

**JAVA SWING BASED –IT returns processing
database**

SQL CONNECTIVITY USING JDBC

A

Report

*Submitted in partial fulfillment of the
Requirements for the award of the Degree of*

**BACHELOR OF ENGINEERING
IN
INFORMATION TECHNOLOGY**

By

Talveda Shreya<1602-20-737-041>

Under the guidance of Ms B. Leelavathy



**Department of Information Technology
Vasavi College of Engineering (Autonomous)
(Affiliated to Osmania University)
Ibrahimbagh, Hyderabad-31**

2020-2021

BONAFIDE CERTIFICATE

This is to certify that this project report titled
‘IT Returns Processing Database’
is a project work of **Talveda Shreya** bearing
roll no. 1602-20-737-041 who carried out this
project under my supervision in the IV semester
for the academic year 2021- 2022

Signature

External Examiner

Signature

Internal Examiner

AIM AND PRIORITY OF THE PROJECT

To create a **Java GUI-based** desktop application that connects students looking for internships with project managers looking for interns. It takes values like account number, name, contact details which are then updated in the database using JDBC connectivity.

ARCHITECTURE AND TECHNOLOGY

Software used:

Java Eclipse, Oracle 11g Database, Java SE version 13, SQL*Plus.

Java SWING:

Java SWING is a GUI widget toolkit for Java. It is part of Oracle's Java Foundation Classes (JFC) - an API for providing a graphical user interface (GUI) for Java programs.

Swing was developed to provide a more sophisticated set of GUI components than the earlier AWT. Swing provides a look and feel that emulates the look and feel of several platforms, and also supports a pluggable look and feel that allows applications to have a look and feel unrelated to the underlying platform. It has more powerful and flexible components than AWT. In addition to familiar components such as buttons, check boxes and labels, Swing provides several advanced components such as tabbed panel, scroll panes, trees, tables, and lists.

SQL:

Structure Query Language(SQL) is a database query language used for storing and managing data in **Relational** DBMS. SQL was the first commercial language introduced for E.F Codd's Relational model of database. Today almost all RDBMS (MySQL, Oracle, Infomix, Sybase, MS Access) use **SQL** as the standard database query language. SQL is used to perform all types of data operations in RDBMS.

ABSTRACT

IT returns processing database is used to represent the system of income tax payment. The database processes when the details are to be updated and how they are to be filed. This system has accountant details and the bank details of the accountants which include annual income based on which tax is calculated. This tax is validated to payment. Taxpayers should file their IT returns by the due date to avoid fine.

Requirement analysis:

List of tables:

- Account_holder
- Bank_details
- Payment
- Issues
- Contains

List of Attributes with their Domain Types:

Accountholder:

- | | |
|--------------------|--------------|
| ➤ accountnumber | Number(20) |
| ➤ name | varchar2(20) |
| ➤ contact__details | varchar2(40) |
| ➤ profession | varchar2(20) |
| ➤ Aid | Number |

Bankdetails:

- | | |
|------------------|--------------|
| ➤ Account_number | Number(20) |
| ➤ Annual_income | Number |
| ➤ Address | varchar2(40) |
| ➤ Tax | Number |
| ➤ utr | Number |

Payment:

- | | |
|--------------|--------|
| ➤ Payment_id | Number |
| ➤ Tax | Number |

- Due_date varchar2(10)
- Fine Number

Contain:

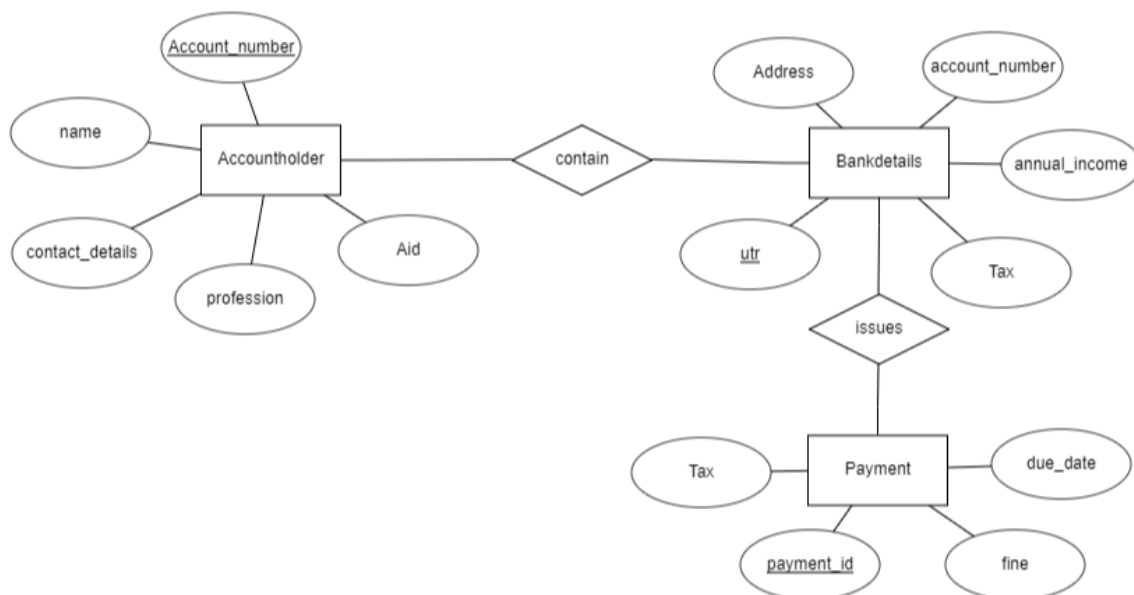
- Account_number Number (20)
- utr Number

Issues:

- Payment_id Number
- utr Number

DESIGN

ENTITY RELATIONSHIP DIAGRAM



DDL OPERATIONS:

```

SQL> create table accountholder(
account_number number,
name varchar2(20),
contact_details varchar2(30),
  
```

profession varchar2(20),
Aid number,
Primary key(account_number));

Table created.

```
SQL> create table accountholder(  
  2  account_number number,  
  3  name varchar2(20),  
  4  contact_details varchar2(20),  
  5  profession varchar2(20),  
  6  Aid number,  
  7  primary key(account_number));
```

Table created.

```
SQL> desc accountholder
```

Name	Null?	Type
ACCOUNT_NUMBER	NOT NULL	NUMBER
NAME		VARCHAR2(20)
CONTACT_DETAILS		VARCHAR2(20)
PROFESSION		VARCHAR2(20)
AID		NUMBER

```
SQL> create table bankdetails (  
  2  utr number,  
  3  account_number number,  
  4  annual_income number,  
  5  address varchar2(40),  
  6  tax number,  
  7  primary key(utr),  
  8  foreign key(account_number) references accountholder);
```

Table created.

```
SQL> create table bankdetails(
  2   utr number,
  3   account_number number,
  4   annual_income number,
  5   address varchar2(40),
  6   tax number,
  7   primary key(utr),
  8   foreign key(account_number) references accountholder);

Table created.
```

```
SQL> desc bankdetails
```

Name	Null?	Type
UTR	NOT NULL	NUMBER
ACCOUNT_NUMBER		NUMBER
ANNUAL_INCOME		NUMBER
ADDRESS		VARCHAR2(40)
TAX		NUMBER

```
SQL> create table payment (
  2   payment_id number,
  3   tax number,
  4   due_date varchar2(20),
  5   fine number,
  6   primary key(payment_id));
```

Table created.

```
SQL> desc payment
```

Name	Null?	Type
PAYMENT_ID	NOT NULL	NUMBER
TAX		NUMBER
DUE_DATE		VARCHAR2(20)
FINE		NUMBER

```
SQL> create table contain (
  2   account_number number,
  3   utr number,
  4   foreign key(account_number) references accountholder,
  5   foreign key(utr) references bankdetails);
```


Table created.

```
SQL> create table contain(  
2  account_number number,  
3  utr number,  
4  foreign key(account_number) references accountholder,  
5  foreign key(utr) references bankdetails);
```

Table created.

```
SQL> desc contain
```

Name	Null?	Type
ACCOUNT_NUMBER		NUMBER
UTR		NUMBER

```
SQL> create table issues (  
2  payment_id number,  
3  utr number,  
4  foreign key(payment_id) references payment,  
5  foreign key(utr) references bankdetails);
```

Table created.

```
SQL> create table issues(  
2  payment_id number,  
3  utr number,  
4  foreign key(payment_id) references payment,  
5  foreign key(utr) references bankdetails);
```

Table created.

```
SQL> desc issues
```

Name	Null?	Type
PAYMENT_ID		NUMBER
UTR		NUMBER

DML OPERATIONS:

```
SQL> insert into accountholder values(&account_number,&'name',&contact_details,&'profession',&Aid);
Enter value for account_number: 456123789012
Enter value for name: shreya
Enter value for contact_details: 9876543210
Enter value for profession: engineer
Enter value for aid: 10
old 1: insert into accountholder values(&account_number,&'name',&contact_details,&'profession',&Aid)
new 1: insert into accountholder values(456123789012,'shreya',9876543210,'engineer',10)

1 row created.
```

```
SQL> insert into accountholder values(&account_number,&'name',&contact_details,&'profession',&Aid);
Enter value for account_number: 523678941023
Enter value for name: john
Enter value for contact_details: 8790321461
Enter value for profession: doctor
Enter value for aid: 12
old 1: insert into accountholder values(&account_number,&'name',&contact_details,&'profession',&Aid)
new 1: insert into accountholder values(523678941023,'john',8790321461,'doctor',12)

1 row created.
```

```
SQL> /
Enter value for account_number: 701359468021
Enter value for name: ram
Enter value for contact_details: 7536984120
Enter value for profession: professor
Enter value for aid: 14
old 1: insert into accountholder values(&account_number,&'name',&contact_details,&'profession',&Aid)
new 1: insert into accountholder values(701359468021,'ram',7536984120,'professor',14)

1 row created.
```

```
SQL> /
Enter value for account_number: 921045638705
Enter value for name: neha
Enter value for contact_details: 6347891023
Enter value for profession: engineer
Enter value for aid: 15
old 1: insert into accountholder values(&account_number,&'name',&contact_details,&'profession',&Aid)
new 1: insert into accountholder values(921045638705,'neha',6347891023,'engineer',15)

1 row created.
```

```
SQL> /
Enter value for account_number: 963478520178
Enter value for name: sai
Enter value for contact_details: 8523690147
Enter value for profession: doctor
Enter value for aid: 16
old 1: insert into accountholder values(&account_number,&'name',&contact_details,&'profession',&Aid)
new 1: insert into accountholder values(963478520178,'sai',8523690147,'doctor',16)

1 row created.
```

```
SQL> select * from accountholder;
```

ACCOUNT_NUMBER	NAME	CONTACT_DETAILS	PROFESSION

AID			

4.5612E+11	shreya 10	9876543210	engineer
5.2368E+11	john 12	8790321461	doctor
7.0136E+11	ram 14	7536984120	professor

ACCOUNT_NUMBER	NAME	CONTACT_DETAILS	PROFESSION

AID			

9.2105E+11	neha 15	6347891023	engineer
9.6348E+11	sai 16	8523690147	doctor

2.

```
SQL> insert into bankdetails values(&utr,&account_number,&annual_income,&'&address',&tax);
Enter value for utr: 1256
Enter value for account_number: 456123789012
Enter value for annual_income: 2100000
Enter value for address: hyderabad
Enter value for tax: 630000
old 1: insert into bankdetails values(&utr,&account_number,&annual_income,&'&address',&tax)
new 1: insert into bankdetails values(1256,456123789012,2100000,'hyderabad',630000)

1 row created.
```

```
SQL> insert into bankdetails values(&utr,&account_number,&annual_income,&'&address',&tax);
Enter value for utr: 5234
Enter value for account_number: 523678941023
Enter value for annual_income: 2500000
Enter value for address: Bangalore
Enter value for tax: 750000
old 1: insert into bankdetails values(&utr,&account_number,&annual_income,&'&address',&tax)
new 1: insert into bankdetails values(5234,523678941023,2500000,'Bangalore',750000)

1 row created.
```

```
SQL> /
Enter value for utr: 8927
Enter value for account_number: 701359468021
Enter value for annual_income: 1900000
Enter value for address: Chennai
Enter value for tax: 570000
old 1: insert into bankdetails values(&utr,&account_number,&annual_income,&'&address',&tax)
new 1: insert into bankdetails values(8927,701359468021,1900000,'Chennai',570000)

1 row created.
```

```
SQL> /
Enter value for utr: 1326
Enter value for account_number: 921045638705
Enter value for annual_income: 2000000
Enter value for address: mumbai
Enter value for tax: 600000
old 1: insert into bankdetails values(&utr,&account_number,&annual_income,&'&address',&tax)
new 1: insert into bankdetails values(1326,921045638705,2000000,'mumbai',600000)

1 row created.
```

```
SQL> /
Enter value for utr: 1567
Enter value for account_number: 963478520178
Enter value for annual_income: 1800000
Enter value for address: kerala
Enter value for tax: 540000
old 1: insert into bankdetails values(&utr,&account_number,&annual_income,&'&address',&tax)
new 1: insert into bankdetails values(1567,963478520178,1800000,'kerala',540000)

1 row created.
```

```
SQL> select * from bankdetails;
```

UTR	ACCOUNT_NUMBER	ANNUAL_INCOME	ADDRESS
1256 630000	4.5612E+11	2100000	hyderabad
5234 750000	5.2368E+11	2500000	Banglore
8927 570000	7.0136E+11	1900000	Chennai

UTR	ACCOUNT_NUMBER	ANNUAL_INCOME	ADDRESS
1326 600000	9.2105E+11	2000000	mumbai
1567 540000	9.6348E+11	1800000	kerala

3.

```
SQL> insert into contain values(&account_number,&utr);
Enter value for account_number: 456123789012
Enter value for utr: 1256
old 1: insert into contain values(&account_number,&utr)
new 1: insert into contain values(456123789012,1256)

1 row created.

SQL> /
Enter value for account_number: 523678941023
Enter value for utr: 5234
old 1: insert into contain values(&account_number,&utr)
new 1: insert into contain values(523678941023,5234)

1 row created.

SQL> /
Enter value for account_number: 701359468021
Enter value for utr: 8927
old 1: insert into contain values(&account_number,&utr)
new 1: insert into contain values(701359468021,8927)

1 row created.

SQL> /
Enter value for account_number: 921045638705
Enter value for utr: 1326
old 1: insert into contain values(&account_number,&utr)
new 1: insert into contain values(921045638705,1326)

1 row created.

SQL> /
Enter value for account_number: 963478520178
Enter value for utr: 1567
old 1: insert into contain values(&account_number,&utr)
new 1: insert into contain values(963478520178,1567)

1 row created.
```

```
SQL> select * from contain;
```

ACCOUNT_NUMBER	UTR
4.5612E+11	1256
5.2368E+11	5234
7.0136E+11	8927
9.2105E+11	1326
9.6348E+11	1567

4.

```
SQL> insert into payment values(&payment_id,&tax,'&due_date',&fine);
Enter value for payment_id: 80256
Enter value for tax: 630000
Enter value for due_date: may 4
Enter value for fine: 1000
old 1: insert into payment values(&payment_id,&tax,'&due_date',&fine)
new 1: insert into payment values(80256,630000,'may 4',1000)

1 row created.
```

```
SQL> insert into payment values(&payment_id,&tax,'&due_date',&fine);
Enter value for payment_id: 89641
Enter value for tax: 750000
Enter value for due_date: may 19
Enter value for fine: 1500
old 1: insert into payment values(&payment_id,&tax,'&due_date',&fine)
new 1: insert into payment values(89641,750000,'may 19',1500)

1 row created.
```

```
SQL> /
Enter value for payment_id: 80369
Enter value for tax: 570000
Enter value for due_date: may 27
Enter value for fine: 1500
old 1: insert into payment values(&payment_id,&tax,'&due_date',&fine)
new 1: insert into payment values(80369,570000,'may 27',1500)

1 row created.
```

```
SQL> /
Enter value for payment_id: 81236
Enter value for tax: 600000
Enter value for due_date: june 2
Enter value for fine: 2000
old 1: insert into payment values(&payment_id,&tax,'&due_date',&fine)
new 1: insert into payment values(81236,600000,'june 2',2000)

1 row created.
```

```
SQL> /
Enter value for payment_id: 91000
Enter value for tax: 540000
Enter value for due_date: june 6
Enter value for fine: 2500
old 1: insert into payment values(&payment_id,&tax,'&due_date',&fine)
new 1: insert into payment values(91000,540000,'june 6',2500)

1 row created.
```

```

1 row created.

SQL> select * from payment;

PAYMENT_ID      TAX DUE_DATE      FINE
-----
      80256      630000 may  4      1000
      80369      570000 may 27      1500
      89641      750000 may 19      1500
      81236      600000 june  2      2000
      91000      540000 june  6      2500

```

5.

```

SQL> insert into issues values(&payment_id,&utr);
Enter value for payment_id: 80256
Enter value for utr: 1256
old  1: insert into issues values(&payment_id,&utr)
new  1: insert into issues values(80256,1256)

1 row created.

SQL> /
Enter value for payment_id: 80369
Enter value for utr: 8927
old  1: insert into issues values(&payment_id,&utr)
new  1: insert into issues values(80369,8927)

1 row created.

SQL> /
Enter value for payment_id: 89641
Enter value for utr: 5234
old  1: insert into issues values(&payment_id,&utr)
new  1: insert into issues values(89641,5234)

1 row created.

SQL> /
Enter value for payment_id: 81236
Enter value for utr: 1326
old  1: insert into issues values(&payment_id,&utr)
new  1: insert into issues values(81236,1326)

1 row created.

SQL> /
Enter value for payment_id: 91000
Enter value for utr: 1567
old  1: insert into issues values(&payment_id,&utr)
new  1: insert into issues values(91000,1567)

1 row created.

```

```

SQL> select * from issues;

PAYMENT_ID      UTR
-----
      80256      1256
      80369      8927
      89641      5234
      81236      1326
      91000      1567

```

IMPLEMENTATION:

Front end programs and its connectivity

Java Database Connectivity (JDBC) is an application programming interface (API) for the programming language Java, which defines how a client may access a database. It is a Java-based data access technology used for Java database connectivity. It is part of the Java Standard Edition platform, from Oracle Corporation. It provides methods to query and update data in a database and is oriented towards relational databases.

The connection to the database can be performed using Java programming (JDBC API) as:

```
public conn(){
    try{
        Class.forName("oracle.jdbc.OracleDriver");
        conn=DriverManager.getConnection("jdbc:oracle:thin:@218.248.0.7:1521:rdbms",
        "it20737041","vasavi");
        System.out.println("Connected");
        s = conn.createStatement();
    }
    catch(Exception e) {
        e.printStackTrace();
    }
}
```

Thus, the connection from Java to Oracle database is performed and therefore, can be used for updating tables in the database directly.

Account_holder:

```
import java.awt.*;

import java.awt.event.*;

import javax.swing.*;

import javax.swing.table.DefaultTableModel;

import java.sql.*;
```

```

import java.util.Properties;

public class Account_holder{

private JPanel pn, pn1, pn2, pn3;

private JFrame jf;

private JButton JB_insert, JB_modify, JB_view, JB_delete;

private JLabel JL_account_number, JL_name, JL_contact_details, JL_profession, JL_aid;

private JTextField JTF_account_number, JTF_name, JTF_contact_details, JTF_profession, JTF_aid;

private JMenuItem insert2, update2, view2, delete2;

private List Accountholders_List;

private Choice account_number;

public Account_holder(JPanel pn, JFrame jf, JMenuItem insert2, JMenuItem update2, JMenuItem
view2, JMenuItem delete2){

try {

Class.forName("oracle.jdbc.driver.OracleDriver");

}

catch (Exception e) {

System.err.println("Unable to find and load driver");

System.exit(1);

}

this.jf = jf;

this.insert2 = insert2;

this.update2 = update2;

this.view2 = view2;

```



```

this.delete2 = delete2;

JL_account_number = new JLabel("Account_number : ");

JTF_account_number = new JTextField(10);

JL_name = new JLabel("name: ");

JTF_name = new JTextField(10);

JL_contact_details = new JLabel("contact_details: ");

JTF_contact_details = new JTextField(10);

JL_profession = new JLabel("profession: ");

JTF_profession = new JTextField(10);

JL_aid = new JLabel("Aid : ");

JTF_aid = new JTextField(10);

this.pn=pn;

}

private void displaySQLExceptions(SQLException e) {

JOptionPane.showMessageDialog(pn1,"\nSQLException: " + e.getMessage() + "\n"+"SQLState: "
+

e.getSQLState() + "\n"+"VendorError: " + e.getErrorCode() + "\n");

}

public void load_details(){

try{

account_number = new Choice();

account_number.removeAll();

```

```

Connection con1 =
DriverManager.getConnection("jdbc:oracle:thin:@218.248.0.7:1521:rdbms","it20737041",
"vasavi");

Statement st1 = con1.createStatement();

ResultSet rs1 = st1.executeQuery("select * from account_holder");

while(rs1.next()) {

account_number.add(rs1.getString("Aid"));

}

}

catch(SQLException e) {

displaySQLErrors(e);

}

}

public void loadaddetails(){

try{

Accountholders_List = new List();

Accountholders_List.removeAll();

Connection con2 =
DriverManager.getConnection("jdbc:oracle:thin:@218.248.0.7:1521:rdbms","it20737041",
"vasavi");

Statement st2 = con2.createStatement();

ResultSet rs2 = st2.executeQuery("select * from account_holder");

while(rs2.next()) {

Accountholders_List.add(rs2.getString("account_number"));

}

```

```
}

catch(SQLException e) {

displaySQLErrors(e);

}

}

public void buildGUI(){

insert2.addActionListener(new ActionListener(){

@Override

public void actionPerformed(ActionEvent aevt){

JB_insert = new JButton("Submit");

JTF_aid.setText(null);

JTF_account_number.setText(null);

JTF_name.setText(null);

JTF_contact_details.setText(null);

JTF_profession.setText(null);

loaddetails();

pn.removeAll();

jf.invalidate();

jf.validate();

jf.repaint();

pn1 = new JPanel();

pn1.setLayout(new GridLayout(10, 10));

pn1.add(JL_account_number);
```

```
pn1.add(JTF_account_number);

pn1.add(JL_name);

pn1.add(JTF_name);

pn1.add(JL_contact_details);

pn1.add(JTF_contact_details);

pn1.add(JL_profession);

pn1.add(JTF_profession);

pn1.add(JL_aid);

pn1.add(JTF_aid);

pn3 = new JPanel(new FlowLayout());

pn3.add(JB_insert);

pn1.setBounds(115, 80, 300, 250);

pn3.setBounds(200, 350, 75, 35);

pn2 = new JPanel(new FlowLayout());

Accountholders_List = new List(10);

loaddetails();

pn2.add(Accountholders_List);

pn2.setBounds(200, 350, 300, 180);

pn.add(pn1);

pn.add(pn3);

pn.add(pn2);

pn.setLayout(new BorderLayout());

jf.add(pn);
```

```

jf.setSize(800, 800);

jf.validate();

JB_insert.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent aevt){

try{

Connection con
=DriverManager.getConnection("jdbc:oracle:thin:@218.248.0.7:1521:rdbms", "it20737041",
"vasavi");

String query = "INSERT INTO account_holder (account_number, name, contact_details,
profession, aid) values(?, ?, ?, ?, ?)";

PreparedStatement stp = con.prepareStatement(query);

stp.setString(1, JTF_account_number.getText());

stp.setString(2, JTF_name.getText());

stp.setString(3, JTF_contact_details.getText());

stp.setString(4, JTF_profession.getText());

stp.setString(5, JTF_aid.getText());

int i = stp.executeUpdate();

con.close();

if(i > 0){

JOptionPane.showMessageDialog(pn, "\nInserted successfully");

}

}

catch(SQLException e) {

```

```
displaySQLErrors(e);

}

}

});

}

});

update2.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent aevt){

JB_modify = new JButton("Modify");

JTF_account_number.setText(null);

JTF_name.setText(null);

JTF_contact_details.setText(null);

JTF_profession.setText(null);

JTF_aid.setText(null);

pn.removeAll();

jf.invalidate();

jf.validate();

jf.repaint();

pn1 = new JPanel();

pn1.setLayout(new GridLayout(10, 10));

pn1.add(JL_account_number);

pn1.add(JTF_account_number);
```

```
pn1.add(JL_name);

pn1.add(JTF_name);

pn1.add(JL_contact_details);

pn1.add(JTF_contact_details);

pn1.add(JL_profession);

pn1.add(JTF_profession);

pn1.add(JL_aid);

pn1.add(JTF_aid);

pn3 = new JPanel(new FlowLayout());

pn3.add(JB_modify);

pn1.setBounds(115, 80, 300, 250);

pn3.setBounds(200, 350, 75, 35);

pn2 =new JPanel(new FlowLayout());

Accountholders_List = new List(10);

loaddetails();

pn2.add(Accountholders_List);

pn2.setBounds(200, 350, 300, 180);

pn.add(pn1);

pn.add(pn3);

pn.add(pn2);

pn.setLayout(new BorderLayout());

jf.add(pn);

jf.setSize(800, 800);
```

```

jf.validate();

Accountholders_List.addItemListener(new ItemListener() {

    public void itemStateChanged(ItemEvent ievt){

        try {

            Connection con3 =
            DriverManager.getConnection("jdbc:oracle:thin:@218.248.0.7:1521:rdbms","it20737041",
            "vasavi");

            Statement st3 = con3.createStatement();

            ResultSet rs3 = st3.executeQuery("select * from account_holder");

            while (rs3.next()) {

                if

                (rs3.getString("account_number").equals(Accountholders_List.getSelectedItem()))

                break;

            }

            if (!rs3.isAfterLast()) {

                JTF_account_number.setText(rs3.getString("account_number"));

                JTF_name.setText(rs3.getString("name"));

                JTF_contact_details.setText(rs3.getString("contact_details"));

                JTF_profession.setText(rs3.getString("profession"));

                JTF_aid.setText(rs3.getString("aid"));

            }

        }

        catch (SQLException selectException) {

            displaySQLErrors(selectException);

```



```

}

}

});

JB_modify.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent aevt){

try{

int a = JOptionPane.showConfirmDialog(pn, "Are you sure want to update:");

if(a == JOptionPane.YES_OPTION){

String query = "update account_holder set name = ?, contact_details = ?, profession = ?, aid = ?
where account_number = ?";

Connection con4
=DriverManager.getConnection("jdbc:oracle:thin:@218.248.0.7:1521:rdbms","it20737041",
"vasavi");

PreparedStatement stp2 = con4.prepareStatement(query);

stp2.setString(1, JTF_name.getText());

stp2.setString(2, JTF_contact_details.getText());

stp2.setString(3, JTF_profession.getText());

stp2.setString(4, JTF_aid.getText());

stp2.setString(5, JTF_account_number.getText());

int i = stp2.executeUpdate();{

JOptionPane.showMessageDialog(pn, "\nUpdated rows succesfully");

}

```

```
loaddetails();

}

}

catch(SQLException e){

displaySQLErrors(e);

}

}

});

}

});

delete2.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent aevt){

JB_delete = new JButton("Delete");

JTF_account_number.setText(null);

JTF_name.setText(null);

JTF_contact_details.setText(null);

JTF_profession.setText(null);

JTF_aid.setText(null);

pn.removeAll();

jf.invalidate();

jf.validate();

jf.repaint();
```

```
pn1 = new JPanel();

pn1.setLayout(new GridLayout(10, 10));

pn1.add(JL_account_number);

pn1.add(JTF_account_number);

pn1.add(JL_name);

pn1.add(JTF_name);

pn1.add(JL_contact_details);

pn1.add(JTF_contact_details);

pn1.add(JL_profession);

pn1.add(JTF_profession);

pn1.add(JL_aid);

pn1.add(JTF_aid);

pn3 = new JPanel(new FlowLayout());

pn3.add(JB_delete);

pn1.setBounds(115, 80, 300, 250);

pn3.setBounds(200, 350, 75, 35);

pn2 = new JPanel(new FlowLayout());

Accountholders_List = new List(10);

loaddetails();

pn2.add(Accountholders_List);

pn2.setBounds(200, 350, 300, 200);

pn.add(pn1);

pn.add(pn3);
```

```

pn.add(pn2);

pn.setLayout(new BorderLayout());

jf.add(pn);

jf.setSize(800, 800);

jf.validate();

Accountholders_List.addItemListener(new ItemListener() {

    public void itemStateChanged(ItemEvent ievt){

        try {

            Connection con5 =
            DriverManager.getConnection("jdbc:oracle:thin:@218.248.0.7:1521:rdbms","it20737041",
            "vasavi");

            Statement st4 = con5.createStatement();

            ResultSet rs4 = st4.executeQuery("select * from account_holder");

            while (rs4.next()) {

                if

                (rs4.getString("account_number").equals(Accountholders_List.getSelectedItem()))

                break;

            }

            if (!rs4.isAfterLast()) {

                JTF_account_number.setText(rs4.getString("account_number"));

                JTF_name.setText(rs4.getString("name"));

                JTF_contact_details.setText(rs4.getString("contact_details"));

                JTF_profession.setText(rs4.getString("profession"));

                JTF_aid.setText(rs4.getString("aid"));

```

```

}

}

catch (SQLException selectException) {

displaySQLErrors(selectException);

}

}

});

JB_delete.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent aevt){

try {

int a = JOptionPane.showConfirmDialog(pn, "Are you sure want to Delete:");

if(a == JOptionPane.YES_OPTION){

String query = "DELETE FROM account_number WHERE account_number = ? ";

Connection con6 =
DriverManager.getConnection("jdbc:oracle:thin:@218.248.0.7:1521:rdbms","it20737041",
"vasavi");

PreparedStatement stp3 = con6.prepareStatement(query);

stp3.setString(1, JTF_account_number.getText());

int i = stp3.executeUpdate(query);

if(i>0){

JOptionPane.showMessageDialog(pn, "\nDeleted rows succesfully");

}

loaddetails();

```

```
}

}

catch(SQLException e){

displaySQLErrors(e);

}

}

});

}

});

view2.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent aevt){

pn.removeAll();

jf.invalidate();

jf.validate();

jf.repaint();

JLabel view = new JLabel("account_holder details View");

view.setForeground(Color.BLACK);

JB_view = new JButton("View");

Font myFont = new Font("Serif", Font.BOLD,25);

view.setFont((myFont));

pn1 = new JPanel();

pn2 = new JPanel();
```

```
pn1.add(view);

pn2.add(JB_view);

pn.add(pn1);

pn.add(pn2);

pn.setBounds(500, 800, 300, 300);

pn.setLayout(new FlowLayout());

jf.add(pn);

jf.setSize(800, 800);

jf.validate();

JB_view.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent aevt){

JFrame jfr = new JFrame("account_holder Details");

JTable jt;

DefaultTableModel model = new DefaultTableModel();

jt = new JTable(model);

model.addColumn("account_number");

model.addColumn("Name");

model.addColumn("contact_details");

model.addColumn("profession");

model.addColumn("aid");

try {
```

```
Connection con7 =
DriverManager.getConnection("jdbc:oracle:thin:@218.248.0.7:1521:rdbms","it20737041",
"vasavi");

Statement st4 = con7.createStatement();

ResultSet rs5 = st4.executeQuery("select * from account_holder");

while(rs5.next()){

model.addRow(new Object[]{rs5.getString("account_number"), rs5.getString("name"),
rs5.getString("contact_details"), rs5.getString("profession"), rs5.getString("aid")});

}

}

catch(SQLException e){

displaySQLErrors(e);

}

jt.setEnabled(false);

jt.setBounds(30, 40, 300, 300);

JScrollPane jsp = new JScrollPane(jt);

jfr.add(jsp);

jfr.setSize(800, 400);

jfr.setVisible(true);

}

});

}

});

}}
```


Bankdetails :

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import javax.swing.table.DefaultTableModel;
import java.sql.*;
import java.util.Properties;
public class Bankdetails{
    private JPanel pn, pn1, pn2, pn3;
    private JFrame jf;
    private JButton JB_insert, JB_modify, JB_view, JB_delete;
    private JLabel JL_utr, JL_account_number, JL_annual_income, JL_address, JL_tax;
    private JTextField JTF_utr, JTF_account_number, JTF_annual_income, JTF_address,
    JTF_tax;
    private JMenuItem insert2, update2, view2, delete2;
    private List Accountholders_List;
    private Choice account_number;
    public Bankdetails(JPanel pn, JFrame jf, JMenuItem insert2, JMenuItem update2,
    JMenuItem view2, JMenuItem
    delete2){
        try {
            Class.forName("oracle.jdbc.driver.OracleDriver");
        }
        catch (Exception e) {
            System.err.println("Unable to find and load driver");
            System.exit(1);
        }
        this.jf = jf;
        this.insert2 = insert2;
        this.update2 = update2;
        this.view2 = view2;
        this.delete2 = delete2;
        JL_utr = new JLabel("Utr : ");
        JTF_utr = new JTextField(10);
        JL_account_number = new JLabel("account number : ");
        JTF_account_number = new JTextField(10);
        JL_annual_income = new JLabel("annual income : ");
        JTF_annual_income = new JTextField(10);
        JL_address = new JLabel("Address : ");
        JTF_address = new JTextField(10);
        JL_tax = new JLabel("Tax : ");
        JTF_tax = new JTextField(10);
        this.pn=pn;
    }
    private void displaySQLErrors(SQLException e) {
```

```

JOptionPane.showMessageDialog(pn1, "\nSQLException: " + e.getMessage() +
"\n"+"SQLState: " +
e.getSQLState() + "\n"+"VendorError: " + e.getErrorCode() + "\n");
}
public void load_details(){
try{
account_number = new Choice();
account_number.removeAll();
Connection con1 =
DriverManager.getConnection("jdbc:oracle:thin:@218.248.0.7:1521:rdbms", "it2073704
1", "vasavi");
Statement st1 = con1.createStatement();
ResultSet rs1 = st1.executeQuery("select * from bankdetails");
while(rs1.next()) {
account_number.add(rs1.getString("utr"));
}
}
catch(SQLException e) {
displaySQLErrors(e);
}
}
public void loaddetails(){
try{
Accountholders_List = new List();
Accountholders_List.removeAll();
Connection con2 =
DriverManager.getConnection("jdbc:oracle:thin:@218.248.0.7:1521:rdbms", "it2073704
1", "vasavi");
Statement st2 = con2.createStatement();
ResultSet rs2 = st2.executeQuery("select * from bankdetails");
while(rs2.next()) {
Accountholders_List.add(rs2.getString("utr"));
}
}
catch(SQLException e) {
displaySQLErrors(e);
}
}
public void buildGUI(){
insert2.addActionListener(new ActionListener(){
@Override
public void actionPerformed(ActionEvent aevt){
JB_insert = new JButton("Submit");
JTF_utr.setText(null);
JTF_account_number.setText(null);
JTF_annual_income.setText(null);

```

```

JTF_address.setText(null);
JTF_tax.setText(null);
loaddetails();
pn.removeAll();
jf.invalidate();
jf.validate();
jf.repaint();
pn1 = new JPanel();
pn1.setLayout(new GridLayout(10,10));
pn1.add(JL_utr);
pn1.add(JTF_utr);
pn1.add(JL_account_number);
pn1.add(JTF_account_number);
pn1.add(JL_annual_income);
pn1.add(JTF_annual_income);
pn1.add(JL_address);
pn1.add(JTF_address);
pn1.add(JL_tax);
pn1.add(JTF_tax);
pn3 = new JPanel(new FlowLayout());
pn3.add(JB_insert);
pn1.setBounds(115, 80, 300, 250);
pn3.setBounds(200, 350, 75, 35);
pn2 = new JPanel(new FlowLayout());
Accountholders_List = new List(10);
loaddetails();
pn2.add(Accountholders_List);
pn2.setBounds(200, 350, 300, 180);
pn.add(pn1);
pn.add(pn3);
pn.add(pn2);
pn.setLayout(new BorderLayout());
jf.add(pn);
jf.setSize(800, 800);
jf.validate();
JB_insert.addActionListener(new ActionListener() {
@Override
public void actionPerformed(ActionEvent aevt){
try{
Connection con =
DriverManager.getConnection("jdbc:oracle:thin:@218.248.0.7:1521:rdbms","it2073704
1", "vasavi");
String query = "INSERT INTO bankdetails (utr, account_number, annual_income, address,
tax) values(?, ?, ?, ?, ?)";
PreparedStatement stp = con.prepareStatement(query);
stp.setString(1, JTF_utr.getText());

```

```

stp.setString(2, JTF_account_number.getText());
stp.setString(3, JTF_annual_income.getText());
stp.setString(4, JTF_address.getText());
stp.setString(5, JTF_tax.getText());
int i = stp.executeUpdate();
con.close();
if(i > 0){
JOptionPane.showMessageDialog(pn,"\\nInserted successfully");
}
}
catch(SQLException e) {
displaySQLErrors(e);
}
}
});
}
});
update2.addActionListener(new ActionListener() {
@Override
public void actionPerformed(ActionEvent aevt){
JB_modify = new JButton("Modify");
JTF_utr.setText(null);
JTF_account_number.setText(null);
JTF_annual_income.setText(null);
JTF_address.setText(null);
JTF_tax.setText(null);
pn.removeAll();
jf.invalidate();
jf.validate();
jf.repaint();
pn1 = new JPanel();
pn1.setLayout(new GridLayout(10, 10));
pn1.add(JL_utr);
pn1.add(JTF_utr);
pn1.add(JL_account_number);
pn1.add(JTF_account_number);
pn1.add(JL_annual_income);
pn1.add(JTF_annual_income);
pn1.add(JL_address);
pn1.add(JTF_address);
pn1.add(JL_tax);
pn1.add(JTF_tax);
pn3 = new JPanel(new FlowLayout());
pn3.add(JB_modify);
pn1.setBounds(115, 80, 300, 250);
pn3.setBounds(200, 350, 75, 35);

```

```

pn2 =new JPanel(new FlowLayout());
Accountholders_List = new List(10);
loaddetails();
pn2.add(Accountholders_List);
pn2.setBounds(200, 350, 300, 180);
pn.add(pn1);
pn.add(pn3);
pn.add(pn2);
pn.setLayout(new BorderLayout());
jf.add(pn);
jf.setSize(800, 800);
jf.validate();
Accountholders_List.addItemListener(new ItemListener() {
public void itemStateChanged(ItemEvent ievt){
try {
Connection con3 =
DriverManager.getConnection("jdbc:oracle:thin:@218.248.0.7:1521:rdbms","it2073704
1", "vasavi");
Statement st3 = con3.createStatement();
ResultSet rs3 = st3.executeQuery("select * from bankdetails");
while (rs3.next()) {
if
(rs3.getString("utr").equals(Accountholders_List.getSelectedItem()))
break;
}
if (!rs3.isAfterLast()) {
JTF_utr.setText(rs3.getString("utr"));
JTF_account_number.setText(rs3.getString("account_number"));
JTF_annual_income.setText(rs3.getString("annual_income"));
JTF_tax.setText(rs3.getString("address"));
JTF_address.setText(rs3.getString("tax"));
}
}
catch (SQLException selectException) {
displaySQLErrors(selectException);
}
}
});
JB_modify.addActionListener(new ActionListener() {
@Override
public void actionPerformed(ActionEvent aevt){
try{
int a = JOptionPane.showConfirmDialog(pn, "Are you sure want to update:");
if(a == JOptionPane.YES_OPTION){
String query = "update bankdetails set accountnumber = ?, tax = ?, annualincome = ?,
address = ? where utr = ?";

```

```

Connection con4 =
DriverManager.getConnection("jdbc:oracle:thin:@218.248.0.7:1521:rdbms","it2073704
1", "vasavi");
PreparedStatement stp2 = con4.prepareStatement(query);
stp2.setString(1, JTF_account_number.getText());
stp2.setString(3, JTF_annual_income.getText());
stp2.setString(4, JTF_address.getText());
stp2.setString(2, JTF_tax.getText());
stp2.setString(5, JTF_utr.getText());
int i = stp2.executeUpdate();
if(i>0){
JOptionPane.showMessageDialog(pn,"\nUpdated rows succesfully");
}
loaddetails();
}
}
catch(SQLException e){
displaySQLErrors(e);
}
}
});
}
});
delete2.addActionListener(new ActionListener() {
@Override
public void actionPerformed(ActionEvent aevt){
JB_delete = new JButton("Delete");
JTF_utr.setText(null);
JTF_account_number.setText(null);
JTF_annual_income.setText(null);
JTF_address.setText(null);
JTF_tax.setText(null);
pn.removeAll();
jf.invalidate();
jf.validate();
jf.repaint();
pn1 = new JPanel();
pn1.setLayout(new GridLayout(10, 10));
pn1.add(JL_utr);
pn1.add(JTF_utr);
pn1.add(JL_account_number);
pn1.add(JTF_account_number);
pn1.add(JL_annual_income);
pn1.add(JTF_annual_income);
pn1.add(JL_address);
pn1.add(JTF_address);

```

```

pn1.add(JL_tax);
pn1.add(JTF_tax);
pn3 = new JPanel(new FlowLayout());
pn3.add(JB_delete);
pn1.setBounds(115, 80, 300, 250);
pn3.setBounds(200, 350, 75, 35);
pn2 = new JPanel(new FlowLayout());
Accountholders_List = new List(10);
loaddetails();
pn2.add(Accountholders_List);
pn2.setBounds(200, 350, 300, 200);
pn.add(pn1);
pn.add(pn3);
pn.add(pn2);
pn.setLayout(new BorderLayout());
jf.add(pn);
jf.setSize(800, 800);
jf.validate();
Accountholders_List.addItemListener(new ItemListener() {
    public void itemStateChanged(ItemEvent ievt){
        try {
            Connection con5 =
                DriverManager.getConnection("jdbc:oracle:thin:@218.248.0.7:1521:rdbms","it2073704
1", "vasavi");
            Statement st4 = con5.createStatement();
            ResultSet rs4 = st4.executeQuery("select * from bankdetails");
            while (rs4.next()) {
                if
                    (rs4.getString("utr").equals(Accountholders_List.getSelectedItem()))
                    break;
            }
            if (!rs4.isAfterLast()) {
                JTF_utr.setText(rs4.getString("utr"));
                JTF_account_number.setText(rs4.getString("account_number"));
                JTF_annual_income.setText(rs4.getString("annual_income"));
                JTF_tax.setText(rs4.getString("address"));
                JTF_address.setText(rs4.getString("tax"));
            }
        }
        catch (SQLException selectException) {
            displaySQLErrors(selectException);
        }
    }
});
JB_delete.addActionListener(new ActionListener() {
    @Override

```

```

public void actionPerformed(ActionEvent aevt){
try {
int a = JOptionPane.showConfirmDialog(pn,"Are you sure want to Delete:");
if(a == JOptionPane.YES_OPTION){
String query = "DELETE FROM bankdetails WHERE utr = ?";
Connection con6 =
DriverManager.getConnection("jdbc:oracle:thin:@218.248.0.7:1521:rdbms","it2073704
1", "vasavi");
PreparedStatement stp3 = con6.prepareStatement(query);
stp3.setString(1, JTF_utr.getText());
int i = stp3.executeUpdate();
if(i>0){
JOptionPane.showMessageDialog(pn,"\nDeleted rows succesfully");
}
loaddetails();
}
}
catch(SQLException e){
displaySQLErrors(e);
}
});
}
});
view2.addActionListener(new ActionListener() {
@Override
public void actionPerformed(ActionEvent aevt){
pn.removeAll();
jf.invalidate();
jf.validate();
jf.repaint();
JLabel view = new JLabel("bank details View");
view.setForeground(Color.BLACK);
JB_view = new JButton("View");
Font myFont = new Font("Serif",Font.BOLD,25);
view.setFont(myFont);
pn1 = new JPanel();
pn2 = new JPanel();
pn1.add(view);
pn2.add(JB_view);
pn.add(pn1);
pn.add(pn2);
pn.setBounds(500, 800, 300, 300);
pn.setLayout(new FlowLayout());
jf.add(pn);
jf.setSize(800, 800);

```



```

jf.validate();
JB_view.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent aevt){
        JFrame jfr = new JFrame("bank details");
        JTable jt;
        DefaultTableModel model = new DefaultTableModel();
        jt = new JTable(model);
        model.addColumn("utr");
        model.addColumn("account_number");
        model.addColumn("annual_income");
        model.addColumn("address");
        model.addColumn("tax");
        try {
            Connection con7 =
            DriverManager.getConnection("jdbc:oracle:thin:@218.248.0.7:1521:rdbms",
            "it20737041", "vasavi");
            Statement st4 = con7.createStatement();
            ResultSet rs5 = st4.executeQuery("select * from bankdetails");
            while(rs5.next()){
                model.addRow(new Object[]{rs5.getString("utr"), rs5.getString("account_number"),
                rs5.getString("annual_income"), rs5.getString("address"), rs5.getString("tax")});
            }
        }
        catch(SQLException e){
            displaySQLErrors(e);
        }
        jt.setEnabled(false);
        jt.setBounds(30, 40, 300, 300);
        JScrollPane jsp = new JScrollPane(jt);
        jfr.add(jsp);
        jfr.setSize(800, 400);
        jfr.setVisible(true);
    }
});
}
}
}
}

```

Payment :

```

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import javax.swing.table.DefaultTableModel;
import java.sql.*;
import java.util.Properties;

```

```

public class Payment{
private JPanel pn, pn1, pn2, pn3;
private JFrame jf;
private JButton JB_insert, JB_modify, JB_view, JB_delete;
private JLabel JL_payment_id, JL_tax, JL_due_date, JL_fine, JL_account_number;
private JTextField JTF_payment_id, JTF_tax, JTF_due_date, JTF_fine,
JTF_account_number;
private JMenuItem insert2, update2, view2, delete2;
private List Accountholders_List;
private Choice account_number;
public Payment(JPanel pn, JFrame jf, JMenuItem insert2, JMenuItem update2,
JMenuItem view2,
JMenuItem delete2){
try {
Class.forName("oracle.jdbc.driver.OracleDriver");
}
catch (Exception e) {
System.err.println("Unable to find and load driver");
System.exit(1);
}
this.jf = jf;
this.insert2 = insert2;
this.update2 = update2;
this.view2 = view2;
this.delete2 = delete2;
JL_payment_id = new JLabel("Payment id :");
JTF_payment_id = new JTextField(10);
JL_tax= new JLabel("Tax :");
JTF_tax = new JTextField(10);
JL_due_date = new JLabel("Due Date:");
JTF_due_date = new JTextField(10);
JL_fine = new JLabel("Fine :");
JTF_fine = new JTextField(10);
JL_account_number = new JLabel("Account number :");
JTF_account_number = new JTextField(10);
this.pn = pn;
}
private void displaySQLExceptions(SQLException e) {
JOptionPane.showMessageDialog(pn1, "\nSQLException: " + e.getMessage() +
"\n"+"SQLState: " +
e.getSQLState() + "\n"+"VendorError: " + e.getErrorCode() + "\n");
}
public void load_details(){
try{
account_number = new Choice();
account_number.removeAll();

```

```

Connection con1 =
DriverManager.getConnection("jdbc:oracle:thin:@218.248.0.7:1521:rdbms","it2073704
1", "vasavi");
Statement st1 = con1.createStatement();
ResultSet rs1 = st1.executeQuery("select * from Payment");
while(rs1.next()) {
account_number.add(rs1.getString("payment_id"));
}
}
catch(SQLException e) {
displaySQLErrors(e);
}
}
public void loaddetails(){
try{
Accountholders_List = new List();
Accountholders_List.removeAll();
Connection con2 =
DriverManager.getConnection("jdbc:oracle:thin:@218.248.0.7:1521:rdbms","it2073704
1", "vasavi");
Statement st2 = con2.createStatement();
ResultSet rs2 = st2.executeQuery("select * from Payment");
while(rs2.next()) {
Accountholders_List.add(rs2.getString("payment_id"));
}
}
catch(SQLException e) {
displaySQLErrors(e);
}
}
public void buildGUI(){
insert2.addActionListener(new ActionListener(){
@Override
public void actionPerformed(ActionEvent aevt){
JB_insert = new JButton("Submit");
JTF_payment_id.setText(null);
JTF_tax.setText(null);
JTF_due_date.setText(null);
JTF_fine.setText(null);
JTF_account_number.setText(null);
loaddetails();
pn.removeAll();
jf.invalidate();
jf.validate();
jf.repaint();
pn1 = new JPanel();

```

```

pn1.setLayout(new GridLayout(10,10));
pn1.add(JL_payment_id);
pn1.add(JTF_payment_id);
pn1.add(JL_tax);
pn1.add(JTF_tax);
pn1.add(JL_due_date);
pn1.add(JTF_due_date);
pn1.add(JL_fine);
pn1.add(JTF_fine);
pn1.add(JL_account_number);
pn1.add(JTF_account_number);
pn3 = new JPanel(new FlowLayout());
pn3.add(JB_insert);
pn1.setBounds(115, 80, 300, 250);
pn3.setBounds(200, 350, 75, 35);
pn2 = new JPanel(new FlowLayout());
Accountholders_List = new List(10);
loaddetails();
pn2.add(Accountholders_List);
pn2.setBounds(200, 350, 300, 180);
pn.add(pn1);
pn.add(pn3);
pn.add(pn2);
pn.setLayout(new BorderLayout());
jf.add(pn);
jf.setSize(800, 800);
jf.validate();
JB_insert.addActionListener(new ActionListener() {
@Override
public void actionPerformed(ActionEvent aevt){
try{
Connection con =
DriverManager.getConnection("jdbc:oracle:thin:@218.248.0.7:1521:rdbms","it2073704
1","vasavi");
String query = "INSERT INTO Payment (payment_id,tax, due_date, fine,
account_number) values(?, ?, ?, ?)";
PreparedStatement stp1 = con.prepareStatement(query);
stp1.setString(1, JTF_payment_id.getText());
stp1.setString(2, JTF_tax.getText());
stp1.setString(3, JTF_due_date.getText());
stp1.setString(4, JTF_fine.getText());
stp1.setString(4, JTF_account_number.getText());
int i = stp1.executeUpdate();
con.close();
if(i > 0){
JOptionPane.showMessageDialog(pn,"\\nInserted successfully");

```

```

}
loaddetails();
}
catch(SQLException e) {
displaySQLErrors(e);
}
}
});
}
});
update2.addActionListener(new ActionListener() {
@Override
public void actionPerformed(ActionEvent aevt){
JB_modify=new JButton("Modify");
JTF_payment_id.setText(null);
JTF_tax.setText(null);
JTF_due_date.setText(null);
JTF_fine.setText(null);
JTF_account_number.setText(null);
pn.removeAll();
jf.invalidate();
jf.validate();
jf.repaint();
pn1=new JPanel();
pn1.setLayout(new GridLayout(10, 10));
pn1.add(JL_payment_id);
pn1.add(JTF_payment_id);
pn1.add(JL_tax);
pn1.add(JTF_tax);
pn1.add(JL_due_date);
pn1.add(JTF_due_date);
pn1.add(JL_fine);
pn1.add(JTF_fine);
pn1.add(JL_account_number);
pn1.add(JTF_account_number);
pn3 = new JPanel(new FlowLayout());
pn3.add(JB_modify);
pn1.setBounds(115, 80, 300, 250);
pn3.setBounds(200, 350, 75, 35);
pn2 =new JPanel(new FlowLayout());
Accountholders_List = new List(10);
loaddetails();
pn2.add(Accountholders_List);
pn2.setBounds(200, 350, 300, 180);
pn.add(pn1);
pn.add(pn3);

```

```

pn.add(pn2);
pn.setLayout(new BorderLayout());
jf.add(pn);
jf.setSize(800, 800);
jf.validate();
Accountholders_List.addItemListener(new ItemListener() {
    public void itemStateChanged(ItemEvent ievt){
        try {
            Connection con3 =
            DriverManager.getConnection("jdbc:oracle:thin:@218.248.0.7:1521:rdbms",
            "it20737041", "vasavi");
            Statement st3 = con3.createStatement();
            ResultSet rs3 = st3.executeQuery("select * from Payment");
            while (rs3.next()) {
                if
                (rs3.getString("payment_id").equals(Accountholders_List.getSelectedItem()))
                break;
            }
            if (!rs3.isAfterLast()) {
                JTF_payment_id.setText(rs3.getString("payment_id"));
                JTF_tax.setText(rs3.getString("tax"));
                JTF_due_date.setText(rs3.getString("due_date"));
                JTF_fine.setText(rs3.getString("fine"));
                JTF_account_number.setText(rs3.getString("account_number"));
            }
        }
        catch (SQLException selectException) {
            displaySQLErrors(selectException);
        }
    }
});
JB_modify.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent aevt){
        try{
            int a = JOptionPane.showConfirmDialog(pn, "Are you sure want to update:");
            if(a == JOptionPane.YES_OPTION){
                String query = "update Payment set tax = ?, fine = ?, due_date = ?, account_number = ?
                where paymentid = ?";
                Connection con4 =
                DriverManager.getConnection("jdbc:oracle:thin:@218.248.0.7:1521:rdbms",
                "it20737041", "vasavi");
                PreparedStatement stp2 = con4.prepareStatement(query);
                stp2.setString(1, JTF_tax.getText());
                stp2.setString(2, JTF_fine.getText());
                stp2.setString(3, JTF_due_date.getText());
            }
        }
    }
});

```

```

stp2.setString(4, JTF_account_number.getText());
stp2.setString(5, JTF_payment_id.getText());
int i = stp2.executeUpdate();
if(i>0){
JOptionPane.showMessageDialog(pn,"\\nUpdated rows succesfully");
}
loaddetails();
}
}
catch(SQLException e){
displaySQLErrors(e);
}
}
});
}
});
delete2.addActionListener(new ActionListener() {
@Override
public void actionPerformed(ActionEvent aevt){
JB_delete = new JButton("Delete");
JTF_payment_id.setText(null);
JTF_tax.setText(null);
JTF_due_date.setText(null);
JTF_fine.setText(null);
JTF_account_number.setText(null);
pn.removeAll();
jf.invalidate();
jf.validate();
jf.repaint();
pn1 = new JPanel();
pn1.setLayout(new GridLayout(10, 10));
pn1.add(JL_payment_id);
pn1.add(JTF_payment_id);
pn1.add(JL_tax);
pn1.add(JTF_tax);
pn1.add(JL_due_date);
pn1.add(JTF_due_date);
pn1.add(JL_fine);
pn1.add(JTF_fine);
pn1.add(JL_account_number);
pn1.add(JTF_account_number);
pn3 = new JPanel(new FlowLayout());
pn3.add(JB_delete);
pn1.setBounds(115, 80, 300, 250);
pn3.setBounds(200, 350, 75, 35);
pn2 = new JPanel(new FlowLayout());

```

```

Accountholders_List = new List(10);
loaddetails();
pn2.add(Accountholders_List);
pn2.setBounds(200, 350, 300, 200);
pn.add(pn1);
pn.add(pn3);
pn.add(pn2);
pn.setLayout(new BorderLayout());
jf.add(pn);
jf.setSize(800, 800);
jf.validate();
Accountholders_List.addItemListener(new ItemListener() {
    public void itemStateChanged(ItemEvent ievt){
        try {
            Connection con5 =
                DriverManager.getConnection("jdbc:oracle:thin:@218.248.0.7:1521:rdbms",
                    "it20737041", "vasavi");
            Statement st4 = con5.createStatement();
            ResultSet rs4 = st4.executeQuery("select * from Payment");
            while (rs4.next()) {
                if
                    (rs4.getString("account_number").equals(Accountholders_List.getSelectedItem()))
                    break;
            }
            if (!rs4.isAfterLast()) {
                JTF_payment_id.setText(rs4.getString("payment_id"));
                JTF_tax.setText(rs4.getString("tax"));
                JTF_due_date.setText(rs4.getString("due_date"));
                JTF_fine.setText(rs4.getString("fine"));
                JTF_account_number.setText(rs4.getString("account_number"));
            }
        }
        catch (SQLException selectException) {
            displaySQLErrors(selectException);
        }
    }
});
JB_delete.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent aevt){
        try {
            int a = JOptionPane.showConfirmDialog(pn,"Are you sure want to Delete:");
            if(a == JOptionPane.YES_OPTION){
                String query = "DELETE FROM Payment WHERE payment_id = ?";
            }
        }
    }
});

```



```

Connection con6 =
DriverManager.getConnection("jdbc:oracle:thin:@218.248.0.7:1521:rdbms",
"it20737041", "vasavi");
PreparedStatement stp3 = con6.prepareStatement(query);
stp3.setString(1, JTF_payment_id.getText());
int i = stp3.executeUpdate();
if(i > 0){
JOptionPane.showMessageDialog(pn,"\\nDeleted rows succesfully");
}
loaddetails();
}
}
catch(SQLException e){
displaySQLErrors(e);
}
}
});
}
});
view2.addActionListener(new ActionListener() {
@Override
public void actionPerformed(ActionEvent aevt){
pn.removeAll();
jf.invalidate();
jf.validate();
jf.repaint();
JLabel view = new JLabel("payment details View");
view.setForeground(Color.BLACK);
JB_view = new JButton("View");
Font myFont = new Font("Serif",Font.BOLD,25);
view.setFont((myFont));
pn1 = new JPanel();
pn2 = new JPanel();
pn1.add(view);
pn2.add(JB_view);
pn.add(pn1);
pn.add(pn2);
pn.setBounds(500, 800, 300, 300);
pn.setLayout(new FlowLayout());
jf.add(pn);
jf.setSize(800, 800);
jf.validate();
JB_view.addActionListener(new ActionListener() {
@Override
public void actionPerformed(ActionEvent aevt){
JFrame jfr = new JFrame("Payment details");

```

```

JTable jt;
DefaultTableModel model = new DefaultTableModel();
jt = new JTable(model);
model.addColumn("payment_id");
model.addColumn("tax");
model.addColumn("due_date");
model.addColumn("fine");
model.addColumn("account_number");
try {
    Connection con7 =
    DriverManager.getConnection("jdbc:oracle:thin:@218.248.0.7:1521:rdbms",
    "it20737041", "vasavi");
    Statement st4 = con7.createStatement();
    ResultSet rs5 = st4.executeQuery("select * from payment");
    while(rs5.next()){
        model.addRow(new Object[]{rs5.getString("payment_id"), rs5.getString("tax"),
        rs5.getString("due_date"), rs5.getString("fine"), rs5.getString("account_number")});
    }
}
catch(SQLException e){
    displaySQLErrors(e);
}
jt.setEnabled(false);
jt.setBounds(30, 40, 300, 300);
JScrollPane jsp = new JScrollPane(jt);
jfr.add(jsp);
jfr.setSize(800, 400);
jfr.setVisible(true);
}
});
}
});
}
}
}
}
}

```

Portal :

```
import java.awt.*;
```

```
import javax.swing.*;
```

```
import java.awt.Color;
```

```
import java.awt.FlowLayout;
```

```
import java.awt.Font;
```

```
import java.awt.GridLayout;

import java.awt.Label;

import java.awt.event.WindowAdapter;

import java.awt.event.WindowEvent;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JMenu;

import javax.swing.JMenuBar;

import javax.swing.JMenuItem;

import javax.swing.JOptionPane;

import javax.swing.JPanel;

public class PortalUI extends JFrame{

    private static final long serialVersionUID = 1L;

    private JMenuBar mnu;

    private JMenu mnuAccount_holder;

    private JMenu mnuBankdetails;

    private JMenu mnuPayment;


    private JMenuItem insert1, update1, delete1, view1;

    private JMenuItem insert2, update2, delete2, view2;

    private JMenuItem insert3, update3, delete3, view3;

    private JLabel labelName;

    private static JPanel p0, p1;
```

```
public void initialize() {

    mnu = new JMenuBar();

    mnuAccount_holder = new JMenu("Account Holder");

    mnuBankdetails = new JMenu("Bank details");

    mnuPayment = new JMenu("Payment");

    Color lightblue= new Color(51, 204, 255);

    labelName=new JLabel("IT Returns Processing Database");

    labelName.setFont(new Font("Serif", Font.PLAIN, 40));

    labelName.setForeground(Color.BLACK);

    p1=new JPanel();

    p0=new JPanel();

    insert1 = new JMenuItem("Insert");

    update1 = new JMenuItem("Update");

    delete1 = new JMenuItem("Delete");

    view1 = new JMenuItem("View");

    insert2 = new JMenuItem("Insert");

    update2 = new JMenuItem("Update");

    delete2 = new JMenuItem("Delete");

    view2 = new JMenuItem("View");

    insert3 = new JMenuItem("Insert");

    update3 = new JMenuItem("Update");

    delete3 = new JMenuItem("Delete");

    view3 = new JMenuItem("View");
```

```
}

void addComponentsToFrame() {

    mnuAccount_holder.add(insert1);

    mnuAccount_holder.add(delete1);

    mnuAccount_holder.add(update1);

    mnuAccount_holder.add(view1);

    mnuBankdetails.add(insert2);

    mnuBankdetails.add(delete2);

    mnuBankdetails.add(update2);

    mnuBankdetails.add(view2);

    mnuPayment.add(insert3);

    mnuPayment.add(delete3);

    mnuPayment.add(update3);

    mnuPayment.add(view3);

    mnu.add(mnuAccount_holder);

    mnu.add(mnuBankdetails);

    mnu.add(mnuPayment);

    setJMenuBar(mnu);

    p1.add(labelName);

    p1.setAlignmentY(CENTER_ALIGNMENT);

    p1.setBounds(500, 500, 800, 100);

    p0.add(p1);

    p0.setBackground(Color.WHITE);
```

```

add(p0);

}

void closeWindow(){

try {

int a = JOptionPane.showConfirmDialog(this,"Are you sure want to Quit ?");

if(a == JOptionPane.YES_OPTION){

System.exit(0);

}

else if (a == JOptionPane.NO_OPTION) {

setDefaultCloseOperation(JFrame.DO_NOTHING_ON_CLOSE);

}

else if (a == JOptionPane.CANCEL_OPTION) {

setDefaultCloseOperation(JFrame.DO_NOTHING_ON_CLOSE);

}

}

catch(Exception e) {

System.out.println(e);

}

}

void register() {

Account_holder ah=new Account_holder(p0, PortalUI.this, insert1, update1, view1, delete1);

ah.buildGUI();

Bankdetails bd = new Bankdetails(p0, PortalUI.this, insert2, update2, view2, delete2);

```

```
bd.buildGUI();
```

```
Payment pt= new Payment(p0, PortalUI.this, insert3, update3, view3, delete3);
```

```
pt.buildGUI();
```

```
addWindowListener(new WindowAdapter(){
```

```
public void windowClosing(WindowEvent we)
```

```
{
```

```
closeWindow();
```

```
}
```

```
});
```

```
}
```

```
public PortalUI() {
```

```
initialize();
```

```
addComponentsToFrame();
```

```
register();
```

```
pack();
```

```
setTitle("IT Returns Processing Database");
```

```
setSize(800, 800);
```

```
setVisible(true);
```

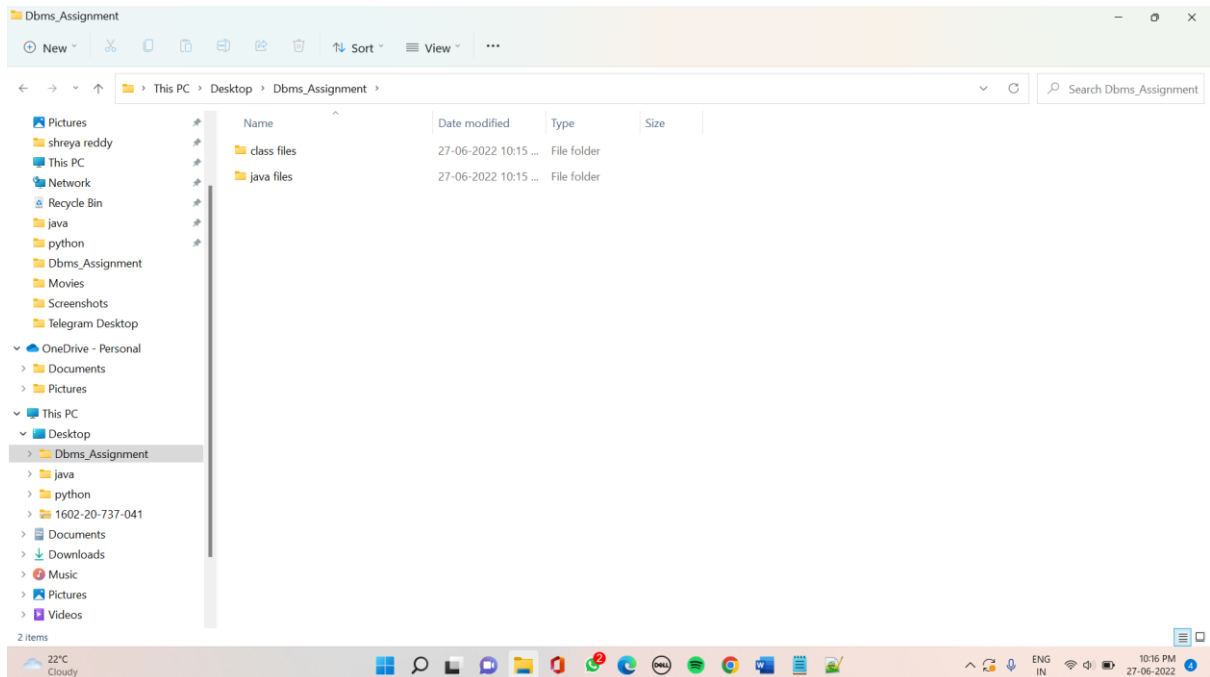
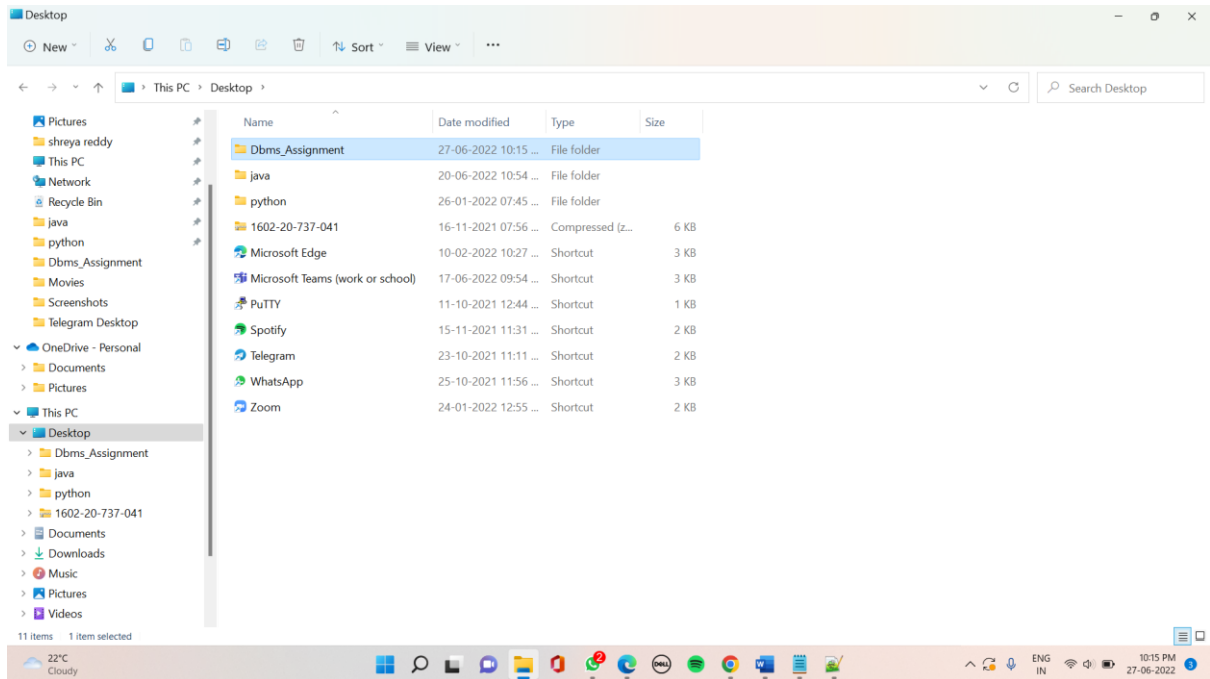
```
}
```

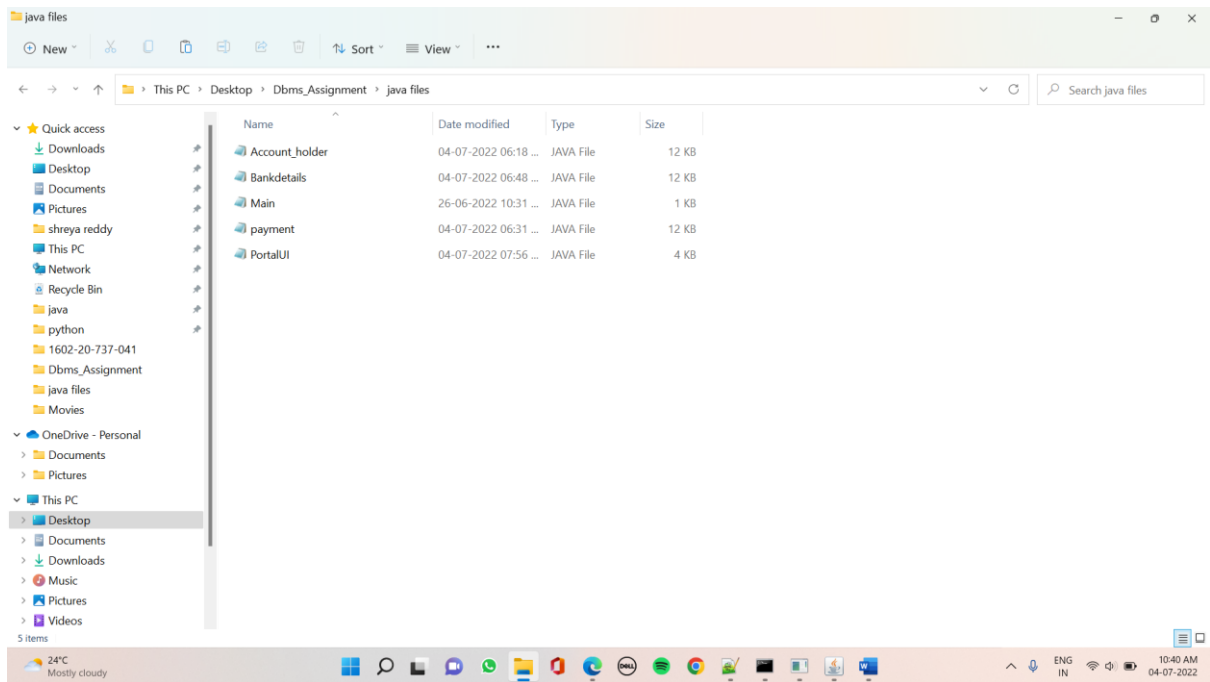
```
}
```

Github link and Folder Structure :

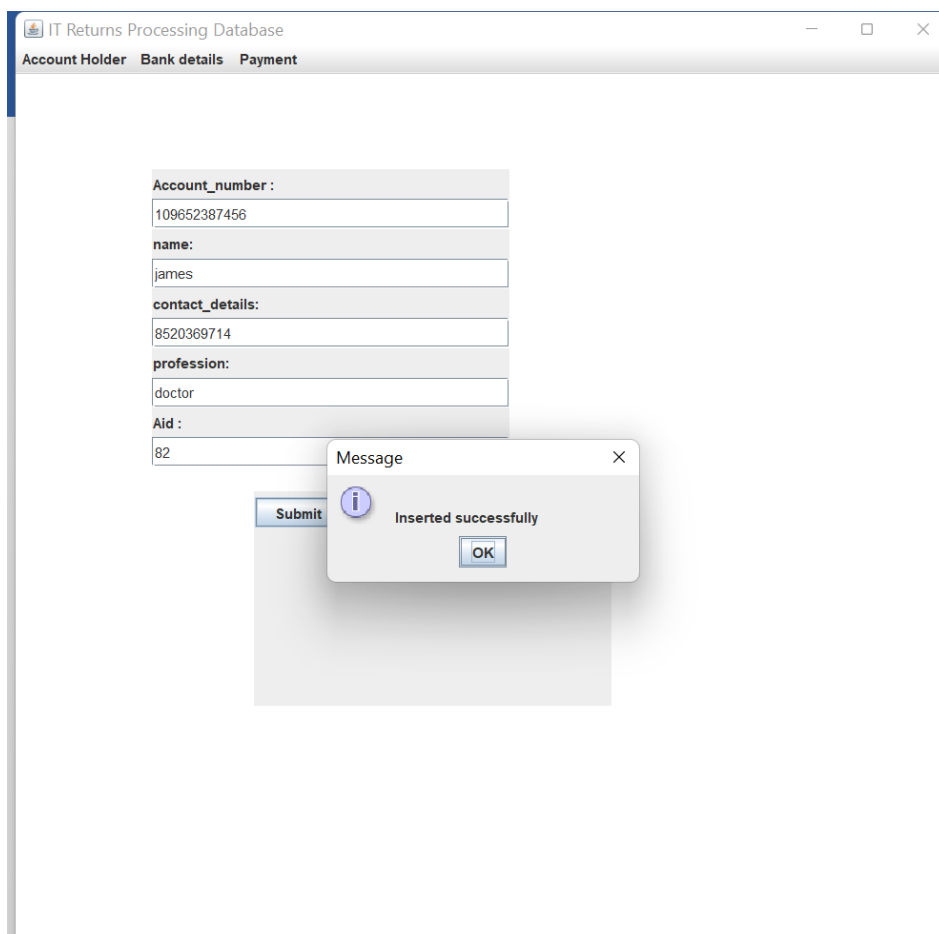
Link :

<https://github.com/Shreyareddy2803/IT-returns-processing-database>





TESTING :



IT Returns Processing Database

Account Holder Bank details Payment

Account_number :
109652387456

name:
kate

contact_details:
8523697410

profession:
doctor

Aid :
82

Modify

Message

Updated rows succesfully

OK

IT Returns Processing Database

Account Holder Bank details Payment

Account_number :
123698574026

name:
hxs

contact_details:
9563526323

profession:
hxue

Aid :
23

Delete

Select an Option

Are you sure want to Delete:

Yes No Cancel

account_holder Details				
account_number	Name	contact_details	profession	aid
109652387456	kate	8523697410	doctor	82
123698574026	hxs	9563526323	hxue	23
123456987023	john	9876543210	doctor	12
987654321078	shreya	9632587410	engineer	15

IT Returns Processing Database

Account Holder Bank details Payment

Utr :

2563

account number :

987654321078

annual income :

5000000

Address :

hyd

Tax :

50000

Submit

Message

Inserted successfully

OK

bank details				
utr	account_number	annual_income	address	tax
2563	987654321078	5000000	hyd	50000
2356	123698574026	5000000	hyd	20000

IT Returns Processing Database

Account Holder Bank details Payment

Utr :

2356

account number :

123698574026

annual income :

5000000

Address :

mumbai

Tax :

20000

Modify

Select an Option

?

Are you sure want to update:

YesNoCancel

IT Returns Processing Database

Account HolderBank detailsPayment

Payment id :

23

Tax :

50000

Due Date:

july 3

Fine :

5600

Account number :

987654321078

Submit

IT Returns Processing Database

Account HolderBank detailsPayment

payment details View

View

Select an Option

?

Are you sure want to Quit ?

Yes

No

Cancel

DATABASE OUTPUTS :

```
C:\Users\shreya reddy\Downloads\instantclient_19_8\instantclient_19_8\sqlplus.exe
```

ACCOUNT_NUMBER	NAME	CONTACT_DETAILS	PROFESSION

AID			

1.0965E+11	kate	8523697410	doctor
82			
1.2370E+11	hxs	9563526323	hxue
23			
1.2346E+11	john	9876543210	doctor
12			

ACCOUNT_NUMBER	NAME	CONTACT_DETAILS	PROFESSION

AID			

9.8765E+11	shreya	9632587410	engineer
15			

```
SQL> _
```

```
C:\Users\shreya reddy\Downloads\instantclient_19_8\instantclient_19_8\sqlplus.exe
```

```
SQL> select * from bankdetails;
```

UTR	ANNUAL_INCOME	ADDRESS	TAX

ACCOUNT_NUMBER			

2563	5000000	hyd	50000
9.8765E+11			
2356	5000000	hyd	20000
1.2370E+11			

```
SQL> _
```

```
C:\Users\shreya reddy\Downloads\instantclient_19_8\instantclient_19_8\sqlplus.exe
SQL> select * from payment;

PAYMENT_ID      TAX DUE_DATE      FINE
-----
2503            21000 -2023        2000
2035            25000 -2021        3000
2968            30000 -2022        1500
2541            20000 -2020        2000
2614            12000 -2024        2000
3614            19000 -2026        3000

6 rows selected.

SQL>
```

DISCUSSION AND FUTURE WORK

So far this project has helped us to register for any project that is conducted online by faculty and students will be able to register for those projects and easily submit their ideas. In the future, we try to include an option which allows students to review their projects which are checked by the faculty and can clarify their mistakes.

REFERENCES:

1. Abraham Silberschatz, Henry F Korth, S. Sudarshan, Database System Concepts, 6th Edition, McGraw-Hill International Edition, 2010.
2. https://docs.oracle.com/cd/E11882_01/server.112/e40540/intro.htm#CNCPT001
3. <https://docs.oracle.com/javase/8/docs/>