCode.setFont(new Font("LucidaGrande", Font.PLAIN, 17));

AddProduct.add(ProductCode);

error1 = new JLabel(" \u274C Enter the Produt Code”);

error1.setForeground(Color.red);

error1.setFont(new Font("LucidaGrande", Font.PLAIN, 11));

error1.setBounds(263, 83, 300, 15);

error1.setVisible(false);

AddProduct.add(error1);

ProductCode.addMouseListener(new MouseAdapter() {

public void mouseClicked(MouseEvent e) {

ProductCode.setBorder(BorderFactory.createLineBorder(new Color(179, 179, 179)));

error1.setVisible(false);

} });

JLabel PN = new JLabel("Product Name"); // PC=Product code

PN.setBounds(555, 10, 122, 30);

PN.setForeground(new Color(96, 96, 96));

PN.setFont(new Font("LucidaGrande", Font.BOLD, 17));

AddProduct.add(PN);

JTextField ProductName = new JTextField();

ProductName.setBounds(550, 40, 250, 40);

ProductName.setBorder(BorderFactory.createLineBorder(new Color(179, 179, 179)));

ProductName.setForeground(new Color(192, 192, 192));

ProductName.setFont(new Font("LucidaGrande", Font.PLAIN, 17));

AddProduct.add(ProductName);

error2 = new JLabel(" \u274C Enter the product name");

error2.setForeground(Color.red);

error2.setFont(new Font("LucidaGrande", Font.PLAIN, 11));

error2.setBounds(553, 83, 300, 15);

error2.setVisible(false);

AddProduct.add(error2);

ProductName.addMouseListener(new MouseAdapter() {

public void mouseClicked(MouseEvent e) {

ProductName.setBorder(BorderFactory.createLineBorder(new Color(179, 179, 179)));

error2.setVisible(false);

} });

JLabel BL = new JLabel("Brand"); // BL=Brand Label

BL.setBounds(33, 100, 122, 30);

BL.setForeground(new Color(96, 96, 96));

BL.setFont(new Font("LucidaGrande", Font.BOLD, 17));

AddProduct.add(BL);

JComboBox Brand = new JComboBox();

Brand.setBounds(20, 130, 250, 40);

Brand.setBorder(BorderFactory.createLineBorder(new Color(179, 179, 179)));

Brand.setForeground(new Color(96, 96, 96));

Brand.setBackground(Color.white);

Brand.setFont(new Font("LucidaGrande", Font.PLAIN, 17));

AddProduct.add(Brand);

try {

Connection con = DriverManager.getConnection("jdbc:ucanaccess://C:\\Users\\AHsan

Usmani\\Desktop\\Programs\\Project GMS\\src\\sms.accdb");

Statement st1 = con.createStatement();

ResultSet res = st1.executeQuery("select Brand\_Title from Brand");

while (res.next()) {

Brand.addItem(res.getString(1)); }

} catch (Exception evt) { System.out.println(evt); }

error3 = new JLabel(" \u274C Select the Brand");

error3.setForeground(Color.red);

error3.setFont(new Font("LucidaGrande", Font.PLAIN, 11));

error3.setBounds(23, 173, 300, 15);

error3.setVisible(false);

AddProduct.add(error3);

Brand.addMouseListener(new MouseAdapter() {

public void mouseClicked(MouseEvent e) {

Brand.setBorder(BorderFactory.createLineBorder(new Color(179, 179, 179)));

error3.setVisible(false);

} });

JLabel CL = new JLabel("Category"); // CL=Category Label

CL.setBounds(318, 100, 122, 30);

CL.setForeground(new Color(96, 96, 96));

CL.setFont(new Font("LucidaGrande", Font.BOLD, 17));

AddProduct.add(CL);

// String category;

JComboBox Category = new JComboBox();

Category.setBounds(310, 130, 250, 40);

Category.setBorder(BorderFactory.createLineBorder(new Color(179, 179, 179)));

Category.setForeground(new Color(96, 96, 96));

Category.setBackground(Color.white);

Category.setFont(new Font("LucidaGrande", Font.PLAIN, 17));

AddProduct.add(Category);

try {

Connection con = DriverManager.getConnection("jdbc:ucanaccess://C:\\Users\\AHsan

Usmani\\Desktop\\Programs\\Project GMS\\src\\sms.accdb");

Statement st1 = con.createStatement();

ResultSet res = st1.executeQuery("select Cat\_Title from Category");

while (res.next()) {

Category.addItem(res.getString(1)); }

} catch (Exception evt) { System.out.println(evt); }

error4 = new JLabel(" \u274C Select the Category");

error4.setForeground(Color.red);

error4.setFont(new Font("LucidaGrande", Font.PLAIN, 11));

error4.setBounds(313, 173, 300, 15);

error4.setVisible(false);

AddProduct.add(error4);

Category.addMouseListener(new MouseAdapter() {

public void mouseClicked(MouseEvent e) {

Category.setBorder(BorderFactory.createLineBorder(new Color(179, 179, 179)));

error4.setVisible(false);

} });

JLabel PTU = new JLabel("Total Units"); // PTU=Product total units

PTU.setBounds(595, 100, 122, 30);

PTU.setForeground(new Color(96, 96, 96));

PTU.setFont(new Font("LucidaGrande", Font.BOLD, 17));

AddProduct.add(PTU);

JTextField ProductUnits = new JTextField();

ProductUnits.setBounds(590, 130, 210, 40);

ProductUnits.setBorder(BorderFactory.createLineBorder(new Color(179, 179, 179)));

ProductUnits.setForeground(new Color(192, 192, 192));

ProductUnits.setFont(new Font("LucidaGrande", Font.PLAIN, 17));

AddProduct.add(ProductUnits);

error5 = new JLabel(" \u274C Enter the Units"); error5.setForeground(Color.red);

error5.setFont(new Font("LucidaGrande", Font.PLAIN, 11));

error5.setBounds(593, 173, 300, 15);

error5.setVisible(false);

AddProduct.add(error5);

ProductUnits.addMouseListener(new MouseAdapter() {

public void mouseClicked(MouseEvent e) {

ProductUnits.setBorder(BorderFactory.createLineBorder(new Color(179, 179, 179)));

error5.setVisible(false);

} });

JLabel PMnu = new JLabel("Manufacture Date"); PMnu.setBounds(28, 200, 170, 30);

PMnu.setForeground(new Color(96, 96, 96));

PMnu.setFont(new Font("LucidaGrande", Font.BOLD, 17));

AddProduct.add(PMnu);

JTextField Manufacture = new JTextField();

Manufacture.setBounds(20, 230, 250, 40);

Manufacture.setBorder(BorderFactory.createLineBorder(new Color(179, 179, 179)));

Manufacture.setForeground(new Color(192, 192, 192));

Manufacture.setFont(new Font("LucidaGrande", Font.PLAIN, 17));

AddProduct.add(Manufacture);

JLabel PExp = new JLabel("Expire Date"); // PExp=Product Expire

PExp.setBounds(318, 200, 170, 30);

PExp.setForeground(new Color(96, 96, 96));

PExp.setFont(new Font("LucidaGrande", Font.BOLD, 17));

AddProduct.add(PExp);

JTextField Expire = new JTextField();

Expire.setBounds(310, 230, 250, 40);

Expire.setBorder(BorderFactory.createLineBorder(new Color(179, 179, 179)));

Expire.setForeground(new Color(192, 192, 192));

Expire.setFont(new Font("LucidaGrande", Font.PLAIN, 17));

AddProduct.add(Expire);

JLabel PP = new JLabel("Price Per Unit"); // PP=Product Price

PP.setBounds(595, 200, 122, 30);

PP.setForeground(new Color(96, 96, 96));

PP.setFont(new Font("LucidaGrande", Font.BOLD, 17));

AddProduct.add(PP);

JTextField Price = new JTextField();

Price.setBounds(590, 230, 210, 40);

Price.setBorder(BorderFactory.createLineBorder(new Color(179, 179, 179)));

Price.setForeground(new Color(192, 192, 192));

Price.setFont(new Font("LucidaGrande", Font.PLAIN, 17));

AddProduct.add(Price);

error9 = new JLabel(" \u274C Enter the Price");

error9.setForeground(Color.red);

error9.setFont(new Font("LucidaGrande", Font.PLAIN, 11));

error9.setBounds(593, 273, 300, 15);

error9.setVisible(false);

AddProduct.add(error9);

Price.addMouseListener(new MouseAdapter() {

public void mouseClicked(MouseEvent e) {

Price.setBorder(BorderFactory.createLineBorder(new Color(179, 179, 179)));

error9.setVisible(false);

} });

JLabel PD = new JLabel("Description"); // PD=Product Description

PD.setBounds(25, 285, 122, 30);

PD.setForeground(new Color(96, 96, 96));

PD.setFont(new Font("LucidaGrande", Font.BOLD, 17));

AddProduct.add(PD);

JTextArea ProductDescription = new JTextArea();

ProductDescription.setFont(new Font("LucidaGrande", Font.PLAIN, 22));

ProductDescription.setBounds(20, 320, 780, 200);

ProductDescription.setForeground(new Color(192, 192, 192));

ProductDescription.setBorder(BorderFactory.createLineBorder(new Color(179, 179, 179)));

AddProduct.add(ProductDescription);

JButton ProSave = new JButton("Save");

ProSave.setBounds(330, 528, 150, 33);

ProSave.setBackground(new Color(0, 128, 255));

ProSave.setForeground(Color.white);

ProSave.setFont(new Font("LucidaGrande", Font.PLAIN, 26));

AddProduct.add(ProSave);

ProSave.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

if (ProductCode.getText().length() == 0) { ProductCode.setBorder(BorderFactory.createLineBorder(Color.red));

error1.setVisible(true);

}

if(Manufacture.getText().length()==0||Expire.getText().length()==0)

{Manufacture.setText("N/A"); Expire.setText("N/A"); }

else if (ProductName.getText().length() == 0) {

ProductName.setBorder(BorderFactory.createLineBorder(Color.red));

error2.setVisible(true); }

else if (Brand.getItemAt(Brand.getSelectedIndex()) == Brand.getItemAt(0)) {

Brand.setBorder(BorderFactory.createLineBorder(Color.red));

error3.setVisible(true); }

else if (Category.getItemAt(Category.getSelectedIndex()) == Category.getItemAt(0)) { Category.setBorder(BorderFactory.createLineBorder(Color.red));

error4.setVisible(true); }

else if (ProductUnits.getText().length() == 0) {

ProductUnits.setBorder(BorderFactory.createLineBorder(Color.red));

error5.setVisible(true); }

else if (Price.getText().length() == 0) {

Price.setBorder(BorderFactory.createLineBorder(Color.red));

error9.setVisible(true); }

else {

try {

Connection con = DriverManager.getConnection("jdbc:ucanaccess://C:\\Users\\AHsan

Usmani\\Desktop\\Programs\\Project GMS\\src\\sms.accdb");

Statement st1 = con.createStatement();

int i = st1.executeUpdate("insert into Product(Pro\_ID, Product\_Code, Product\_Name, Product\_Brand, Product\_Category, Total\_Units, Manufacture\_Date, Expire\_Date, Price\_Per\_Unit) values('"+ ProID.getText() + "', '" + ProductCode.getText() + "', '" + ProductName.getText() + "', '" + Brand.getSelectedItem().toString() + "', '" + Category.getSelectedItem().toString() + "', '" + ProductUnits.getText()+ "', '" + Manufacture.getText() + "', '" + Expire.getText() + "', '"+ Price.getText() + "') ");

if (i > 0) {

ImageIcon ig1 = new ImageIcon("C:\\Users\\AHsan Usmani\\Desktop\\Programs\\Project

GMS\\src\\done.png");

JOptionPane.showMessageDialog(null, "Product add successfully.", "Done",

JOptionPane.INFORMATION\_MESSAGE, ig1);

ResultSet res = st1.executeQuery("Select max(Pro\_ID) from Product");

if (res.next()) {

int id = res.getInt(1);

++id;

ProID.setText(String.valueOf(id)); }

ProductCode.setText("");

ProductName.setText("");

ProductUnits.setText("");

Price.setText("");

Manufacture.setText("");

Expire.setText("");

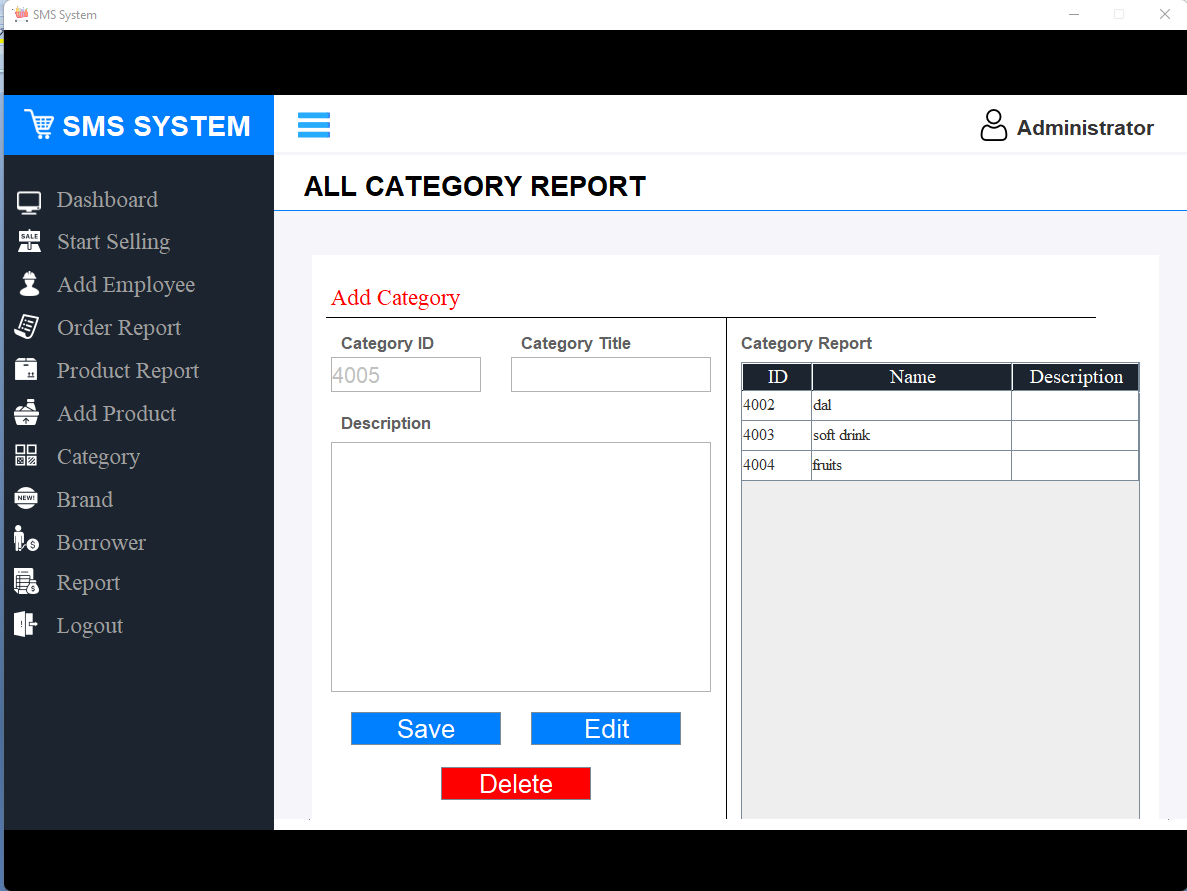
Brand.setSelectedIndex(0);

Category.setSelectedIndex(0); }

} catch (Exception evt) { System.out.print(evt); }

} } });

CATEGORY WINDOW SCREEN



**Output form information**

Category panel is used to categorized the products in super market management system in this panel we can add, remove and modify categories .

**Coding of Category Panel**

JPanel CATEGORYPanel = new JPanel();

CATEGORYPanel.setBackground(Color.white);

CATEGORYPanel.setLayout(null);

TP.add(CATEGORYPanel);

JLabel CateText1 = new JLabel("Add Category");

CateText1.setBounds(20, 0, 750, 80);

CateText1.setForeground(Color.red);

CateText1.setFont(new Font("Times New Roman", Font.PLAIN, 23));

CATEGORYPanel.add(CateText1);

JSeparator CateSeP1 = new JSeparator();

CateSeP1.setBounds(15, 60, 770, 1);

CateSeP1.setBorder(BorderFactory.createLineBorder(new Color(0, 0, 0), 1));

CATEGORYPanel.add(CateSeP1);

JLabel CateCode = new JLabel("Category ID"); // ProCode=Product Code

CateCode.setBounds(30, 70, 150, 30);

CateCode.setForeground(new Color(96, 96, 96));

CateCode.setFont(new Font("LucidaGrande", Font.BOLD, 17));

CATEGORYPanel.add(CateCode);

JTextField CategoryID = new JTextField();

CategoryID.setFont(new Font("LucidaGrande", Font.PLAIN, 22));

CategoryID.setBounds(20, 100, 150, 35);

CategoryID.setEditable(false);

CategoryID.setBackground(new Color(255, 255, 255));

CategoryID.setForeground(new Color(192, 192, 192));

CategoryID.setBorder(BorderFactory.createLineBorder(new Color(179, 179, 179)));

CATEGORYPanel.add(CategoryID);

try {

Connection con = DriverManager.getConnection("jdbc:ucanaccess://C:\\Users\\AHsan

Usmani\\Desktop\\Programs\\Project GMS\\src\\sms.accdb");

Statement st1 = con.createStatement();

ResultSet res1 = st1.executeQuery("select max(Cat\_ID) from Category");

if (res1.next()) {

int catID = res1.getInt(1);

++catID;

CategoryID.setText(String.valueOf(catID)); }

} catch (Exception evt) { System.out.println(evt); }

JLabel CateT = new JLabel("Category Title"); // CateT= Category title

CateT.setBounds(210, 70, 150, 30);

CateT.setForeground(new Color(96, 96, 96));

CateT.setFont(new Font("LucidaGrande", Font.BOLD, 17));

CATEGORYPanel.add(CateT);

JTextField CategoryTitle = new JTextField();

CategoryTitle.setFont(new Font("LucidaGrande", Font.PLAIN, 22));

CategoryTitle.setBounds(200, 100, 200, 35);

CategoryTitle.setForeground(new Color(192, 192, 192));

CategoryTitle.setBorder(BorderFactory.createLineBorder(new Color(179, 179, 179)));

CATEGORYPanel.add(CategoryTitle);

JLabel Cerror1 = new JLabel(" \u274C Enter the Brand Name");

Cerror1.setForeground(Color.red);

Cerror1.setFont(new Font("LucidaGrande", Font.PLAIN, 11));

Cerror1.setBounds(203, 138, 300, 15);

Cerror1.setVisible(false);

CATEGORYPanel.add(Cerror1);

CategoryTitle.addMouseListener(new MouseAdapter() {

public void mouseClicked(MouseEvent e) {

CategoryTitle.setBorder(BorderFactory.createLineBorder(new Color(179, 179, 179)));

Cerror1.setVisible(false);

} });

JLabel CateDes = new JLabel("Description"); CateDes.setBounds(30, 150, 150, 30);

CateDes.setForeground(new Color(96, 96, 96));

CateDes.setFont(new Font("LucidaGrande", Font.BOLD, 17));

CATEGORYPanel.add(CateDes);

JTextArea CategoryDescription = new JTextArea();

CategoryDescription.setFont(new Font("LucidaGrande", Font.PLAIN, 22));

CategoryDescription.setBounds(20, 185, 380, 250);

CategoryDescription.setForeground(new Color(192, 192, 192));

CategoryDescription.setBorder(BorderFactory.createLineBorder(new Color(179, 179, 179)));

CATEGORYPanel.add(CategoryDescription);

JSeparator CateSeP2 = new JSeparator(JSeparator.VERTICAL);

CateSeP2.setBounds(415, 60, 1, 560);

CateSeP2.setBorder(BorderFactory.createLineBorder(new Color(0, 0, 0), 1));

CATEGORYPanel.add(CateSeP2);

JLabel CateRep = new JLabel("Category Report");

CateRep.setBounds(430, 70, 150, 30);

CateRep.setForeground(new Color(96, 96, 96));

CateRep.setFont(new Font("LucidaGrande", Font.BOLD, 17));

CATEGORYPanel.add(CateRep);

DefaultTableModel Catmodel = new DefaultTableModel();

String Catcol[] = {"ID","Name","Description" };

JTable ct = new JTable(Catmodel); // ct = Category Table

ct.setRowHeight(30);

Catmodel.setColumnIdentifiers(Catcol);

JTableHeader cth = ct.getTableHeader(); // cth = category table header

cth.setFont(new Font("Times New Roman", Font.PLAIN, 20));

cth.setForeground(Color.white);

cth.setBackground(new Color(28, 37, 47));

cth.getColumnModel().getColumn(0).setMaxWidth(70);

cth.getColumnModel().getColumn(1).setMaxWidth(200);

cth.getColumnModel().getColumn(1).setMinWidth(200);

ct.setFont(new Font("Times New Roman", Font.PLAIN, 16));

ct.getTableHeader().setReorderingAllowed(false);

JScrollPane Catpane = new JScrollPane();

Catpane.setViewportView(ct);

Catpane.setBounds(430,105,400,500);

CATEGORYPanel.add(Catpane);

try {

Connection con= DriverManager.getConnection("jdbc:ucanaccess://C:\\Users\\AHsan

Usmani\\Desktop\\Programs\\Project GMS\\src\\sms.accdb");

Statement st1=con.createStatement();

ResultSet res= st1.executeQuery("Select \* from Category"); {

while(res.next()) {

String CatID=res.getString("Cat\_ID");

String CatName=res.getString("Cat\_Title");

String CatDes=res.getString("Cat\_Description");

String CatDetails[]= {CatID, CatName,CatDes};

Catmodel.addRow(CatDetails);

} }

}catch(Exception e) {System.out.println(e);}

// Ending of Category table

JButton CateSave = new JButton("Save");

CateSave.setBounds(40, 455, 150, 33);

CateSave.setBackground(new Color(0, 128, 255));

CateSave.setForeground(Color.white);

CateSave.setFont(new Font("LucidaGrande", Font.PLAIN, 26));

CATEGORYPanel.add(CateSave);

CateSave.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

if (CategoryTitle.getText().length() == 0) {

CategoryTitle.setBorder(BorderFactory.createLineBorder(Color.red));

Cerror1.setVisible(true); }

else {

try {

Connection con = DriverManager.getConnection("jdbc:ucanaccess://C:\\Users\\AHsan

Usmani\\Desktop\\Programs\\Project GMS\\src\\sms.accdb");

Statement st = con.createStatement();

int i = st.executeUpdate("insert into Category(Cat\_ID,Cat\_Title) values('"

+ CategoryID.getText() + "','" + CategoryTitle.getText() + "')");

if (i > 0) {

ImageIcon ig1 = new ImageIcon("C:\\Users\\AHsan Usmani\\Desktop\\Programs\\Project

GMS\\src\\done.png");

JOptionPane.showMessageDialog(null, "Category add successfully.", "Done",JOptionPane.

INFORMATION\_MESSAGE, ig1);

Catmodel.addRow(new Object[] {CategoryID.getText(),CategoryTitle.getText()});

ResultSet res1 = st.executeQuery("select max(Cat\_ID) from Category");

if (res1.next()) {

int catID = res1.getInt(1);

++catID; CategoryID.setText(String.valueOf(catID));

CategoryTitle.setText(""); } }

} catch (Exception evt) { System.out.println(evt); }

} } });

JButton CateEdit = new JButton("Edit");

CateEdit.setBounds(220, 455, 150, 33);

CateEdit.setBackground(new Color(0, 128, 255));

CateEdit.setForeground(Color.white);

CateEdit.setFont(new Font("LucidaGrande", Font.PLAIN, 26));

CATEGORYPanel.add(CateEdit);

CateEdit.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

try {

Connection con=DriverManager.getConnection("jdbc:ucanaccess://C:\\Users\\AHsan

Usmani\\Desktop\\Programs\\Project GMS\\src\\sms.accdb");

Statement st=con.createStatement();

int i=st.executeUpdate("update Category set Cat\_Title='"+CategoryTitle.getText()

+"' where Cat\_ID='"+CategoryID.getText()+"' ");

if(i>0) {

int j=ct.getSelectedRow();

if(j>=0) {

Catmodel.setValueAt(CategoryTitle.getText(), j, 1);

JOptionPane.showMessageDialog(null, "Category add successfully.", "Done",

JOptionPane.INFORMATION\_MESSAGE);

CategoryTitle.setText(null);

ResultSet res1 = st.executeQuery("select max(Cat\_ID) from Category");

if (res1.next()) {

int catID = res1.getInt(1);

++catID;

CategoryID.setText(String.valueOf(catID));

CateSave.setEnabled(true);

} } }

}catch(Exception ev) {System.out.println(ev); }

} });

JButton CateDelete = new JButton("Delete");

CateDelete.setBounds(130, 510, 150, 33);

CateDelete.setBackground(Color.red);

CateDelete.setForeground(Color.white);

CateDelete.setFont(new Font("LucidaGrande", Font.PLAIN, 26));

CATEGORYPanel.add(CateDelete);

CateDelete.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

if(ct.getSelectedRowCount()==1) {

try {

Connection con=DriverManager.getConnection("jdbc:ucanaccess://C:\\Users\\AHsan Usmani\\Desktop\\Programs\\Project GMS\\src\\sms.accdb");

Statement st=con.createStatement();

int i=st.executeUpdate("Delete from Category where Cat\_ID='"+CategoryID.getText()+"'");

if(i>0) {

Catmodel.removeRow(ct.getSelectedRow());

JOptionPane.showMessageDialog(null,"Category Deleted", "Warrning",

JOptionPane.INFORMATION\_MESSAGE);

CategoryTitle.setText(null);

ResultSet res1 = st.executeQuery("select max(Cat\_ID) from Category");

if (res1.next()) {

int catID = res1.getInt(1);

++catID; CategoryID.setText(String.valueOf(catID));

CateSave.setEnabled(true); } }

}catch(Exception ev) {System.out.println(ev);} }

else {

if(ct.getRowCount()==0)

JOptionPane.showMessageDialog(null,"Empty row", "Warrning",JOptionPane.WARNING\_MESSAGE);

else

JOptionPane.showMessageDialog(null,"Please select the row", "Warrning",

JOptionPane.WARNING\_MESSAGE); } } });

ct.addMouseListener(new MouseAdapter() {

public void mouseClicked(MouseEvent e) {

String empid = Catmodel.getValueAt(ct.getSelectedRow(), 0).toString();

String CateName=Catmodel.getValueAt(ct.getSelectedRow(), 1).toString();

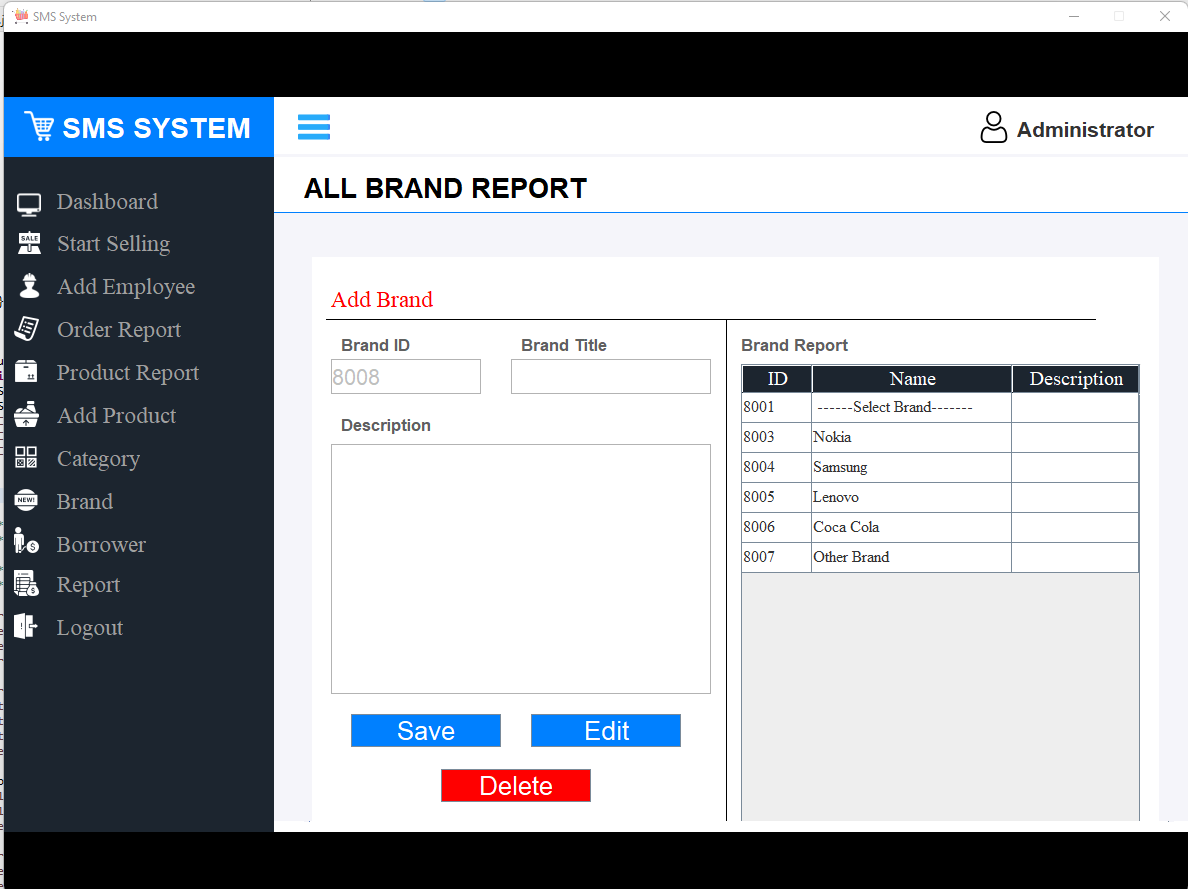
CategoryID.setText(empid);

CategoryTitle.setText(CateName);

CateSave.setEnabled(false);

} });

BRAND WINDOW SCREEN



**Output form information**

Brand panel is used to classified the products in diffirent brands in super market management system in this panel we can add, remove and modify brands.

**Coding of Brand Panel**

JPanel BrandPanel = new JPanel();

BrandPanel.setBackground(Color.white);

BrandPanel.setLayout(null);

TP.add(BrandPanel);

JLabel BrandText1 = new JLabel("Add Brand");

BrandText1.setBounds(20, 0, 750, 80);

BrandText1.setForeground(Color.red);

BrandText1.setFont(new Font("Times New Roman", Font.PLAIN, 23));

BrandPanel.add(BrandText1);

JSeparator BrandSeP1 = new JSeparator();

BrandSeP1.setBounds(15, 60, 770, 1);

BrandSeP1.setBorder(BorderFactory.createLineBorder(new Color(0, 0, 0), 1));

BrandPanel.add(BrandSeP1);

JLabel BrandCode = new JLabel("Brand ID"); // ProCode=Product Code

BrandCode.setBounds(30, 70, 150, 30);

BrandCode.setForeground(new Color(96, 96, 96));

BrandCode.setFont(new Font("LucidaGrande", Font.BOLD, 17));

BrandPanel.add(BrandCode);

JTextField BrandID = new JTextField();

BrandID.setFont(new Font("LucidaGrande", Font.PLAIN, 22));

BrandID.setBounds(20, 100, 150, 35);

BrandID.setEditable(false);

BrandID.setBackground(new Color(255, 255, 255));

BrandID.setForeground(new Color(192, 192, 192));

BrandID.setBorder(BorderFactory.createLineBorder(new Color(179, 179, 179)));

BrandPanel.add(BrandID);

try {

Connection con = DriverManager.getConnection("jdbc:ucanaccess://C:\\Users

\\AHsan Usmani\\Desktop\\Programs\\Project GMS\\src\\sms.accdb");

Statement st1 = con.createStatement();

ResultSet res1 = st1.executeQuery("select max(Brand\_ID) from Brand");

if (res1.next()) {

int BID = res1.getInt(1);

++BID;

BrandID.setText(String.valueOf(BID)); }

} catch (Exception evt) { System.out.println(evt); }

JLabel BrandT = new JLabel("Brand Title"); // CateT= Category title

BrandT.setBounds(210, 70, 150, 30);

BrandT.setForeground(new Color(96, 96, 96));

BrandT.setFont(new Font("LucidaGrande", Font.BOLD, 17));

BrandPanel.add(BrandT);

JTextField BrandTitle = new JTextField();

BrandTitle.setFont(new Font("LucidaGrande", Font.PLAIN, 22));

BrandTitle.setBounds(200, 100, 200, 35);

BrandTitle.setForeground(new Color(192, 192, 192));

BrandTitle.setBorder(BorderFactory.createLineBorder(new Color(179, 179, 179)));

BrandPanel.add(BrandTitle);

JLabel Berror1 = new JLabel(" \u274C Enter the Brand Name");

Berror1.setForeground(Color.red);

Berror1.setFont(new Font("LucidaGrande", Font.PLAIN, 11));

Berror1.setBounds(203, 138, 300, 15);

Berror1.setVisible(false);

BrandPanel.add(Berror1);

BrandTitle.addMouseListener(new MouseAdapter() {

public void mouseClicked(MouseEvent e) {

BrandTitle.setBorder(BorderFactory.createLineBorder(new Color(179, 179, 179)));

Berror1.setVisible(false);

} });

JLabel BrandDes = new JLabel("Description"); BrandDes.setBounds(30, 150, 150, 30);

BrandDes.setForeground(new Color(96, 96, 96));

BrandDes.setFont(new Font("LucidaGrande", Font.BOLD, 17));

BrandPanel.add(BrandDes);

JTextArea BrandDescription = new JTextArea();

BrandDescription.setFont(new Font("LucidaGrande", Font.PLAIN, 22));

BrandDescription.setBounds(20, 185, 380, 250);

BrandDescription.setForeground(new Color(192, 192, 192));

BrandDescription.setBorder(BorderFactory.createLineBorder(new Color(179, 179, 179)));

BrandPanel.add(BrandDescription);

JSeparator BrandSeP2 = new JSeparator(JSeparator.VERTICAL);

BrandSeP2.setBounds(415, 60, 1, 560);

BrandSeP2.setBorder(BorderFactory.createLineBorder(new Color(0, 0, 0), 1));

BrandPanel.add(BrandSeP2);

JLabel BrandRep = new JLabel("Brand Report");

BrandRep.setBounds(430, 70, 150, 30);

BrandRep.setForeground(new Color(96, 96, 96));

BrandRep.setFont(new Font("LucidaGrande", Font.BOLD, 17));

BrandPanel.add(BrandRep);

//BrandTable starting

DefaultTableModel Brandmodel = new DefaultTableModel();

String Bcol[] = {"ID","Name","Description" };

JTable Bt = new JTable(Brandmodel); // Bt = Brand Table

Bt.setRowHeight(30);

Brandmodel.setColumnIdentifiers(Bcol);

JTableHeader bth = Bt.getTableHeader(); // bth = Brand table header

bth.setFont(new Font("Times New Roman", Font.PLAIN, 20));

bth.setForeground(Color.white);

bth.setBackground(new Color(28, 37, 47));

bth.getColumnModel().getColumn(0).setMaxWidth(70);

bth.getColumnModel().getColumn(1).setMaxWidth(200);

bth.getColumnModel().getColumn(1).setMinWidth(200);

Bt.setFont(new Font("Times New Roman", Font.PLAIN, 16));

Bt.getTableHeader().setReorderingAllowed(false);

JScrollPane Brandpane = new JScrollPane();

Brandpane.setViewportView(Bt);

Brandpane.setBounds(430,105,400,500);

BrandPanel.add(Brandpane);

try {

Connection con= DriverManager.getConnection("jdbc:ucanaccess://C:\\Users\\AHsan

Usmani\\Desktop\\Programs\\Project GMS\\src\\sms.accdb");

Statement st1=con.createStatement();

ResultSet res= st1.executeQuery("Select \* from Brand"); {

while(res.next()) {

String BID=res.getString("Brand\_ID");

String BName=res.getString("Brand\_Title");

String BDes=res.getString("Brand\_Description");

String BDetails[]= {BID, BName,BDes};

Brandmodel.addRow(BDetails);

} }

}catch(Exception e) {System.out.println(e);}

//BrandTable Ending

JButton BrandSave = new JButton("Save");

BrandSave.setBounds(40, 455, 150, 33);

BrandSave.setBackground(new Color(0, 128, 255));

BrandSave.setForeground(Color.white);

BrandSave.setFont(new Font("LucidaGrande", Font.PLAIN, 26));

BrandPanel.add(BrandSave);

BrandSave.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

if (BrandTitle.getText().length() == 0) {

BrandTitle.setBorder(BorderFactory.createLineBorder(Color.red));

Berror1.setVisible(true); }

else {

try {

Connection con = DriverManager.getConnection("jdbc:ucanaccess://C:\\Users\\AHsan

Usmani\\Desktop\\Programs\\Project GMS\\src\\sms.accdb");

Statement st = con.createStatement();

int i = st.executeUpdate("insert into Brand(Brand\_ID,Brand\_Title) values('" + BrandID.getText()+ "','" + BrandTitle.getText() + "')");

if (i > 0) {

ImageIcon ig1 = new ImageIcon("C:\\Users\\AHsan Usmani\\Desktop\\Programs

\\Project GMS\\src\\done.png");

JOptionPane.showMessageDialog(null, "Brand add successfully.", "Done",

JOptionPane.INFORMATION\_MESSAGE, ig1);

ResultSet res1 = st.executeQuery("select max(Brand\_ID) from Brand");

if (res1.next()) {

Brandmodel.addRow(new Object[] {BrandID.getText(),BrandTitle.getText()});

int BID = res1.getInt(1);

++BID;

BrandID.setText(String.valueOf(BID));

BrandTitle.setText("");

}

}

}

catch (Exception evt) { System.out.println(evt);

}

}

}

});

JButton BrandEdit = new JButton("Edit");

BrandEdit.setBounds(220, 455, 150, 33);

BrandEdit.setBackground(new Color(0, 128, 255));

BrandEdit.setForeground(Color.white);

BrandEdit.setFont(new Font("LucidaGrande", Font.PLAIN, 26));

BrandPanel.add(BrandEdit);

BrandEdit.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

try {

Connection con=DriverManager.getConnection("jdbc:ucanaccess://C:\\Users\\AHsan

Usmani\\Desktop\\Programs\\Project GMS\\src\\sms.accdb");

Statement st=con.createStatement();

int i=st.executeUpdate("update Brand set Brand\_Title='"+BrandTitle.getText()+"'

where Brand\_ID='"+BrandID.getText()+"' ");

if(i>0) {

int j=Bt.getSelectedRow();

if(j>=0) {

Brandmodel.setValueAt(BrandTitle.getText(), j, 1);

JOptionPane.showMessageDialog(null, "Brand add successfully.", "Done",

JOptionPane.INFORMATION\_MESSAGE);

BrandTitle.setText(null);

ResultSet res1 = st.executeQuery("select max(Brand\_ID) from Brand");

if (res1.next()) {

int BID = res1.getInt(1);

++BID;

BrandID.setText(String.valueOf(BID));

BrandSave.setEnabled(true);

}

}

}

}

catch(Exception ev) {System.out.println(ev); }

}

});

JButton BrandDelete = new JButton("Delete");

BrandDelete.setBounds(130, 510, 150, 33);

BrandDelete.setBackground(Color.red);

BrandDelete.setForeground(Color.white);

BrandDelete.setFont(new Font("LucidaGrande", Font.PLAIN, 26));

BrandPanel.add(BrandDelete);

BrandDelete.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent ev) {

if(Bt.getSelectedRowCount()==1) {

try {

Connection con=DriverManager.getConnection("jdbc:ucanaccess://C:\\Users\\AHsan

Usmani\\Desktop\\Programs\\Project GMS\\src\\sms.accdb");

Statement st=con.createStatement();

int i=st.executeUpdate("Delete from Brand where Brand\_ID='"+BrandID.getText()+"'");

if(i>0) {

Brandmodel.removeRow(Bt.getSelectedRow());

JOptionPane.showMessageDialog(null,"Row Deleted", "Warrning", JOptionPane.INFORMATION\_ MESSAGE);

BrandTitle.setText(null);

ResultSet res1 = st.executeQuery("select max(Brand\_ID) from Brand");

if (res1.next()) {

int BID = res1.getInt(1);

++BID;

BrandID.setText(String.valueOf(BID));

BrandSave.setEnabled(true);

BrandSave.setEnabled(true); }

}

} catch(Exception e) {System.out.println(e);}

}

else {

if(et.getRowCount()==0)

JOptionPane.showMessageDialog(null,"Empty row", "Warrning",JOptionPane.WARNING\_MESSAGE);

else{

JOptionPane.showMessageDialog(null,"Please select the row", "Warrning",JOptionPane.WARNING\_MESSAGE);

}

}

}

});

Bt.addMouseListener(new MouseAdapter() {

public void mouseClicked(MouseEvent e) {

String bid = Brandmodel.getValueAt(Bt.getSelectedRow(), 0).toString();

String bName = Brandmodel.getValueAt(Bt.getSelectedRow(), 1).toString();

BrandID.setText(bid);

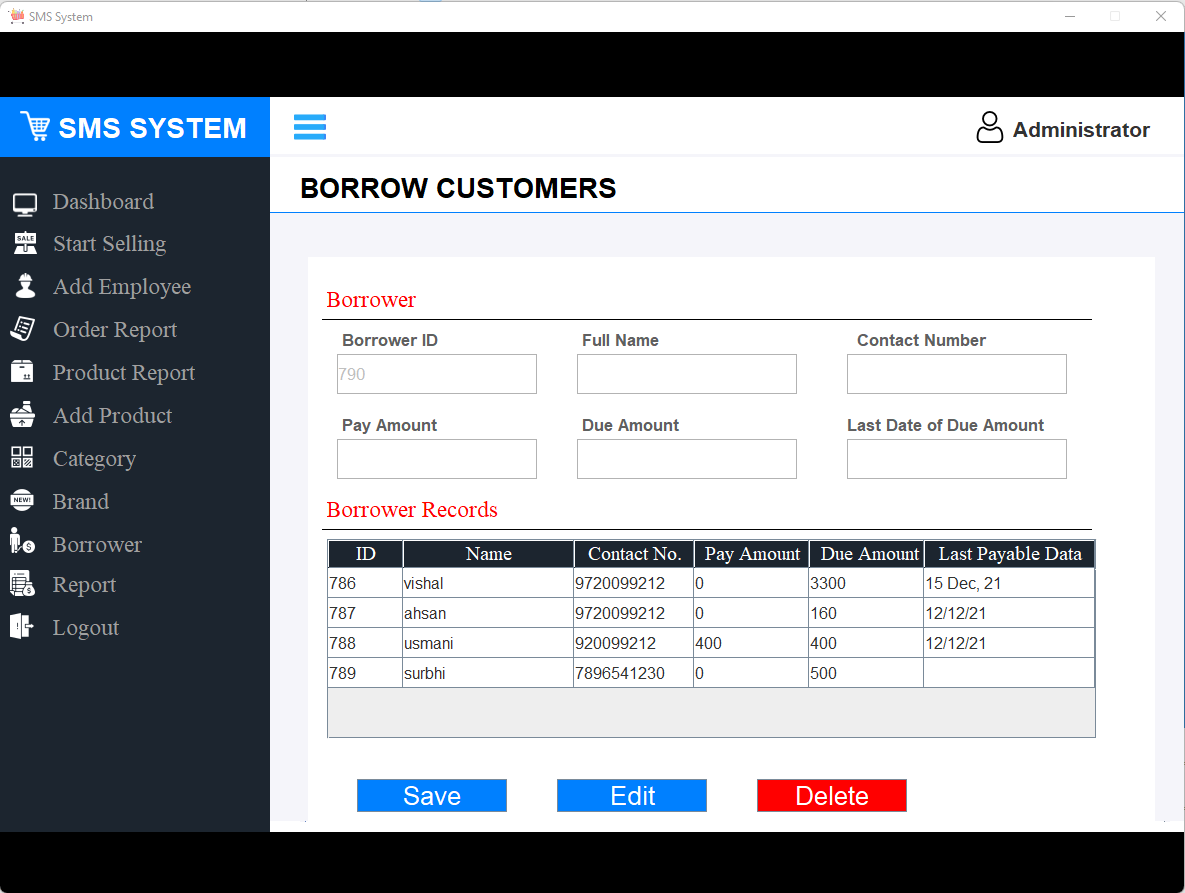
BrandTitle.setText(bName);

BrandSave.setEnabled(false);

}

});

BORROWER WINDOW SCREEN



**Output form information**

Borrower panel is used for taking borrow to the customer in super market management system in this panel we can add, remove and modify borrower details.

**Coding of Borrower Panel**

JPanel BorrowerPanel = new JPanel();

BorrowerPanel.setBackground(Color.white);

BorrowerPanel.setLayout(null);

TP.add(BorrowerPanel);

JLabel BorrowerPanelText1 = new JLabel("Borrower");

BorrowerPanelText1.setBounds(20, 0, 750, 80);

BorrowerPanelText1.setForeground(Color.red);

BorrowerPanelText1.setFont(new Font("Times New Roman", Font.PLAIN, 23));

BorrowerPanel.add(BorrowerPanelText1);

JSeparator BorrowerSeP1 = new JSeparator();

BorrowerSeP1.setBounds(15, 60, 770, 1);

BorrowerSeP1.setBorder(BorderFactory.createLineBorder(new Color(0, 0, 0), 1));

BorrowerPanel.add(BorrowerSeP1);

JLabel BID = new JLabel("Borrower ID"); // BID= Borrower ID

BID.setBounds(35, 65, 122, 30);

BID.setForeground(new Color(96, 96, 96));

BID.setFont(new Font("LucidaGrande", Font.BOLD, 17));

BorrowerPanel.add(BID);

JTextField BorrowerID = new JTextField();

BorrowerID.setBounds(30, 95, 200, 40);

BorrowerID.setBorder(BorderFactory.createLineBorder(new Color(179, 179, 179)));

BorrowerID.setBackground(Color.white);

BorrowerID.setForeground(new Color(192, 192, 192));

BorrowerID.setEditable(false);

BorrowerID.setFont(new Font("LucidaGrande", Font.PLAIN, 17));

BorrowerPanel.add(BorrowerID);

try {

Connection con = DriverManager.getConnection("jdbc:ucanaccess://C:\\Users\\AHsan

Usmani\\Desktop\\Programs\\Project GMS\\src\\sms.accdb");

Statement st1 = con.createStatement();

ResultSet rus = st1.executeQuery("select max(Bor\_ID) from Borrower");

if (rus.next()) {

int BorId = rus.getInt(1);

++BorId;

BorrowerID.setText(String.valueOf(BorId));

}

} catch (Exception evt) {System.out.println(evt); }

JLabel BN = new JLabel("Full Name"); // BN= Borrower Name

BN.setBounds(275, 65, 152, 30);

BN.setForeground(new Color(96, 96, 96));

BN.setFont(new Font("LucidaGrande", Font.BOLD, 17));

BorrowerPanel.add(BN);

JTextField BorrowerName = new JTextField();

BorrowerName.setBounds(270, 95, 220, 40);

BorrowerName.setBorder(BorderFactory.createLineBorder(new Color(179, 179, 179)));

BorrowerName.setForeground(new Color(192, 192, 192));

BorrowerName.setFont(new Font("LucidaGrande", Font.PLAIN, 17));

BorrowerPanel.add(BorrowerName);

JLabel B1error = new JLabel(" \u274C Enter the Name");

B1error.setForeground(Color.red);

B1error.setFont(new Font("LucidaGrande", Font.PLAIN, 11));

B1error.setBounds(273, 138, 300, 15);

B1error.setVisible(false);

BorrowerPanel.add(B1error);

BorrowerName.addMouseListener(new MouseAdapter() {

public void mouseClicked(MouseEvent e) {

BorrowerName.setBorder(BorderFactory.createLineBorder(new Color(179, 179, 179)));

B1error.setVisible(false);

} });

JLabel BNum = new JLabel("Contact Number"); // BNum= Borrower Number

BNum.setBounds(550, 65, 152, 30);

BNum.setForeground(new Color(96, 96, 96));

BNum.setFont(new Font("LucidaGrande", Font.BOLD, 17));

BorrowerPanel.add(BNum);

JTextField BorrowerNumber = new JTextField();

BorrowerNumber.setBounds(540, 95, 220, 40);

BorrowerNumber.setBorder(BorderFactory.createLineBorder(new Color(179, 179, 179)));

BorrowerNumber.setForeground(new Color(192, 192, 192));

BorrowerNumber.setFont(new Font("LucidaGrande", Font.PLAIN, 17));

BorrowerPanel.add(BorrowerNumber);

JLabel B2error = new JLabel(" \u274C Enter contact number");

B2error.setForeground(Color.red);

B2error.setFont(new Font("LucidaGrande", Font.PLAIN, 11));

B2error.setBounds(543, 138, 300, 15);

B2error.setVisible(false);

BorrowerPanel.add(B2error);

BorrowerPanel.addMouseListener(new MouseAdapter() {

public void mouseClicked(MouseEvent e) {

BorrowerPanel.setBorder(BorderFactory.createLineBorder(new Color(179, 179, 179)));

B2error.setVisible(false);

} });

JLabel BPA = new JLabel("Pay Amount"); // BPA= Borrower Pay Amount

BPA.setBounds(35, 150, 152, 30);

BPA.setForeground(new Color(96, 96, 96));

BPA.setFont(new Font("LucidaGrande", Font.BOLD, 17));

BorrowerPanel.add(BPA);

JTextField PayAmount = new JTextField();

PayAmount.setBounds(30, 180, 200, 40);

PayAmount.setBorder(BorderFactory.createLineBorder(new Color(179, 179, 179)));

PayAmount.setForeground(new Color(192, 192, 192));

PayAmount.setFont(new Font("LucidaGrande", Font.PLAIN, 17));

BorrowerPanel.add(PayAmount);

JLabel B3error = new JLabel(" \u274C Enter the amount");

B3error.setForeground(Color.red);

B3error.setFont(new Font("LucidaGrande", Font.PLAIN, 11));

B3error.setBounds(33, 223, 300, 15);

B3error.setVisible(false);

BorrowerPanel.add(B3error);

PayAmount.addMouseListener(new MouseAdapter() {

public void mouseClicked(MouseEvent e) {

PayAmount.setBorder(BorderFactory.createLineBorder(new Color(179, 179, 179)));

B3error.setVisible(false);

} });

JLabel BDA = new JLabel("Due Amount"); // BDA= Borrower Due Amount

BDA.setBounds(275, 150, 152, 30);

BDA.setForeground(new Color(96, 96, 96));

BDA.setFont(new Font("LucidaGrande", Font.BOLD, 17));

BorrowerPanel.add(BDA);

JTextField DueAmount = new JTextField();

DueAmount.setBounds(270, 180, 220, 40);

DueAmount.setBorder(BorderFactory.createLineBorder(new Color(179, 179, 179)));

DueAmount.setForeground(new Color(192, 192, 192));

DueAmount.setFont(new Font("LucidaGrande", Font.PLAIN, 17));

BorrowerPanel.add(DueAmount);

JLabel B4error = new JLabel(" \u274C Enter the due amount");

B4error.setForeground(Color.red);

B4error.setFont(new Font("LucidaGrande", Font.PLAIN, 11));

B4error.setBounds(273, 223, 300, 15);

B4error.setVisible(false);

BorrowerPanel.add(B4error);

DueAmount.addMouseListener(new MouseAdapter() {

public void mouseClicked(MouseEvent e) {

DueAmount.setBorder(BorderFactory.createLineBorder(new Color(179, 179, 179)));

B4error.setVisible(false);

} });

JLabel DOPA = new JLabel("Last Date of Due Amount");

DOPA.setBounds(540, 150, 200, 30);

DOPA.setForeground(new Color(96, 96, 96));

DOPA.setFont(new Font("LucidaGrande", Font.BOLD, 17));

BorrowerPanel.add(DOPA);

JTextField LastDate = new JTextField();

LastDate.setBounds(540, 180, 220, 40);

LastDate.setText("");

LastDate.setBorder(BorderFactory.createLineBorder(new Color(179, 179, 179)));

LastDate.setForeground(new Color(192, 192, 192));

LastDate.setFont(new Font("LucidaGrande", Font.PLAIN, 17));

BorrowerPanel.add(LastDate);

JLabel BR = new JLabel("Borrower Records");// Borrower Report

BR.setBounds(20, 210, 750, 80);

BR.setForeground(Color.red);

BR.setFont(new Font("Times New Roman", Font.PLAIN, 23));

BorrowerPanel.add(BR);

JSeparator BorrowerReport = new JSeparator();

BorrowerReport.setBounds(15, 270, 770, 1);

BorrowerReport.setBorder(BorderFactory.createLineBorder(new Color(0, 0, 0), 1));

BorrowerPanel.add(BorrowerReport);

DefaultTableModel Bromodel = new DefaultTableModel();

String Brocol[] = {"ID","Name","Contact No.","Pay Amount"," Due Amount",

"Last Payable Data" };

JTable Brot = new JTable(Bromodel);

Brot.setRowHeight(30);

Bromodel.setColumnIdentifiers(Brocol);

JTableHeader brth = Brot.getTableHeader(); // brth = Borrower table header

brth.setFont(new Font("Times New Roman", Font.PLAIN, 20));

brth.setForeground(Color.white);

brth.setBackground(new Color(28, 37, 47));

brth.getColumnModel().getColumn(0).setMaxWidth(80);

brth.getColumnModel().getColumn(2).setMaxWidth(120);

brth.getColumnModel().getColumn(2).setMinWidth(120);

brth.getColumnModel().getColumn(3).setMaxWidth(115);

brth.getColumnModel().getColumn(3).setMinWidth(115);

brth.getColumnModel().getColumn(4).setMaxWidth(115);

brth.getColumnModel().getColumn(4).setMinWidth(115);

Brot.setFont(new Font("LucidaGrande", Font.PLAIN, 16));

Brot.getTableHeader().setReorderingAllowed(false);

JScrollPane Borrowerpane = new JScrollPane();

Borrowerpane.setViewportView(Brot);

Borrowerpane.setBounds(20,280,770,200);

BorrowerPanel.add(Borrowerpane);

try {

Connection con = DriverManager.getConnection("jdbc:ucanaccess://C:\\Users\\AHsan

Usmani\\Desktop\\Programs\\Project GMS\\src\\sms.accdb");

Statement st1=con.createStatement();

ResultSet res=st1.executeQuery("select \* from Borrower");

while(res.next()) {

String BroID=res.getString("Bor\_ID");

String BroName=res.getString("Bor\_Name");

String BroContact=res.getString("Contact\_No");

String PA=res.getString("PayAmount");

String DA=res.getString("DueAmount");

String BLastDate=res.getString("LastPayableDate");

Bromodel.addRow(new Object[] {BroID, BroName, BroContact, PA,DA,BLastDate});

} }

catch(Exception e) {System.out.println(e); }

JButton BorrowerSave = new JButton("Save");

BorrowerSave.setBounds(50, 520, 150, 33);

BorrowerSave.setBackground(new Color(0, 128, 255));

BorrowerSave.setForeground(Color.white);

BorrowerSave.setFont(new Font("LucidaGrande", Font.PLAIN, 26));

BorrowerPanel.add(BorrowerSave);

BorrowerSave.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

if (BorrowerName.getText().length() == 0) {

BorrowerName.setBorder(BorderFactory.createLineBorder(Color.red));

B1error.setVisible(true); }

else if (BorrowerNumber.getText().length() == 0) { BorrowerNumber.setBorder(BorderFactory.createLineBorder(Color.red));

B2error.setVisible(true); }

else if (PayAmount.getText().length() == 0) { PayAmount.setBorder(BorderFactory.createLineBorder(Color.red));

B3error.setVisible(true); }

else if (DueAmount.getText().length() == 0) {

DueAmount.setBorder(BorderFactory.createLineBorder(Color.red));

B4error.setVisible(true); }

else {

try {

Connection con = DriverManager.getConnection("jdbc:ucanaccess://C:\\Users\\AHsan

Usmani\\Desktop\\Programs\\Project GMS\\src\\sms.accdb");

Statement st1 = con.createStatement();

int i = st1.executeUpdate("insert into Borrower(Bor\_ID,Bor\_Name, Contact\_No,PayAmount,DueAmount,LastPayableDate) values('"+ BorrowerID.getText() + "','" + BorrowerName.getText() + "','"+ BorrowerNumber.getText() + "', '" +PayAmount.getText() + "', '"+ DueAmount.getText() + "','" + LastDate.getText() + "')");

if (i > 0) {

ImageIcon ig1 = new ImageIcon("C:\\Users\\AHsan Usmani\\Desktop\\Programs\\Project

GMS\\src\\done.png");

JOptionPane.showMessageDialog(null, "Borrwer added successfully.", "Done",

JOptionPane.INFORMATION\_MESSAGE, ig1);

Bromodel.addRow(new Object[] {BorrowerID.getText(),BorrowerName.getText(), BorrowerNumber.getText(),PayAmount.getText(),DueAmount.getText(),LastDate.getText() });

ResultSet rus = st1.executeQuery("select max(Bor\_ID) from Borrower");

if (rus.next()) {

int BorId = rus.getInt(1);

++BorId;

BorrowerID.setText(String.valueOf(BorId));

BorrowerName.setText("");

BorrowerNumber.setText("");

PayAmount.setText("");

DueAmount.setText("");

LastDate.setText("");

}

}

} catch (Exception evt) { System.out.println(evt); }

}

}

});

JButton BorrowerEdit = new JButton("Edit");

BorrowerEdit.setBounds(250, 520, 150, 33);

BorrowerEdit.setBackground(new Color(0, 128, 255));

BorrowerEdit.setForeground(Color.white);

BorrowerEdit.setFont(new Font("LucidaGrande", Font.PLAIN, 26));

BorrowerPanel.add(BorrowerEdit);

BorrowerEdit.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

try {

Connectioncon=DriverManager.getConnection("jdbc:ucanaccess://C:\\Users

\\AHsan Usmani\\Desktop\\Programs\\Project GMS\\src\\sms.accdb");

Statement st=con.createStatement();

int i=st.executeUpdate("update Borrower set Bor\_Name='"+BorrowerName

.getText()+"', Contact\_No='"+BorrowerNumber.getText()+"', PayAmount='"

+PayAmount.getText()+"', DueAmount='"+DueAmount.getText()+"', LastPayableDate

= '"+LastDate.getText()+"' where Bor\_ID ='"+BorrowerID.getText()+"' ");

if(i>0) {

int j=Brot.getSelectedRow();

if(j>=0) {

Bromodel.setValueAt(BorrowerName.getText(), j, 1);

Bromodel.setValueAt(BorrowerNumber.getText(), j, 2);

Bromodel.setValueAt(PayAmount.getText(), j, 3);

Bromodel.setValueAt(DueAmount.getText(), j, 4);

Bromodel.setValueAt(LastDate.getText(), j, 5);

ImageIcon ig1 = new ImageIcon("C:\\Users\\AHsan Usmani\\Desktop

\\Programs\\Project GMS\\src\\done.png");

JOptionPane.showMessageDialog(null, "Borrwer details updated

successfully.", "Done",JOptionPane.INFORMATION\_MESSAGE, ig1);

ResultSet rus = st.executeQuery("select max(Bor\_ID) from Borrower");

if (rus.next()) {

int BorId = rus.getInt(1);

++BorId; BorrowerID.setText(String.valueOf(BorId));

BorrowerName.setText("");

BorrowerNumber.setText("");

PayAmount.setText("");

DueAmount.setText("");

LastDate.setText("");

}

BorrowerSave.setEnabled(true);

Brot.removeRowSelectionInterval(j, 0);

}

}

} catch(Exception ev) {System.out.print(ev);}

}

});

JButton BorrowerDelete = new JButton("Delete");

BorrowerDelete.setBounds(450, 520, 150, 33);

BorrowerDelete.setBackground(Color.red);

BorrowerDelete.setForeground(Color.white);

BorrowerDelete.setFont(new Font("LucidaGrande", Font.PLAIN, 26));

BorrowerPanel.add(BorrowerDelete);

BorrowerDelete.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

if(Brot.getSelectedRowCount()==1) {

try {

Connection con=DriverManager.getConnection("jdbc:ucanaccess://C:\\Users\\AHsan

Usmani\\Desktop\\Programs\\Project GMS\\src\\sms.accdb");

Statement st=con.createStatement();

int i=st.executeUpdate("Delete from Borrower where Bor\_ID='"+BorrowerID.getText()+"'");

if(i>0) { Bromodel.removeRow(Brot.getSelectedRow()) JOptionPane.showMessageDialog(null,"Borrower Deleted", "Warrning",

JOptionPane.INFORMATION\_MESSAGE);

ResultSet rus = st.executeQuery("select max(Bor\_ID) from Borrower");

if (rus.next()) {

int BorId = rus.getInt(1);

++BorId;

BorrowerID.setText(String.valueOf(BorId)); BorrowerName.setText("");

BorrowerNumber.setText("");

PayAmount.setText("");

DueAmount.setText("");

LastDate.setText("");

}

}

}

catch(Exception ev) {System.out.println(ev);} }

else {

if(Brot.getRowCount()==0)

JOptionPane.showMessageDialog(null,"Empty row", "Warrning",

JOptionPane.WARNING\_MESSAGE);

else JOptionPane.showMessageDialog(null,"Please select the row", "Warrning",JOptionPane.WARNING\_MESSAGE);

}

}

});

Brot.addMouseListener(new MouseAdapter() {

public void mouseClicked(MouseEvent e) {

BorrowerID.setText(Bromodel.getValueAt(Brot.getSelectedRow(), 0).toString());

BorrowerName.setText(Bromodel.getValueAt(Brot.getSelectedRow(), 1).toString());

BorrowerNumber.setText(Bromodel.getValueAt(Brot.getSelectedRow(), 2).toString());

PayAmount.setText(Bromodel.getValueAt(Brot.getSelectedRow(), 3).toString());

DueAmount.setText(Bromodel.getValueAt(Brot.getSelectedRow(),4).toString());

LastDate.setText(Bromodel.getValueAt(Brot.getSelectedRow(), 5).toString());

BorrowerSave.setEnabled(false);

}

});

**CONCLUSION**

**Conclusion**

This software has been developed designed to reduce the time taken to handle the sales activity. It is designed to replace an existing manual record system for reducing time taken for calculations and for storing data. This system has been developed with oops concepts. The system is strong to handle daily operations where the database is cleared over certain time. This system will reduce manual work, calculations and will also provide periodic reports any time.

Using this type of mythology we can say that the shopping is very easy and it is very useful for those people, which are not having much time. Time conversion, fast working and every goods are available in one place. Every person are depend itself to shopping and marketing. This type of system is very useful.

**BIBLIOGRAPHY**

**Bibliography**

* Vishal Mega Mart Muzaffarnagar
* javapoint.com
* geeksforgeeks.org
* Java black book 6 Dr. R Nagshwara
* The Complete reference