INTRODUCTION

A bookstore lab mini project is a software development project aimed at creating a digital platform for managing and selling books. The project involves creating a web application that allows users to browse, search, and purchase books online. The platform should also have an inventory management system for tracking books in stock, adding new books to the catalog, and updating book information.Overall, the bookstore lab mini project is an exciting opportunity for students to learn web development and software engineering skills while creating a real-world application that can be used by customers to purchase books online.

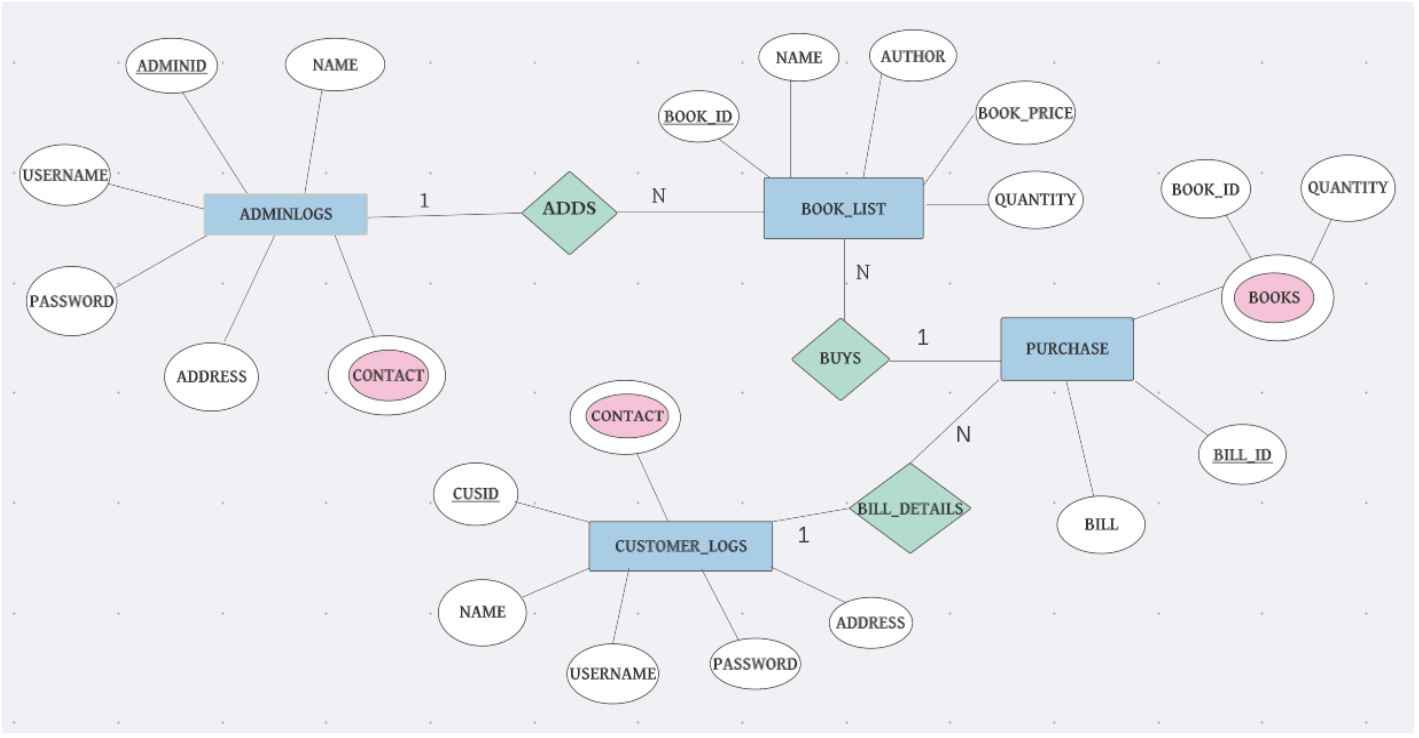
There are several needs and advantages of an online bookstore, some of which are:

- Convenience
- Wide range of books
- Quick delivery
- User reviews
- Eco-friendly

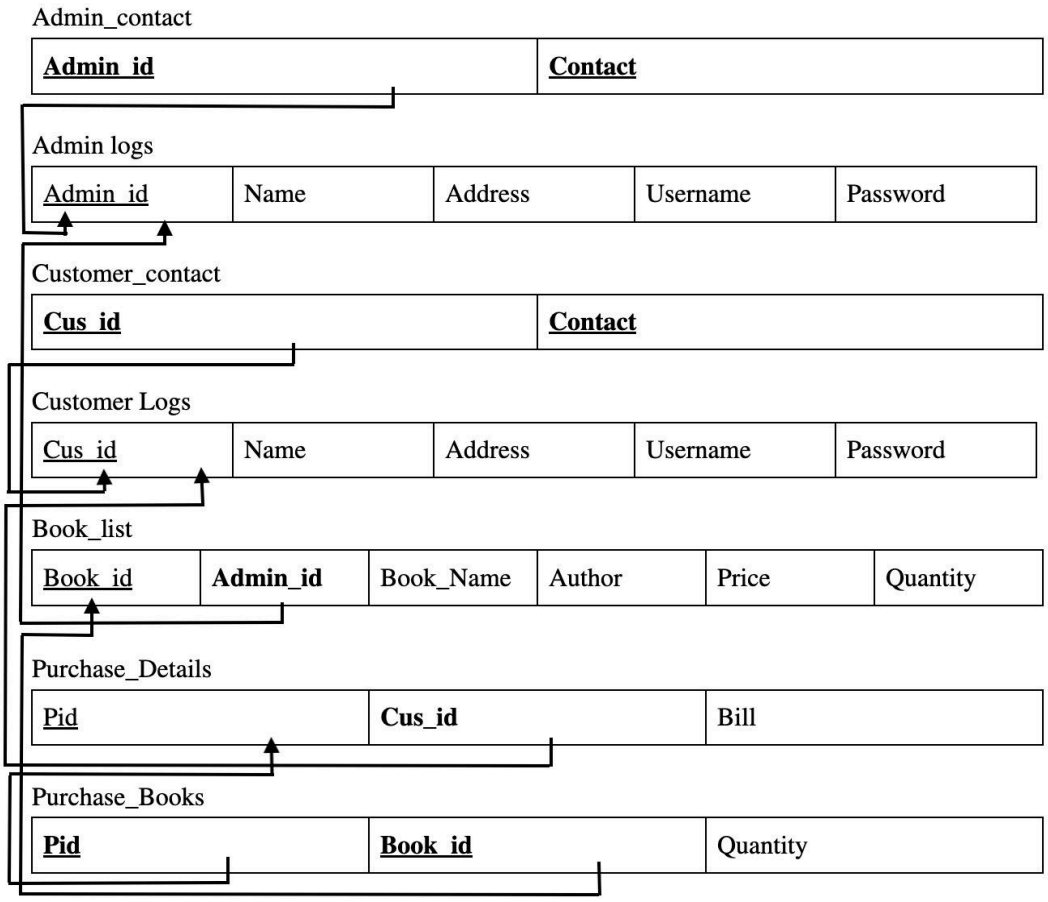
Overall, an online bookstore offers a convenient, cost-effective, and personalized shopping experience for customers, while also providing a larger selection of books and eliminating the need for physical travel.

We have used Swing in Java to design the frontend of our website because it is a plat-form independent, having a rich set of components and customizable look and feel.

ER DIAGRAM



RELATION SCHEMA MODEL



NORMALIZATION

Normalization is a process in database management that involves organizing data in a structured way to minimize data redundancy and ensure data integrity.

So we have used normalization to ensure improved data consistency,simplified database design,Efficient use of storage space etc.,

First Normal Form (1NF): No Multivalued Attribute

It states that an attribute of a table cannot hold multiple values. It must hold only single-valued Attributes.

First normal form disallows the multi-valued Attribute

Relation ADMIN is not in 1NF because of multi-valued attribute CONTACT.So relation ADMIN is divided into two relations namely Admin_Contact and AdminLogs.Similarly for CUSTOMER divided into CustomerLogs and Customer_Contact.

ADMIN

Table: AdminLogs

Columns:

Admin_id	int PK
Name	varchar(45)
Address	varchar(45)
Username	varchar(45)
Password	varchar(45)

Table: Admin_Contact

Columns:

Adminid	int PK
Contact	varchar(45) PK

Admin_id	Name	Address	Username	Password
1	Sruthi	Coimbatore	sruthi	123
2	Abinaya	Permbalur	abi	123
NULL	NULL	NULL	NULL	NULL

Adminid	Contact
1	9876543219
2	9876453210
NULL	NULL

CUSTOMER

Table: CustomerLogs

Columns:

Cus_id	int PK
Name	varchar(45)
Address	varchar(45)
Username	varchar(45)
Password	varchar(45)

Table: Customer_Contact

Columns:

cusid	int PK
Contact	varchar(45) PK

Cus_id	Name	Address	Username	Password
1	Sruthi	Coimbatore	sruthi	123
2	Karthika	Salem	karthika	123
3	Kavya	Kumbakonam	kavya	123
4	Divya	Coimbatore	divya	123
NULL	NULL	NULL	NULL	NULL

cusid	Contact
1	9876543214
2	9876543210
3	9876543217
4	8765432106
NULL	NULL

PURCHASE

Table: Purchase_Books

Columns:

p_id	int PK
bookid	int PK
Quantity	int

Table: Purchase_details

Columns:

Pid	int PK
Cus_id	int
Bill	int

Pid	Cus_id	Bill
1	1	800
2	1	1550
3	1	1550
4	1	1550
5	1	200
6	3	1380
7	1	1440
NULL	NULL	NULL

p_id	bookid	Quantity
1	101	1
2	103	1
3	101	1
4	103	1
4	104	1
5	105	1
6	106	2
7	107	2
NULL	NULL	NULL

SECOND NORMAL FORM

For a table to be in the Second Normal Form,

It should be in the First Normal form.

And, it should not have Partial Dependency.

All the tables satisfy the second normal form since it does not have partial dependency and also in the first normal form.

THIRD NORMAL FORM

A table is said to be in the Third Normal Form when,

It is in the Second Normal form.

And, it doesn't have Transitive Dependency

A relation is in third normal form if it holds at least one of the following conditions for every non-trivial functional dependency X → Y.

X is a super key.

Y is a prime attribute, i.e., each element of Y is part of some candidate key.

Hence,all the tables satisfy the third normal form.

CODE (ACTION LISTENERS)

Purchase action listener (Customer):

```
 purchase.addActionListener(new ActionListener()
{
   public void actionPerformed(ActionEvent ae)
   {
      if (price != 0)
      {
         int y =0;
         String column[] = {"Book Name", "Author", "Price", "Qty"};
         String sel[] = {" ", " ", " ", " "};
         tableModel.insertRow(ind, sel);
         ind++;
         tableModel.insertRow(ind, sel);
```

```
                ind++;
                String sel1[] = {" ", " ", "Bill Amount:", " "};
                sel1[3] = String.valueOf(price);
                tableModel.insertRow(ind, sel1);
                ind++;
                String[][] ss = new String[ind][4];
                for (int i = 0; i < ind; i++) {
                   for (int j = 0; j < 4; j++) {
                      ss[i][j] = String.valueOf(jt2.getModel().getValueAt(i, j));
                   }}
                try
                {
                   Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/BookStore","root","srma6997");
                   for(int i =0;i <ind - 3;i++)
                   {
                      int k=0;
                      for(int j=0;j<20;j++)
                      {
                         if(ret[j]==i)
                            k = j;
                      }
                         PreparedStatement st1 = con.prepareStatement("Update BookStore.Book_list set
Quantity = ? where Book_name = ?");
                         st1.setInt(1, quantity[k]);
                         st1.setString(2, ss[i][0]);
                         st1.executeUpdate();
                   }
                   con.close();
                }
                catch(Exception e) {
                   e.printStackTrace();
                }
                try{
                   Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/BookStore","root","srma6997");
                   Statement st1 = con.createStatement();
                   ResultSet rs= st1.executeQuery("SELECT count(*) as c FROM BookStore.Purchase_details");
                   rs.next();
                   String s1 = "Insert into BookStore.Purchase_details values (?,?,?)";
                   PreparedStatement st=con.prepareStatement(s1);
                   y = rs.getInt("c")+1;
                   st.setInt(1,y);
```

```
                st.setInt(2,cusids);
                st.setInt(3,price);
                st.executeUpdate();
                for(int z =0;z<20;z++) {
                   if(arr[z]>0) {
                      String s2 = "Insert into BookStore.Purchase_Books values (?,?,?)";
                      PreparedStatement st2 = con.prepareStatement(s2);
                      st2.setInt(1, y);
                      st2.setInt(2, bookids[z]);
                      st2.setInt(3,arr[z]);
                      st2.executeUpdate();
                   }}
                con.close();
             }
             catch(Exception e){
                e.printStackTrace();
             }
          try {
             Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/BookStore","root","srma6997");
             String s = "SELECT * FROM BookStore.Book_list order by Book_id";
             Statement st = con.createStatement();
             ResultSet rs = st.executeQuery(s);
             int i =0;
             while (rs.next()) {
                quantity[i] = rs.getInt("Quantity");
                bookids[i] = rs.getInt("Book_id");
                i++;
             }
             con.close();
          }
          catch(Exception e)
          {
             e.printStackTrace();
          }
          jt1.getSelectionModel().clearSelection();
       }});
```

Submit action listener (Admin):

```
submit.addActionListener(new ActionListener()
      {
```

```
        public void actionPerformed(ActionEvent ae)
        {
           try{
              Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/BookStore","root","srma6997");
              for(int j=0;j<enin;j++)
              {
                 String s1 = "SELECT * FROM BookStore.Book_list where Book_name = ? order by
Book_id";
                 PreparedStatement st=con.prepareStatement(s1);
                 st.setString(1,entries[j][0]);
                 ResultSet rs = st.executeQuery();
                 if(!rs.next())
                 {
                    int x =0;
                    String s = "SELECT * FROM BookStore.Book_list order by Book_id";
                    Statement st2 = con.createStatement();
                    ResultSet rs2 = st2.executeQuery(s);
                    while(rs2.next())
                       x  = rs2.getInt("Book_id");
                    PreparedStatement st1=con.prepareStatement("Insert into BookStore.Book_list
values(?,?,?,?,?,?)");
                    st1.setInt(1,x+1);
                    st1.setInt(2,adminids);
                    st1.setString(3,entries[j][0]);
                    st1.setString(4,entries[j][1]);
                    st1.setInt(5,Integer.parseInt(entries[j][2]));
                    st1.setInt(6,Integer.parseInt(entries[j][3]));
                    st1.executeUpdate();
                 }
                 else
                 {
                  PreparedStatement st1=con.prepareStatement("Update BookStore.Book_list set Quantity = ?
where Book_name = ?");
                 PreparedStatement st2=con.prepareStatement("Select * from BookStore.Book_list where
        Book_name = ?");
st2.setString(1,entries[j][0]);
                 ResultSet rslt = st2.executeQuery();
                 rslt.next();
                 int d = Integer.parseInt(entries[j][3]) + rslt.getInt("Quantity");
                 st1.setInt(1,d);
                 st1.setString(2,entries[j][0]);
                 st1.executeUpdate();
```

```
                }}
            con.close();
            JOptionPane.showMessageDialog(f5, "Books added successfully", "Alert",
JOptionPane.INFORMATION_MESSAGE);
        }
        catch(Exception e)
        {
            e.printStackTrace();
        }
 }});
```

Reset action listener (Admin):

```
    breset.addActionListener(new ActionListener()
    {
        public void actionPerformed(ActionEvent ae)
        {
            tableModel1.setRowCount(0);
            index1 = 0;
            enin = 0;
            tableModel1.insertRow(index1, rowss);
            index1++;
            try {
                Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/BookStore","root","srma6997");
                String s = "SELECT * FROM BookStore.Book_list order by Book_id";
                Statement st = con.createStatement();
                ResultSet rs = st.executeQuery(s);
                while (rs.next()) {
                    String[] details = new String[4];
                    details[0] = rs.getString("Book_Name");
                    details[1] = rs.getString("Author");
                    details[2] = String.valueOf(rs.getInt("Book_Price"));
                    details[3] = String.valueOf(rs.getString("Quantity"));
                    tableModel1.insertRow(index1, details);
                    index1++;
                }
                con.close();
            }
            catch(Exception e)
            {
                e.printStackTrace();
            }
```

```
        } });


Search page action listeners:

      rb71.addItemListener(new ItemListener(){
         public void itemStateChanged(ItemEvent e){
            try
             {
               Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/BookStore","root","srma6997");
               Statement st = con.createStatement();
               ResultSet rs = st.executeQuery("Select * from BookStore.Book_list where Book_price = (Select
min(book_price) from BookStore.Book_list)");
               rs.next();
               String sbn = "Book Name: " + rs.getString("Book_Name");
               String sbp = "Book Price: " + rs.getInt("Book_price");
               String set = sbn + "\n" + sbp;
               a71.setText(set);
               t71.setText("");
               t72.setText("");
               rb75.setSelected(true);
            }
            catch(Exception ec)
            {
               ec.printStackTrace();
            }} });
      rb72.addItemListener(new ItemListener(){
         public void itemStateChanged(ItemEvent e){
            try
            {
               Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/BookStore","root","srma6997");
               Statement st = con.createStatement();
               ResultSet rs = st.executeQuery("Select * from BookStore.Book_list where Book_price = (Select
max(book_price) from BookStore.Book_list)");
               rs.next();
               String sbn = " Book Name: " + rs.getString("Book_Name");
               String sbp = " Book Price: " + rs.getInt("Book_price");
               String set = sbn + "\n" + sbp;
               a71.setText(set);
               t71.setText("");
               t72.setText("");
```

```
            rb75.setSelected(true);
          }
          catch(Exception ec)
          {
            ec.printStackTrace();
          }}});
      rb73.addItemListener(new ItemListener(){
        public void itemStateChanged(ItemEvent e) {
          rb75.setSelected(true);
          if (t71.getText().equals(""))
            JOptionPane.showMessageDialog(f7, "Enter required value", "Alert",
JOptionPane.INFORMATION_MESSAGE);
          else {
            try
            {
              Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/BookStore",
"root", "srma6997");
              PreparedStatement st = con.prepareStatement("Select * from BookStore.Book_list where
Book_Name like ? ");
              st.setString(1, t71.getText());
              ResultSet rs = st.executeQuery();
              String set = " ";
              while (rs.next())
                set += rs.getString("Book_Name") + " - Rs. "+ rs.getString("Book_Price") + "\n " ;
              a71.setText(set);
              t72.setText("");
            } catch (Exception ec) {
              ec.printStackTrace();
            }
          }}});
      rb74.addItemListener(new ItemListener(){
        public void itemStateChanged(ItemEvent e) {
          rb75.setSelected(true);
          if (t72.getText().equals(""))
            JOptionPane.showMessageDialog(f7, "Enter bill id ", "Alert",
JOptionPane.INFORMATION_MESSAGE);
          else {
            try {
              Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/BookStore",
"root", "srma6997");
              PreparedStatement st = con.prepareStatement("Select Book_Name,b1.Quantity from
BookStore.Book_list b inner join BookStore.Purchase_Books b1 on Book_id = bookid where p_id = ?");
              st.setInt(1, Integer.parseInt(t72.getText()));
```

```
                ResultSet rs = st.executeQuery();
                String set = " ";
                while (rs.next()) {
                   set += rs.getString("Book_Name") + " - " + rs.getString("Quantity") +"\n ";
                }
                PreparedStatement st1 = con.prepareStatement("Select * from  BookStore.Purchase_details
where pid = ?");
                st1.setInt(1, Integer.parseInt(t72.getText()));
                ResultSet rs1 = st1.executeQuery();
                rs1.next();
                set += "Bill Amount: " + rs1.getInt("Bill");
                a71.setText(set);
                t71.setText("");
             } catch (Exception ec) {
                ec.printStackTrace();
             }
     }} });
```

Login action listener:

```
login1.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent ae) {
           //System.out.println(unt.getText() + String.valueOf(pwt.getPassword()));
           if (ch == "")
           {
              JOptionPane.showMessageDialog(f1, "Please select the type of user", "Alert",
JOptionPane.INFORMATION_MESSAGE);
           }
           else if (ch.equals("Admin"))
           {
              try
              {
                 int flag=0;
                 Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/BookStore","root","srma6997");
                 String s = "SELECT * FROM BookStore.AdminLogs inner join BookStore.Admin_Contact
on Adminid=Admin_id";
                 Statement st = con.createStatement();
                 ResultSet rs = st.executeQuery(s);
                 while (rs.next())
                 {
                    String u = rs.getString("Username");
                    String p = rs.getString("Password");
```

```
                    if(unt.getText().equals(u) && String.valueOf(pwt.getPassword()).equals(p))
                    {
                        contacttxt = rs.getString("Contact");
                        nametxt = rs.getString("Name");
                        adminids = rs.getInt("Admin_id");
                        t51.setText(nametxt);
                        t52.setText(contacttxt);
                        flag=1;
                    }
                }
                if(flag==1)
                {
                    f1.setVisible(false);
                    f5.setSize(1000, 800);
                    f5.getContentPane().setBackground(color);
                    f5.setLayout(null);
                    f5.setVisible(true);
                }
                else
                {
                    JOptionPane.showMessageDialog(f1, "Invalid UserName and Password", "Alert",
JOptionPane.INFORMATION_MESSAGE);
                }
                con.close();
            }
            catch(Exception e)
            {
                e.printStackTrace();
            }
            try {
                tableModel1.insertRow(index1, rowss);
                index1++;
                Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/BookStore","root","srma6997");
                String s = "SELECT * FROM BookStore.Book_list order by Book_id";
                Statement st = con.createStatement();
                ResultSet rs = st.executeQuery(s);
                while (rs.next()) {
                    String[] details = new String[4];
                    details[0] = rs.getString("Book_Name");
                    details[1] = rs.getString("Author");
                    details[2] = String.valueOf(rs.getInt("Book_Price"));
                    details[3] = String.valueOf(rs.getString("Quantity"));
```

```
                    tableModel1.insertRow(index1, details);
                    index1++;
                }
                con.close();
            }
            catch(Exception e)
            {
                e.printStackTrace();
            }
        }
        else if (ch.equals("Customer"))
        {
            try {
                int flag=0;
                Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/BookStore","root","srma6997");
                String s = "CREATE OR REPLACE VIEW Cus_details AS SELECT
Cus_id,Name,Address,Username,Password,Contact FROM BookStore.CustomerLogs inner join
BookStore.Customer_Contact on Cus_id = cusid";
                Statement st = con.createStatement();
                st.executeUpdate(s);
                String s1 = "Select * from Cus_details";
                Statement st1 = con.createStatement();
                ResultSet rs = st1.executeQuery(s1);
                while (rs.next())
                {
                    String u = rs.getString("Username");
                    String p = rs.getString("Password");
                    if(unt.getText().equals(u) && String.valueOf(pwt.getPassword()).equals(p))
                    {
                        contacttxt = rs.getString("Contact");
                        nametxt = rs.getString("Name");
                        addresstxt = rs.getString("Address");
                        namet1.setText(nametxt);
                        contactt1.setText(contacttxt);
                        at1.setText(addresstxt);
                        cusids = rs.getInt("Cus_id");
                        flag=1;
                    }
                }
                if(flag==1)
                {
                    f1.setVisible(false);
```

```
                    f3.setLayout(null);
                    f3.setSize(1000, 800);
                    f3.getContentPane().setBackground(lightblue);
                    f3.setVisible(true);
                }
                else
                {
                    JOptionPane.showMessageDialog(f1, "Invalid UserName and Password", "Alert",
JOptionPane.INFORMATION_MESSAGE);
                }
                con.close();
            }
            catch(Exception e)
            {
                e.printStackTrace();
            }
            try {
                price =0;
                tableModel.setRowCount(0);
                ind = 0;
                tableModel2.setRowCount(0);
                indi = 0;
                pt.setText("0");
                for (int i = 0; i < 20; i++)
                {
                    ret[i] = -1;
                    arr[i] = 0;
                }
                Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/BookStore","root","srma6997");
                String s = "SELECT * FROM BookStore.Book_list order by Book_id";
                Statement st = con.createStatement();
                ResultSet rs = st.executeQuery(s);
                int y =0;
                while (rs.next()) {
                    String[] details = new String[4];
                    details[0] = rs.getString("Book_Name");
                    details[1] = rs.getString("Author");
                    details[2] = String.valueOf(rs.getInt("Book_Price"));
                    details[3] = String.valueOf(rs.getInt("Quantity"));
                    tableModel2.insertRow(indi, details);
                    quantity[indi] = Integer.parseInt(details[3]);
                    p[indi] = Integer.parseInt(details[2]);
```

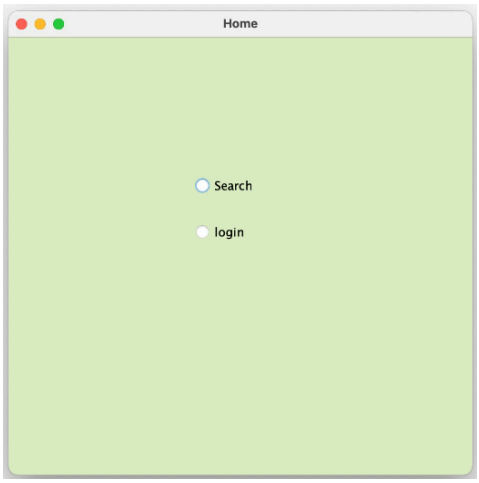
```
                    indi++;
                    bookids[y] = rs.getInt("Book_id");
                    y++;
                }
                con.close();
            }
            catch(Exception e)
            {
                e.printStackTrace();
            }
    }}});
```

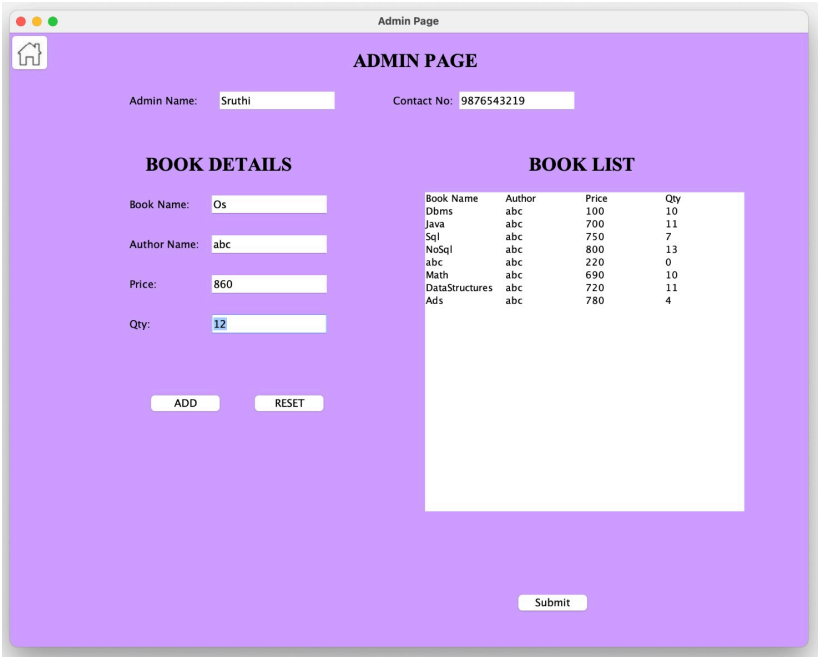
Signup Action Listener:

```
signup2.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent ae) {
            String s = "";
            if (unt1.getText().equals(s) || String.valueOf(pwt1.getPassword()).equals(s) || ct.getText().equals(s) ||
namet.getText().equals(s) || at.getText().equals(s)) {
                JOptionPane.showMessageDialog(f2, "Please fill the required values", "Alert",
JOptionPane.INFORMATION_MESSAGE);
            }
           else {
                try {
                    Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/BookStore","root","srma6997");
                    Statement st1 = con.createStatement();
                    ResultSet rs= st1.executeQuery("SELECT count(*) as c FROM BookStore.CustomerLogs");
                    rs.next();
                    String s1 = "Insert into BookStore.CustomerLogs values (?,?,?,?,?)";
                    PreparedStatement st=con.prepareStatement(s1);
                    st.setInt(1,rs.getInt("c")+1);
                    st.setString(2,namet.getText());
                    st.setString(3,at.getText());
                    st.setString(4,unt1.getText());
                    st.setString(5,String.valueOf(pwt1.getPassword()));
                    st.executeUpdate();
                    String s2 = "Insert into BookStore.Customer_Contact values (?,?)";
                    PreparedStatement st2=con.prepareStatement(s2);
                    st2.setInt(1,rs.getInt("c")+1);
                    st2.setString(2,ct.getText());
                    st2.executeUpdate();
```

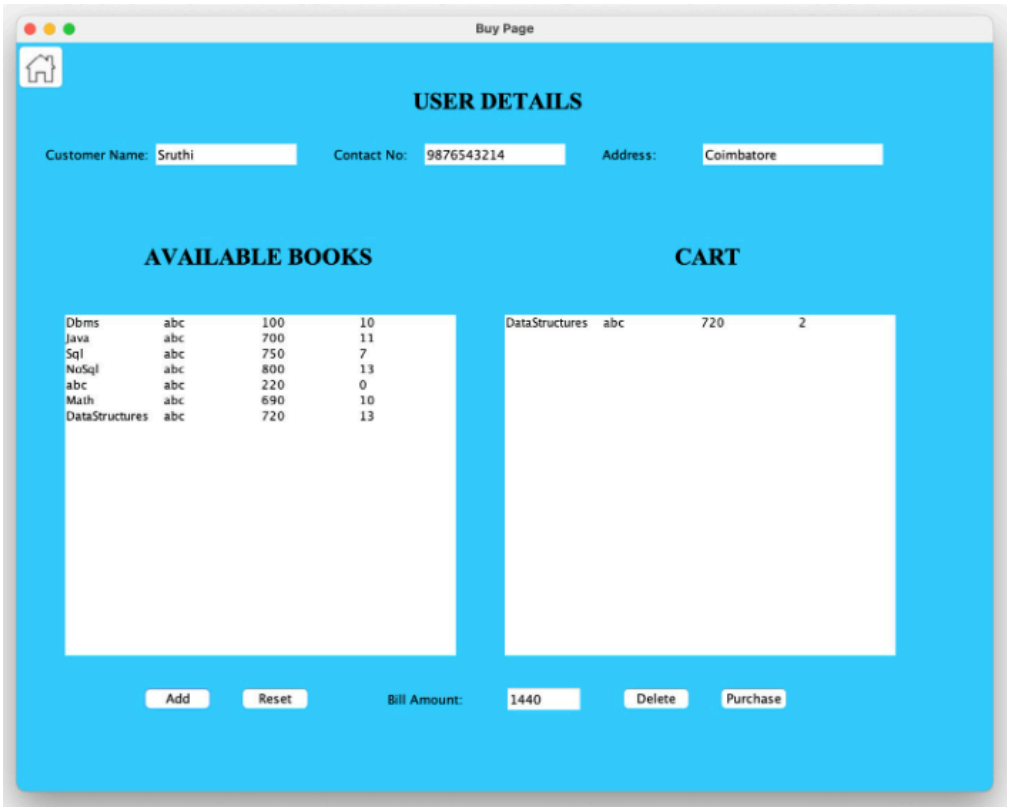
```
                    JOptionPane.showMessageDialog(f2, "Signup Successful", "Alert",
JOptionPane.INFORMATION_MESSAGE);
                    con.close();
                }
                catch(Exception e)
                {
                    e.printStackTrace();
                }
        }
    }
});
```

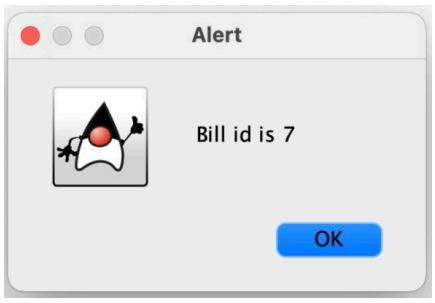
SCREENSHOT OF OUTPUT



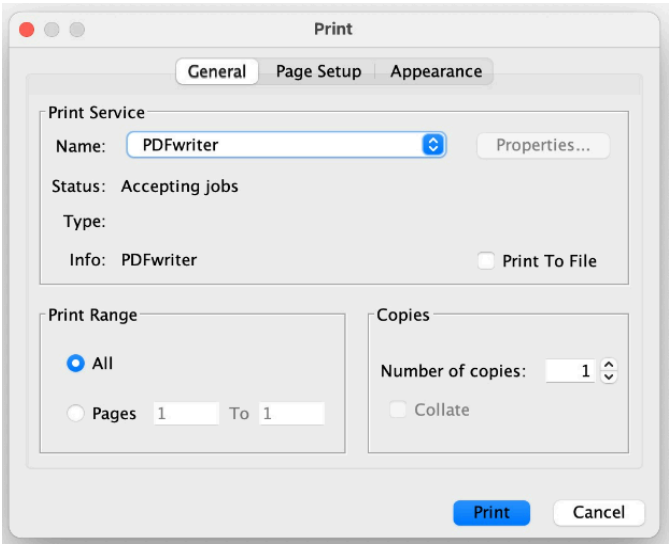


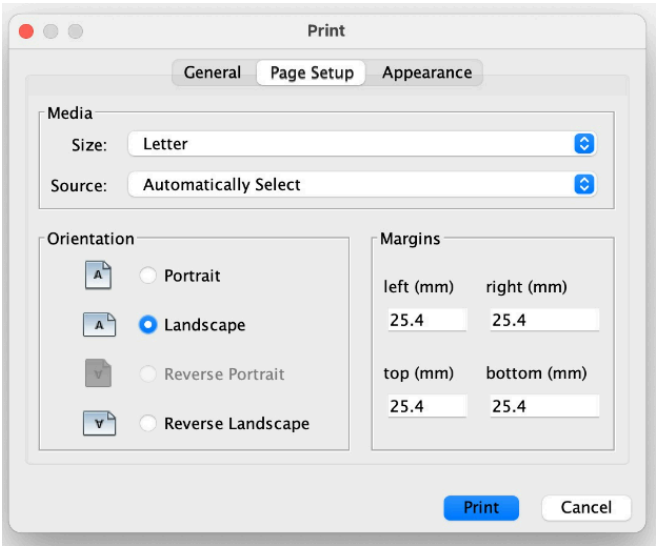
Book_id	Admin_id	Book_Name	Author	Book_Price	Quantity
101	1	Dbms	abc	100	10
102	1	Java	abc	700	11
103	2	Sql	abc	750	7
104	1	NoSql	abc	800	13
105	1	abc	abc	220	0
106	1	Math	abc	690	10
107	1	DataStructures	abc	720	11
108	2	Ads	abc	780	4
109	1	Os	abc	860	12
NULL	NULL	NULL	NULL	NULL	NULL

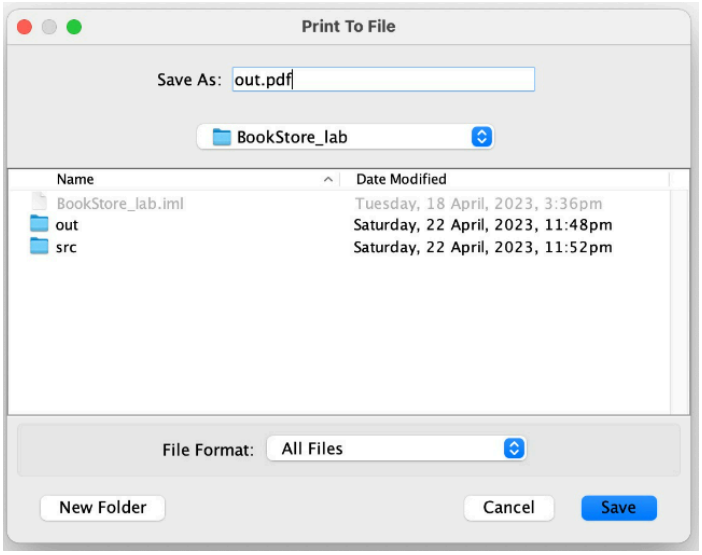



Alert
Bill id is 7
OK


Bill
DataStructures
abc
720
2
Bill Amount:
1440
Print

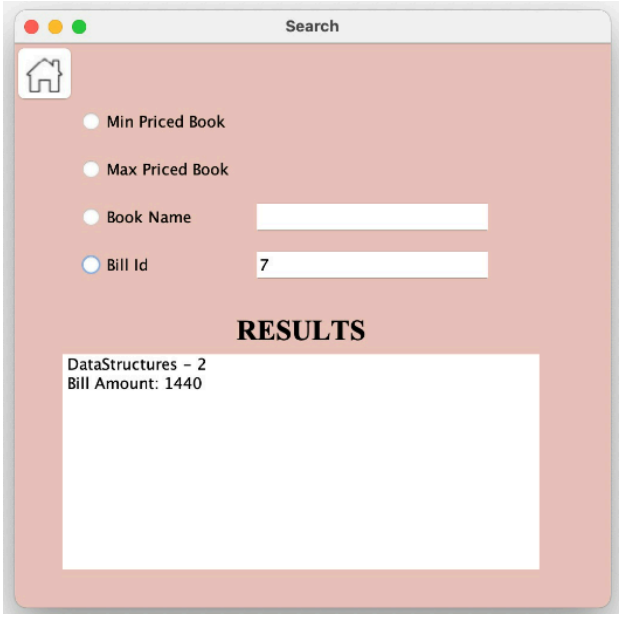

Print
General
Page Setup
Appearance
Print Service
Name:
PDFwriter
Properties...
Status:
Accepting jobs
Type:
Info:
PDFwriter
Print To File
Print Range
All
Pages
1
To
1
Copies
Number of copies:
1
Collate
Print
Cancel


Print
General
Page Setup
Appearance
Media
Size:
Letter
Source:
Automatically Select
Orientation
Portrait
Landscape
Reverse Portrait
Reverse Landscape
Margins
left (mm)
right (mm)
25.4
25.4
top (mm)
bottom (mm)
25.4
25.4
Print
Cancel



Bill

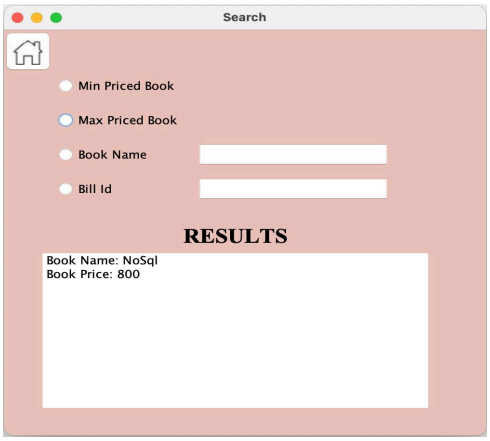
DataStructures	abc	720	2
		Bill Amount:	1440



SCREENSHOT OF THE OUTPUT FOR SQL OPERATION

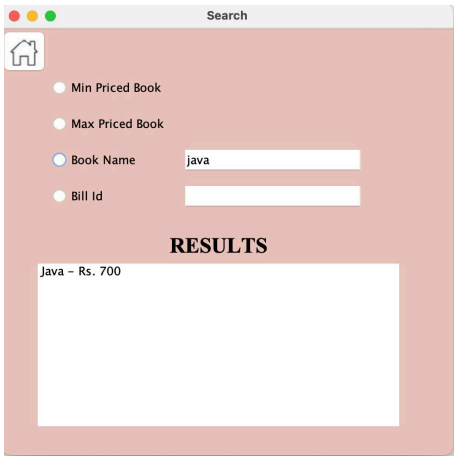
MAX AND SUBQUERY

```
Connection con = DriverManager.getConnection( url: "jdbc:mysql://localhost:3306/BookStore", user: "root", password: "srma6997");
Statement st = con.createStatement();
ResultSet rs = st.executeQuery( sql: "Select * from BookStore.Book_list where Book_price = (Select max(book_price) from BookStore.Book_list)");
rs.next();
String sbn = " Book Name: " + rs.getString( columnLabel: "Book_Name");
String sbp = " Book Price: " + rs.getInt( columnLabel: "Book_price");
String set = sbn + "\n" + sbp;
a71.setText(set);
t71.setText("");
t72.setText("");
rb75.setSelected(true);
```



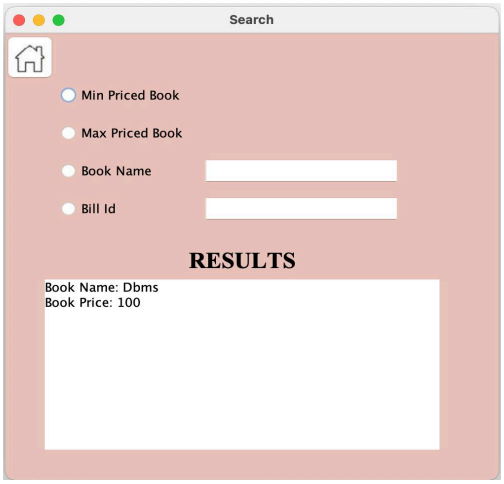
LIKE

```
Connection con = DriverManager.getConnection( url: "jdbc:mysql://localhost:3306/BookStore", user: "root", password: "srma6997");
PreparedStatement st = con.prepareStatement( sql: "Select * from BookStore.Book_list where Book_Name like ? ");
st.setString( parameterIndex: 1, t71.getText());
ResultSet rs = st.executeQuery();
String set = " ";
while (rs.next())
    set += rs.getString( columnLabel: "Book_Name") + " - Rs. "+ rs.getString( columnLabel: "Book_Price") + "\n " ;
a71.setText(set);
t72.setText("");
con.close();
```



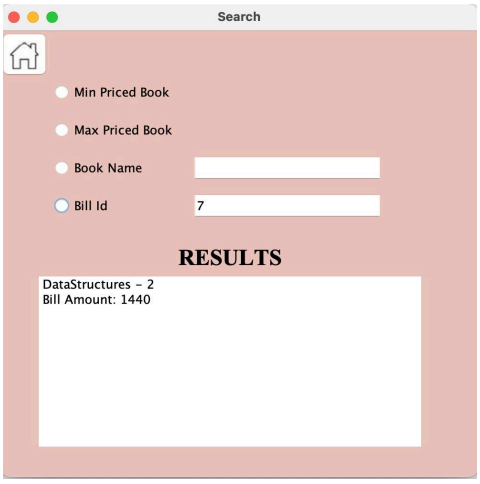
MIN AND SUBQUERY

```
Connection con = DriverManager.getConnection( url: "jdbc:mysql://localhost:3306/BookStore", user: "root", password: "srma6997");
Statement st = con.createStatement();
ResultSet rs = st.executeQuery( sql: "Select * from BookStore.Book_list where Book_price = (Select min(book_price) from BookStore.Book_list)");
rs.next();
String sbn = "Book Name: " + rs.getString( columnLabel: "Book_Name");
String sbp = "Book Price: " + rs.getInt( columnLabel: "Book_price");
String set = sbn + "\n" + sbp;
a71.setText(set);
t71.setText("");
t72.setText("");
rb75.setSelected(true);
```



INNER JOIN

```
Connection con = DriverManager.getConnection( url: "jdbc:mysql://localhost:3306/BookStore", user: "root", password: "srma6997");
PreparedStatement st = con.prepareStatement( sql: "Select Book_Name,b1.Quantity from BookStore.Book_list b inner join BookStore.Purchase_Books b1 on Book_id = bookid where p_id = ?");
st.setInt( parameterIndex: 1, Integer.parseInt(t72.getText()));
ResultSet rs = st.executeQuery();
String set = " ";
while (rs.next()) {
    set += rs.getString( columnLabel: "Book_Name") + " - " + rs.getString( columnLabel: "Quantity") +"\n ";
}
PreparedStatement st1 = con.prepareStatement( sql: "Select * from  BookStore.Purchase_details where pid = ?");
st1.setInt( parameterIndex: 1, Integer.parseInt(t72.getText()));
ResultSet rs1 = st1.executeQuery();
rs1.next();
set += "Bill Amount: " + rs1.getInt( columnLabel: "Bill");
a71.setText(set);
t71.setText("");
con.close();
```



INDEX

```
1 ●  create index index1 on BookStore.CustomerLogs( Username);
```

Action Output

	Time	Action	Response	Duration / Fetch Time
1	14:40:16	create index index1 on BookStore.CustomerLogs(Username)	0 row(s) affected Records: 0 Duplicates: 0 Warnings...	0.035 sec

INSERT STATEMENT

```
PreparedStatement st1=con.prepareStatement( sql: "Insert into BookStore.Book_list values(?,?,?,?,?,?)");
st1.setInt( parameterIndex: 1, x: x+1);
st1.setInt( parameterIndex: 2,adminids);
st1.setString( parameterIndex: 3,entries[j][0]);
st1.setString( parameterIndex: 4,entries[j][1]);
st1.setInt( parameterIndex: 5,Integer.parseInt(entries[j][2]));
st1.setInt( parameterIndex: 6,Integer.parseInt(entries[j][3]));
st1.executeUpdate();
```

UPDATE STATEMENT

```
PreparedStatement st1=con.prepareStatement( sql: "Update BookStore.Book_list set Quantity = ? and Admin_id = ? where Book_name = ?");
st1.setInt( parameterIndex: 1,Integer.parseInt(entries[j][3]));
st1.setInt( parameterIndex: 2,adminids);
st1.setString( parameterIndex: 3,entries[j][0]);
st1.executeUpdate();
```

COUNT

```
SELECT count(*) as c FROM BookStore.CustomerLogs;
```

c
4

```
signup2.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent ae) {
        String s = "";
        if (unt1.getText().equals(s) || String.valueOf(pwt1.getPassword()).equals(s) || ct.getText().equals(s) || namet.getText().equals(s) || at.getText().equals(s)) {
            JOptionPane.showMessageDialog(f2,  message: "Please fill the required values",  title: "Alert", JOptionPane.INFORMATION_MESSAGE);
        } else {
            try {
                Connection con = DriverManager.getConnection( url: "jdbc:mysql://localhost:3306/BookStore", user: "root", password: "srma6997");
                Statement st1 = con.createStatement();
                ResultSet rs= st1.executeQuery( sql: "SELECT count(*) as c FROM BookStore.CustomerLogs");
                rs.next();
                String s1 = "Insert into BookStore.CustomerLogs values (?,?,?,?,?)";
                PreparedStatement st=con.prepareStatement(s1);
                st.setInt( parameterIndex: 1, x: rs.getInt( columnLabel: "c")+1);
                st.setString( parameterIndex: 2,namet.getText());
                st.setString( parameterIndex: 3,at.getText());
                st.setString( parameterIndex: 4,unt1.getText());
                st.setString( parameterIndex: 5,String.valueOf(pwt1.getPassword()));
                st.executeUpdate();
                String s2 = "Insert into BookStore.Customer_Contact values (?,?)";
                PreparedStatement st2=con.prepareStatement(s2);
                st2.setInt( parameterIndex: 1, x: rs.getInt( columnLabel: "c")+1);
                st2.setString( parameterIndex: 2,ct.getText());
                st2.executeUpdate();
                JOptionPane.showMessageDialog(f2,  message: "Signup Successful",  title: "Alert", JOptionPane.INFORMATION_MESSAGE);
                f2.setVisible(false);
                f1.setLayout(null);
                f1.setSize( width: 500,  height: 500);
                f1.getContentPane().setBackground(Color.PINK);
                f1.setVisible(true);
                con.close();
                ch = "";
                unt.setText("");
                pwt.setText("");
                namet.setText("");
                ct.setText("");
                at.setText("");
                unt1.setText("");
                pwt1.setText("");
            }
            catch(Exception e)
            {
                e.printStackTrace();
            }
```

VIEW

```
CREATE OR REPLACE VIEW Cus_details AS SELECT Cus_id,Name,Address,Username,Password,Contact FROM BookStore.CustomerLogs inner join BookStore.Customer_Contact on Cus_id = cusid;
Select * from Cus_details;
```

```
int flag=0;
Connection con = DriverManager.getConnection( url: "jdbc:mysql://localhost:3306/BookStore", user: "root", password: "srma6997");
String s = "CREATE OR REPLACE VIEW Cus_details AS SELECT Cus_id,Name,Address,Username,Password,Contact FROM BookStore.CustomerLogs inner join BookStore.Customer_Contact on Cus_id = cusid";
Statement st = con.createStatement();
st.executeUpdate(s);
String s1 = "Select * from Cus_details";
Statement st1 = con.createStatement();
ResultSet rs = st1.executeQuery(s1);
while (rs.next())
{
    String u = rs.getString( columnLabel: "Username");
    String p = rs.getString( columnLabel: "Password");
    if(unt.getText().equals(u) && String.valueOf(pwt.getPassword()).equals(p))
    {
        contacttxt = rs.getString( columnLabel: "Contact");
        nametxt = rs.getString( columnLabel: "Name");
        addresstxt = rs.getString( columnLabel: "Address");
        namet1.setText(nametxt);
        contactt1.setText(contacttxt);
        at1.setText(addresstxt);
        cusids = rs.getInt( columnLabel: "Cus_id");
        flag=1;
    }
}
if(flag==1)
{
    f1.setVisible(false);
    f3.setLayout(null);
    f3.setSize( width: 1000, height: 800);
    f3.getContentPane().setBackground(lightblue);
    f3.setVisible(true);
}
else
{
    JOptionPane.showMessageDialog(f1, message: "Invalid UserName and Password", title: "Alert", JOptionPane.INFORMATION_MESSAGE);
}
con.close();
```

Cus_id	Name	Address	Username	Password	Contact
1	Sruthi	Coimbatore	sruthi	123	9876543214
2	Karthika	Salem	karthika	123	9876543210
3	Kavya	Kumbakonam	kavya	123	9876543217
4	Divya	Coimbatore	divya	123	8765432106