

JAVA AWT BASED - PETROL BUNK AUTOMATION SYSTEM - 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

Swetha Sai Dakupati <1602-19-737-093>

Under the guidance of Ms B. Leelavathy



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

BONAFIDE CERTIFICATE

This is to certify that this project report titled '**Petrol Bunk Automation System**' is a project work of Ms. Swetha Sai Dakupati bearing roll no. 1602-19-737-093 who carried out the project under my supervision in the IV semester for the academic year 2020- 2021.

Signature

Internal Examiner

Signature

External Examiner

ABSTRACT

Now-a-days petroleum products made an impact by performing a very important role in this world. The cost of demand is very high. The day-to-day analysis describes that the demand of fuel kept increasing but no other alternatives have been found to it.

Petrol Bunk Automation System is a project which contains the entire assembly system of the petrol bunk and its database. The main attraction of this project is that it eliminates human intervention.

On completion of a transaction, when the filling gets completed i.e; the fuel is filled on a particular amount of requirement as per the client and at the end of the transaction, the amount gets added to the old transaction and billing is made for the scheduled time for the payment from the client's balance.

INTRODUCTION

REQUIREMENTS

<u>Table name</u>	<u>Attributes</u>
Bunk	city varchar2(20) area varchar2(20) name varchar2(30)
Petrol_price	city varchar2(30) area varchar2(20) day date price number
Diesel_price	city varchar2(30) area varchar2(20) day date price number
Register	name varchar2(20) phno number(10) pin number(4)
Update_balance	phno number(10) amt number
Booking_petrol	phno number(10) amt number
Booking_diesel	phno number(10) amt number
Feedback	sno number phno number(10) experience varchar2(100) noofstars number complaint varchar2(100)

AIM AND PRIORITY OF THE PROJECT

To create a Java GUI based Petrol Bunk Automation System which takes values like : user phone number, user name etc from person who registers and uses the application for the filling of fuel into his/her vehicle. All these values are to be updated in the database using JDBC connectivity .

ARCHITECTURE AND TECHNOLOGY

Software used:

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

Java AWT:

Java AWT (Abstract Window Toolkit) is an API to develop GUI or window-based applications in java.

Java AWT components are platform-dependent i.e. components are displayed according to the view of operating system. AWT is heavyweight i.e. its components are using the resources of OS.

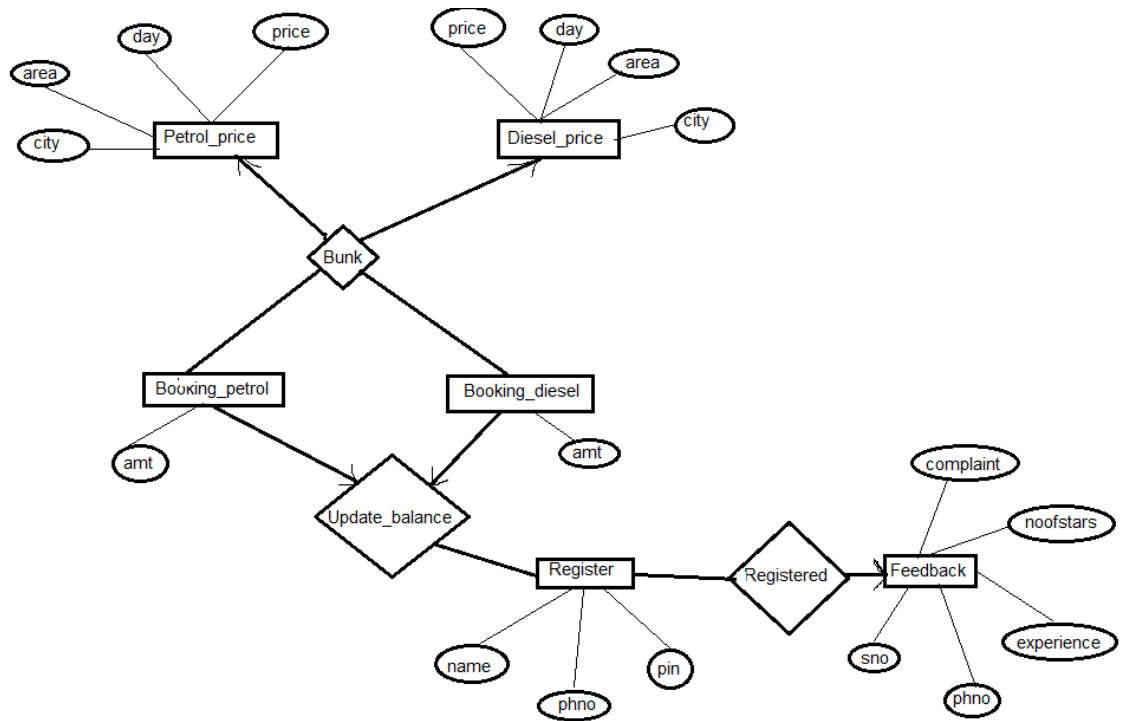
The java.awt package provides classes for AWT API such as TextField, Label, TextArea, RadioButton, CheckBox, Choice, List etc.

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.

DESIGN

Entity relationship diagram



Database Design:

DDL Operations

```

SQL> desc bunk;
Name                               Null?   Type
-----
CITY                               NOT NULL VARCHAR2(20)
AREA                               NOT NULL VARCHAR2(20)
NAME                               NOT NULL VARCHAR2(30)

SQL> desc petrol_price;
Name                               Null?   Type
-----
CITY                               NOT NULL VARCHAR2(30)
AREA                               NOT NULL VARCHAR2(20)
DAY                                NOT NULL DATE
PRICE                             NOT NULL NUMBER

SQL>
SQL> desc diesel_price;
Name                               Null?   Type
-----
CITY                               NOT NULL VARCHAR2(30)
AREA                               NOT NULL VARCHAR2(20)
DAY                                NOT NULL DATE
PRICE                             NOT NULL NUMBER

SQL> desc register
Name                               Null?   Type
-----
NAME                               NOT NULL VARCHAR2(20)
PHNO                               NOT NULL NUMBER(10)
PIN                                NOT NULL NUMBER(4)

SQL> desc update_balance
Name                               Null?   Type
-----
PHNO                               NOT NULL NUMBER(10)
AMT                                NOT NULL NUMBER

SQL> desc booking_petrol
Name                               Null?   Type
-----
PHNO                               NOT NULL NUMBER(10)
AMT                                NOT NULL NUMBER

SQL> desc booking_diesel
Name                               Null?   Type
-----
PHNO                               NOT NULL NUMBER(10)
AMT                                NOT NULL NUMBER

SQL> desc feedback

```



```
SQL> desc feedback
```

Name	Null?	Type
SNO	NOT NULL	NUMBER
PHNO	NOT NULL	NUMBER(10)
EXPERIENCE		VARCHAR2(100)
NOOFSTARS		NUMBER
COMPLAINT		VARCHAR2(100)

```
SQL>
```

DML Operations:

Bunk table

```
Run SQL Command Line

SQL> INSERT INTO bunk VALUES('&city','&area','&name');
Enter value for city: Hyderabad
Enter value for area: Charminar
Enter value for name: HPPotLtd.
old 1: INSERT INTO bunk VALUES('&city','&area','&name')
new 1: INSERT INTO bunk VALUES('Hyderabad','Charminar','HPPotLtd.')

1 row created.

SQL> /
Enter value for city: Vijayawada
Enter value for area: BenzCircle
Enter value for name: BharatPetroleum
old 1: INSERT INTO bunk VALUES('&city','&area','&name')
new 1: INSERT INTO bunk VALUES('Vijayawada','BenzCircle','BharatPetroleum')

1 row created.

SQL> /
Enter value for city: Hyderabad
Enter value for area: MehdiPatnam
Enter value for name: BharatFuelsLtd.
old 1: INSERT INTO bunk VALUES('&city','&area','&name')
new 1: INSERT INTO bunk VALUES('Hyderabad','MehdiPatnam','BharatFuelsLtd.')

1 row created.

SQL> /
Enter value for city: Vijayawada
Enter value for area: AjithSinghNagar
Enter value for name: BharatPetroleum
old 1: INSERT INTO bunk VALUES('&city','&area','&name')
new 1: INSERT INTO bunk VALUES('Vijayawada','AjithSinghNagar','BharatPetroleum')

1 row created.

SQL> /
Enter value for city: Vijayawada
Enter value for area: PUP
Enter value for name: HPPetroleum
old 1: INSERT INTO bunk VALUES('&city','&area','&name')
new 1: INSERT INTO bunk VALUES('Vijayawada','PUP','HPPetroleum')

1 row created.
```

Petrol_price

```

SQL> insert into petrol_price values('&city','&area','&day',&price);
Enter value for city: Hyderabad
Enter value for area: Mehdiapatnam
Enter value for day: 08-may-21
Enter value for price: 89
old 1: insert into petrol_price values('&city','&area','&day',&price)
new 1: insert into petrol_price values('Hyderabad','Mehdiapatnam','08-may-21',89)

1 row created.

SQL> /
Enter value for city: Vijayawada
Enter value for area: Hungary
Enter value for day: 02-aug-98
Enter value for price: 36
old 1: insert into petrol_price values('&city','&area','&day',&price)
new 1: insert into petrol_price values('Vijayawada','Hungary','02-aug-98',36)

1 row created.

SQL> /
Enter value for city: Hyderabad
Enter value for area: IbrahimBagh
Enter value for day: 02-feb-21
Enter value for price: 62
old 1: insert into petrol_price values('&city','&area','&day',&price)
new 1: insert into petrol_price values('Hyderabad','IbrahimBagh','02-feb-21',62)

1 row created.

SQL> /
Enter value for city: IbrahimBagh
Enter value for area: Hyderabad
Enter value for day: 04-jan-21
Enter value for price: 96
old 1: insert into petrol_price values('&city','&area','&day',&price)
new 1: insert into petrol_price values('IbrahimBagh','Hyderabad','04-jan-21',96)

1 row created.

SQL> /
Enter value for city: Vijayawada
Enter value for area: Benz circle
Enter value for day: 04-apr-20
Enter value for price: 63
old 1: insert into petrol_price values('&city','&area','&day',&price)
new 1: insert into petrol_price values('Vijayawada','Benz circle','04-apr-20',63)

1 row created.

SQL> select * from petrol_price;

```

```

SQL> select * from petrol_price;

```

CITY	AREA	DAY	PRICE
Hyderabad	Mehdiapatnam	08-MAY-21	89
Vijayawada	Hungary	02-AUG-98	36
Hyderabad	IbrahimBagh	02-FEB-21	62
IbrahimBagh	Hyderabad	04-JAN-21	96
Vijayawada	Benz circle	04-APR-20	63

Diesel_price

```
Run SQL Command Line

SQL> insert into diesel_price values('&city','&area','&day',&price);
Enter value for city: Vijayawada
Enter value for area: Benz circle
Enter value for day: 04-apr-20
Enter value for price: 98
old 1: insert into diesel_price values('&city','&area','&day',&price)
new 1: insert into diesel_price values('Vijayawada','Benz circle','04-apr-20',
98)

1 row created.

SQL> /
Enter value for city: IbrahimBagh
Enter value for area: Hyderabad
Enter value for day: 04-jan-21
Enter value for price: 126
old 1: insert into diesel_price values('&city','&area','&day',&price)
new 1: insert into diesel_price values('IbrahimBagh','Hyderabad','04-jan-21',1
26)

1 row created.

SQL> /
Enter value for city: Hyderabad
Enter value for area: IbrahimBagh
Enter value for day: 04-jun-20
Enter value for price: 111
old 1: insert into diesel_price values('&city','&area','&day',&price)
new 1: insert into diesel_price values('Hyderabad','IbrahimBagh','04-jun-20',1
11)

1 row created.

SQL> /
Enter value for city: Vijayawada
Enter value for area: Hungary
Enter value for day: 02-aug-98
Enter value for price: 60
old 1: insert into diesel_price values('&city','&area','&day',&price)
new 1: insert into diesel_price values('Vijayawada','Hungary','02-aug-98',60)

1 row created.

SQL> /
Enter value for city: Hyderabad
Enter value for area: Mehdipatnam
Enter value for day: 03-sep-01
Enter value for price: 30
old 1: insert into diesel_price values('&city','&area','&day',&price)
new 1: insert into diesel_price values('Hyderabad','Mehdipatnam','03-sep-01',3
0)

1 row created.

SQL> _

Run SQL Command Line

SQL> select * from diesel_price;

CITY                AREA                DAY                PRICE
-----
Vijayawada          Benz circle          04-APR-20           98
IbrahimBagh          Hyderabad            04-JAN-21           126
Hyderabad            IbrahimBagh          04-JUN-20           111
Vijayawada          Hungary              02-AUG-98           60
Hyderabad            Mehdipatnam          03-SEP-01           30
```

Register

```
SQL> insert into register values('&name',&phno,&pin);
Enter value for name: swetha
Enter value for phno: 6305424006
Enter value for pin: 1234
old 1: insert into register values('&name',&phno,&pin)
new 1: insert into register values('swetha',6305424006,1234)

1 row created.

SQL> /
Enter value for name: swe
Enter value for phno: 60042454036
Enter value for pin: 4321
old 1: insert into register values('&name',&phno,&pin)
new 1: insert into register values('swe',60042454036,4321)
insert into register values('swe',60042454036,4321)
*
ERROR at line 1:
ORA-01438: value larger than specified precision allowed for this column

SQL> /
Enter value for name: sai
Enter value for phno: 9988776655
Enter value for pin: 9632
old 1: insert into register values('&name',&phno,&pin)
new 1: insert into register values('sai',9988776655,9632)

1 row created.

SQL> /
Enter value for name: DSai
Enter value for phno: 9876598765
Enter value for pin: 8521
old 1: insert into register values('&name',&phno,&pin)
new 1: insert into register values('DSai',9876598765,8521)

1 row created.

SQL> /
Enter value for name: gera
Enter value for phno: 7036218996
Enter value for pin: 1478
old 1: insert into register values('&name',&phno,&pin)
new 1: insert into register values('gera',7036218996,1478)

1 row created.

SQL> _
```

Update_balance

```
Run SQL Command Line

SQL> insert into update_balance values(&phno,&amt);
Enter value for phno: 6305424006
Enter value for amt: 100
old 1: insert into update_balance values(&phno,&amt)
new 1: insert into update_balance values(6305424006,100)

1 row created.

SQL> update update_balance set amt=amt+&sum where phno=&phno;
Enter value for sum: 6305424006
Enter value for phno: 62
old 1: update update_balance set amt=amt+&sum where phno=&phno
new 1: update update_balance set amt=amt+6305424006 where phno=62

0 rows updated.

SQL> update update_balance set amt=amt+&sum where phno=&phno;
Enter value for sum: 65
Enter value for phno: 6305424006
old 1: update update_balance set amt=amt+&sum where phno=&phno
new 1: update update_balance set amt=amt+65 where phno=6305424006

1 row updated.

SQL> insert into update_balance values(&phno,&amt);
Enter value for phno: 9988776655
Enter value for amt: 30
old 1: insert into update_balance values(&phno,&amt)
new 1: insert into update_balance values(9988776655,30)

1 row created.

SQL> update update_balance set amt=amt+&sum where phno=&phno;
Enter value for sum: 98
Enter value for phno: 9988776655
old 1: update update_balance set amt=amt+&sum where phno=&phno
new 1: update update_balance set amt=amt+98 where phno=9988776655

1 row updated.

SQL> select * from update_balance;

      PHNO      AMT
-----
6305424006    165
9988776655    128
```

Booking_petrol

```
Run SQL Command Line

SQL> insert into booking_petrol values(&phno,&amt);
Enter value for phno: 6305424006
Enter value for amt: 30
old 1: insert into booking_petrol values(&phno,&amt)
new 1: insert into booking_petrol values(6305424006,30)

1 row created.

SQL> update update_balance set amt=amt-&sum where phno=&phno;
Enter value for sum: 30
Enter value for phno: 6305424006
old 1: update update_balance set amt=amt-&sum where phno=&phno
new 1: update update_balance set amt=amt-30 where phno=6305424006

1 row updated.

SQL>
SQL> insert into booking_petrol values(&phno,&amt);
Enter value for phno: 9988776655
Enter value for amt: 60
old 1: insert into booking_petrol values(&phno,&amt)
new 1: insert into booking_petrol values(9988776655,60)

1 row created.

SQL> update update_balance set amt=amt-&sum where phno=&phno;
Enter value for sum: 60
Enter value for phno: 9988776655
old 1: update update_balance set amt=amt-&sum where phno=&phno
new 1: update update_balance set amt=amt-60 where phno=9988776655

1 row updated.

SQL> select * from booking_petrol;

      PHNO      AMT
-----
6305424006      30
9988776655      60

SQL> select * from update_balance;

      PHNO      AMT
-----
6305424006     105
9988776655      68

SQL> insert into booking_diesel values(&phno,&amt);
Enter value for phno: 6305424006
Enter value for amt: 50
old 1: insert into booking_diesel values(&phno,&amt)
new 1: insert into booking_diesel values(6305424006,50)

1 row created.
```

Booking_diesel

```
SQL> update update_balance set amt=amt-&sum where phno=&phno;
Enter value for sum: 50
Enter value for phno: 6305424006
old 1: update update_balance set amt=amt-&sum where phno=&phno
new 1: update update_balance set amt=amt-50 where phno=6305424006

1 row updated.

SQL> insert into booking_diesel values(&phno,&amt);
Enter value for phno: 9988776655
Enter value for amt: 60
old 1: insert into booking_diesel values(&phno,&amt)
new 1: insert into booking_diesel values(9988776655,60)

1 row created.

SQL> update update_balance set amt=amt-&sum where phno=&phno;
Enter value for sum: 60
Enter value for phno: 9988776655
old 1: update update_balance set amt=amt-&sum where phno=&phno
new 1: update update_balance set amt=amt-60 where phno=9988776655

1 row updated.

SQL> insert into booking_diesel values(&phno,&amt);
Enter value for phno: 6305424006
Enter value for amt: 50
old 1: insert into booking_diesel values(&phno,&amt)
new 1: insert into booking_diesel values(6305424006,50)

1 row created.

SQL> update update_balance set amt=amt-&sum where phno=&phno;
Enter value for sum: 50
Enter value for phno: 6305424006
old 1: update update_balance set amt=amt-&sum where phno=&phno
new 1: update update_balance set amt=amt-50 where phno=6305424006

1 row updated.

SQL> select * from update_balance;

  PHNO      AMT
-----
6305424006      5
9988776655      8

SQL> select * from booking_diesel;

  PHNO      AMT
-----
6305424006     50
9988776655     60
6305424006     50

SQL> _
```

Feedback

```
SQL> insert into feedback values(10,8185051364,'super',5,'NA');
1 row created.

SQL> insert into feedback values(10,9052234383,'super',5,'NA');
1 row created.
```

```
SQL> select * from feedback;

      SNO      PHNO
-----
EXPERIENCE
-----
NOOFSTARS
-----
COMPLAINT
-----
super      10 8185051364
NA          5
      SNO      PHNO
-----
EXPERIENCE
-----
NOOFSTARS
-----
COMPLAINT
-----
super      10 9052234383
NA          5
```

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 void connectToDB()
```



```

    {
        try
        {
            connection
            DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","swetha","manage
            r"); statement = connection.createStatement();

        }
        catch (SQLException connectException)
        {
            System.out.println(connectException.getMessage())
            ;
            System.out.println(connectException.getSQLState()
            );
            System.out.println(connectException.getErrorC
            ode()); System.exit(1);
        }
    }
}

```

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

Insert bunk:

```

package petrolbunkAutomationSystem;

import java.sql.SQLException;
import java.sql.*;
import javax.swing.*;
import java.awt.*;
import java.awt.Color;
import java.awt.FlowLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.JFrame;

public class Insert_Bunk extends JFrame{

    private static final long serialVersionUID = 1L;
    JPanel jp1,jp2,jp3;
    Connection con;
    int i;
    java.sql.Statement stmt;
    JLabel city,area,name;
    JTextField bcity,barea,bname;
    TextArea ta;
    JButton in;
}

```

```

public Insert_Bunk()
{
    try
    {
        Class.forName("oracle.jdbc.driver.OracleDriver");

        con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","swetha"
, "manager");

        stmt=con.createStatement();

    }

    catch (Exception e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }

    city=new JLabel("City Name");
    bcity=new JTextField(30);
    area=new JLabel("Area Name");
    barea=new JTextField(30);
    name=new JLabel("Bunk Name");
    bname=new JTextField(30);

    ta=new TextArea(10,40);
    in=new JButton("submit");
    getContentPane().setBackground(Color.blue);
    jp1=new JPanel(new GridLayout(5,1));
    jp2=new JPanel(new FlowLayout());
    jp3=new JPanel(new FlowLayout());
    jp1.add(city);
    jp1.add(bcity);
    jp1.add(area);
    jp1.add(barea);
    jp1.add(name);
    jp1.add(bname);
    jp2.add(in);
    jp3.add(ta);
    add(jp1);
    add(jp2);
    add(jp3);
    setVisible(true);
    setSize(500,600);
    setTitle("Enter following details:");
    setLayout(new GridLayout(3,2));
    pack();

    in.addActionListener(new ActionListener() {

        @Override
        public void actionPerformed(ActionEvent arg0) {
            // TODO Auto-generated method stub
            try {

```

```

                                i=stmt.executeUpdate("insert into bunk
values('"+bcity.getText()+"','"+barea.getText()+"','"+bname.getText()+"')");
                                }
                                catch (SQLException e) {
                                    // TODO Auto-generated catch block
                                    e.printStackTrace();
                                }

                                ta.append("\n Inserted "+i+" rows successfully");

                            }
                        });
                    }
                }
            }
        }
    }
}

```

Update bunk:

```

package petrolbunkAutomationSystem;

import java.awt.FlowLayout;
import java.awt.GridLayout;
import java.awt.List;
import java.awt.TextArea;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.ItemEvent;
import java.awt.event.ItemListener;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JTextField;

public class Update_Bunk extends JFrame{
    private static final long serialVersionUID = 1L;
    JPanel jp1,jp2,jp3;
    Connection con;
    int i;
    ResultSet rs;
    String sel;
    java.sql.Statement stmt;
    JLabel city,area,name;
    JTextField bcity,barea,bname;
    TextArea ta;
    JButton in;
    List lis;

    public Update_Bunk()
    {

```

```

    try {

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

        con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","swetha"
,"manager");

        stmt=con.createStatement();

    }

    catch (Exception e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }

    city=new JLabel("City Name");
    bcity=new JTextField(30);
    area=new JLabel("Area Name");
    barea=new JTextField(30);
    name=new JLabel("Bunk Name");
    bname=new JTextField(30);

    ta=new TextArea(10,40);
    in=new JButton("submit");
    jp1=new JPanel(new GridLayout(5,1));
    jp2=new JPanel(new FlowLayout());
    jp3=new JPanel(new FlowLayout());
    jp1.add(city);
    jp1.add(bcity);
    jp1.add(area);
    jp1.add(barea);
    jp1.add(name);
    jp1.add(bname);
    jp2.add(in);
    jp3.add(ta);
    lis=new List();
    add(jp1);
    add(jp2);
    add(jp3);
    add(lis);
    setVisible(true);
    setSize(500,600);
    setTitle("Enter following details:");
    setLayout(new GridLayout(3,2));
    pack();

    try {
        rs=stmt.executeQuery("select area from bunk");
        while(rs.next()) {
            lis.add(rs.getString(1));
        }
    }
    catch (SQLException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}

```

```

lis.addItemListener(new ItemListener() {
    @Override
    public void itemStateChanged(ItemEvent arg0) {
        // TODO Auto-generated method stub
        try {
            sel=lis.getSelectedItem();
            rs=stmt.executeQuery("select city,area,name from
bunk where area='"+lis.getSelectedItem()+"");
            if(rs.next()) {
                bcity.setText(rs.getString(1));
                barea.setText(rs.getString(2));
                bname.setText(rs.getString(3));
            }
        }
        catch (SQLException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }
});
in.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent arg0) {
        // TODO Auto-generated method stub
        try {
            i=stmt.executeUpdate(" update bunk set
name='"+bname.getText()+"' where city='"+bcity.getText()+"'and
area='"+barea.getText()+"' ");
        }
        catch (SQLException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
        ta.append("\nUpdated "+i+" rows successfully");
    }
});
}
}

```

Delete Bunk:

```

package petrolbunkAutomationSystem;

import java.awt.FlowLayout;
import java.awt.GridLayout;
import java.awt.List;
import java.awt.TextArea;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.ItemEvent;
import java.awt.event.ItemListener;

```

```

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JTextField;

public class Delete_Bunk extends JFrame{
    private static final long serialVersionUID = 1L;
    JPanel jp1,jp2,jp3;
    Connection con;
    int i;
    ResultSet rs;
    String sel;
    java.sql.Statement stmt;
    JLabel city,area,name;
    JTextField bcity,barea,bname;
    TextArea ta;
    JButton in;
    List lis;

    public Delete_Bunk()
    {
        try {
            Class.forName("oracle.jdbc.driver.OracleDriver");

            con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","swetha"
,"manager");

            stmt=con.createStatement();
        }
        catch (Exception e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }

        city=new JLabel("City Name");
        bcity=new JTextField(30);
        area=new JLabel("Area Name");
        barea=new JTextField(30);
        name=new JLabel("Bunk Name");
        bname=new JTextField(30);

        ta=new TextArea(10,40);
        in=new JButton("submit");
        jp1=new JPanel(new GridLayout(5,1));
        jp2=new JPanel(new FlowLayout());
        jp3=new JPanel(new FlowLayout());

        jp1.add(city);
        jp1.add(bcity);
        jp1.add(area);
        jp1.add(barea);

```

```

jp1.add(name);
jp1.add(bname);
jp2.add(in);
jp3.add(ta);
lis=new List();
add(jp1);
add(jp2);
add(jp3);
add(lis);
setVisible(true);
setSize(500,600);
setTitle("Enter following details:");
setLayout(new GridLayout(3,2));
pack();

try {
    rs=stmt.executeQuery("select area from bunk");
    while(rs.next()) {
        lis.add(rs.getString(1));
    }
}
catch (SQLException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
}

lis.addItemListener(new ItemListener() {
    @Override
    public void itemStateChanged(ItemEvent arg0) {
        // TODO Auto-generated method stub
        try {
            sel=lis.getSelectedItem();
            rs=stmt.executeQuery("select city,area,name from
bunk where area='"+lis.getSelectedItem()+"'");
            if(rs.next()) {
                bcity.setText(rs.getString(1));
                barea.setText(rs.getString(2));
                bname.setText(rs.getString(3));
            }
        }
        catch (SQLException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }
});

in.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent arg0) {
        // TODO Auto-generated method stub
        try {
            i=stmt.executeUpdate("delete from bunk where
city='"+bcity.getText()+"'and area='"+barea.getText()+"' ");

```

```

    }
    catch (SQLException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
    ta.append("\n Deleted "+i+" rows successfully");
}
});
}
}
}

```

Main Program:

```

package petrolbunkAutomationSystem;

import java.awt.Color;
import java.awt.FlowLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.JFrame;
import javax.swing.JMenu;
import javax.swing.JMenuBar;
import javax.swing.JMenuItem;

public class Frame extends JFrame {

    private static final long serialVersionUID = 1L;
    JMenuBar menubar;
    JMenu menu1,menu2,menu3,menu4,menu5,menu6,menu7,menu8;
    JMenuItem item1,item2,item3,item4,item5,item6,item7,item8;
    JMenuItem up1,up2,up3,up4,up5,up6,up7,up8,d11,d12,d13,d14,d15,d16,d17,d18;

    public void firstframe() {

        menubar=new JMenuBar();
        menu1=new JMenu("Bunk details");
        menu2=new JMenu("Petrol Price");
        menu3=new JMenu("Diesel Price");
        menu4=new JMenu("SignUp");
        menu5=new JMenu("Update My Balance");
        menu6=new JMenu("Book Petrol");
        menu7=new JMenu("Book Diesel");
        menu8=new JMenu("Feedback");

        item1=new JMenuItem("Insert");
        up1=new JMenuItem("Update");
        d11=new JMenuItem("Delete");
        item2=new JMenuItem("Insert");
        up2=new JMenuItem("Update");
        d12=new JMenuItem("Delete");
        item3=new JMenuItem("Insert");
        up3=new JMenuItem("Update");
        d13=new JMenuItem("Delete");
        item4=new JMenuItem("Insert");
    }
}

```



```

up4=new JMenuItem("Update");
dl4=new JMenuItem("Delete");
item5=new JMenuItem("Insert");
up5=new JMenuItem("Update");
dl5=new JMenuItem("Delete");
item6=new JMenuItem("Insert");
up6=new JMenuItem("Update");
dl6=new JMenuItem("Delete");
item7=new JMenuItem("Insert");
up7=new JMenuItem("Update");
dl7=new JMenuItem("Delete");
item8=new JMenuItem("Insert");
up8=new JMenuItem("Update");
dl8=new JMenuItem("Delete");

getContentPane().setBackground(Color.blue);
setVisible(true);
setSize(500,600);
setTitle("PetrolBunk Automation System");
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
setLayout(new FlowLayout());
setJMenuBar(menubar);
menubar.add(menu1);
menubar.add(menu2);
menubar.add(menu3);
menubar.add(menu4);
menubar.add(menu5);
menubar.add(menu6);
menubar.add(menu7);
menubar.add(menu8);
menu1.add(item1);
menu1.add(up1);
menu1.add(dl1);
menu2.add(item2);
menu2.add(up2);
menu2.add(dl2);
menu3.add(item3);
menu3.add(up3);
menu3.add(dl3);
menu4.add(item4);
menu4.add(up4);
menu4.add(dl4);
menu5.add(item5);
menu5.add(up5);
menu5.add(dl5);
menu6.add(item6);
menu6.add(up6);
menu6.add(dl6);
menu7.add(item7);
menu7.add(up7);
menu7.add(dl7);
menu8.add(item8);
menu8.add(up8);
menu8.add(dl8);

```

```

item1.addActionListener(new ActionListener() {

@Override
    public void actionPerformed(ActionEvent e) {
        // TODO Auto-generated method stub
        new Insert_Bunk();
    }

});

up1.addActionListener(new ActionListener() {

@Override
    public void actionPerformed(ActionEvent arg0) {
        // TODO Auto-generated method stub
        new Update_Bunk();
    }

});

dl1.addActionListener(new ActionListener() {

@Override
    public void actionPerformed(ActionEvent e) {
        // TODO Auto-generated method stub
        new Delete_Bunk();
    }

});

item2.addActionListener(new ActionListener() {

@Override
    public void actionPerformed(ActionEvent e) {
        // TODO Auto-generated method stub
        new Insert_PetrolPrice();
    }

});

up2.addActionListener(new ActionListener() {

@Override
    public void actionPerformed(ActionEvent arg0) {
        // TODO Auto-generated method stub
        new Update_PetrolPrice();
    }

});

dl2.addActionListener(new ActionListener() {

@Override
    public void actionPerformed(ActionEvent e) {

```

```

        // TODO Auto-generated method stub
        new Delete_PetrolPrice();
    }
});

item3.addActionListener(new ActionListener() {

@Override
    public void actionPerformed(ActionEvent e) {
        // TODO Auto-generated method stub
        new Insert_DieselPrice();
    }
});

up3.addActionListener(new ActionListener() {

@Override
    public void actionPerformed(ActionEvent arg0) {
        // TODO Auto-generated method stub
        new Update_DieselPrice();
    }
});

dl3.addActionListener(new ActionListener() {

@Override
    public void actionPerformed(ActionEvent e) {
        // TODO Auto-generated method stub
        new Delete_DieselPrice();
    }
});

item4.addActionListener(new ActionListener() {

@Override
    public void actionPerformed(ActionEvent e) {
        // TODO Auto-generated method stub
        new Insert_Register();
    }
});

up4.addActionListener(new ActionListener() {

@Override
    public void actionPerformed(ActionEvent arg0) {
        // TODO Auto-generated method stub
        new Update_Register();
    }
});

dl4.addActionListener(new ActionListener() {

```

```

@Override
    public void actionPerformed(ActionEvent e) {
        // TODO Auto-generated method stub
        new Delete_Register();
    }
});

item5.addActionListener(new ActionListener() {

@Override
    public void actionPerformed(ActionEvent e) {
        // TODO Auto-generated method stub
        new Insert_UpdateBalance();
    }
});

up5.addActionListener(new ActionListener() {

@Override
    public void actionPerformed(ActionEvent arg0) {
        // TODO Auto-generated method stub
        new Update_UpdateBalance();
    }
});

dl5.addActionListener(new ActionListener() {

@Override
    public void actionPerformed(ActionEvent e) {
        // TODO Auto-generated method stub
        new Delete_UpdateBalance();
    }
});

item6.addActionListener(new ActionListener() {

@Override
    public void actionPerformed(ActionEvent e) {
        // TODO Auto-generated method stub
        new Insert_BookPetrol();
    }
});

up6.addActionListener(new ActionListener() {

@Override
    public void actionPerformed(ActionEvent arg0) {
        // TODO Auto-generated method stub
        new Update_BookPetrol();
    }
});

```

```

});

dl6.addActionListener(new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent e) {
// TODO Auto-generated method stub
        new Delete_BookPetrol();

    }

});

item7.addActionListener(new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent e) {
// TODO Auto-generated method stub
        new Insert_BookDiesel();

    }

});

up7.addActionListener(new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent arg0) {
// TODO Auto-generated method stub
        new Update_BookDiesel();

    }

});

dl7.addActionListener(new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent e) {
// TODO Auto-generated method stub
        new Delete_BookDiesel();

    }

});

item8.addActionListener(new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent e) {
// TODO Auto-generated method stub
        new Insert_Feedback();

    }

});

up8.addActionListener(new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent arg0) {
// TODO Auto-generated method stub

```

```

        new Update_Feedback();
    }
});

dl8.addActionListener(new ActionListener() {

    @Override
    public void actionPerformed(ActionEvent arg0) {
        // TODO Auto-generated method stub
        new Delete_Feedback();
    }

});
}

public static void main(String[] args) {

    Frame f = new Frame();
    f.firstframe();

}

}

```

GITHUH LINK:

<https://github.com/Swethashub-source/PetrolBunkAutomationSystem>

TESTING

The phone number given by the user while registering should be exactly ten digit and cannot insert into the table if its number of digits are greater or less than 10.

The pin number given by the user while registration should be only 4 digit and cannot be accepted otherwise.

Register:

Name	Renu
Phone Number	96325874
Enter pin	1234

Inserted 0 rows successfully

Register: [minimize] [maximize] [close]

Name	urt
Phone Number	1234567894
Enter pin	124

Inserted 0 rows successfully

Register:

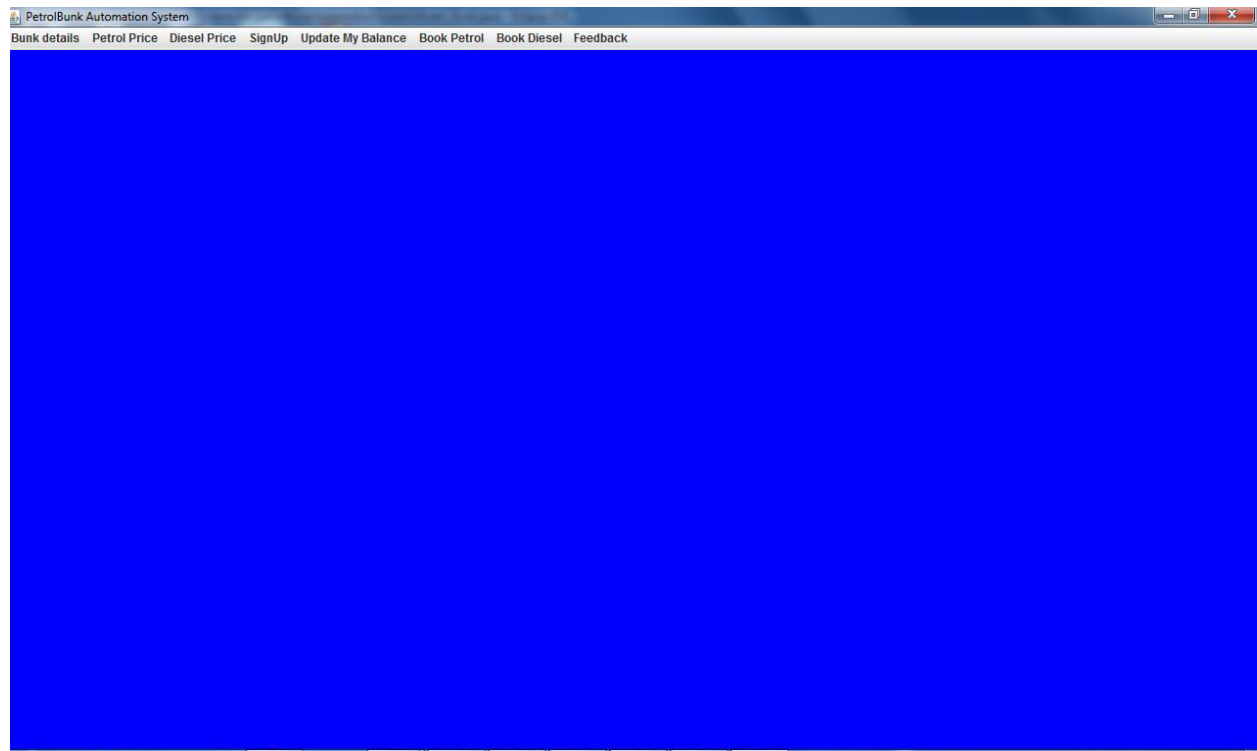
Name	Renu
Phone Number	98745632144
Enter pin	4125

Inserted 0 rows successfully

RESULTS

The main page:

It displays the title of the project and it has menu items which are named after the five tables. Each menu item provides three options that can be performed on that particular table: Insert, Update and Delete.



Bunk Table:

Enter following details:

City Name	Vijayawada
Area Name	RailwayStation
Bunk Name	IndianPetroleumLtd.

submit

Inserted 1 rows successfully

Enter following details:

City Name	Vijayawada	<input type="button" value="submit"/>
Area Name	PVP	
Bunk Name	PVP Pvt Petroleum	

Updated 1 rows successfully

Hyderguda
PVP
RailwayStation
asnagar
Gurjapur
asnagar

Enter following details:

City Name	Vijayawada	<input type="button" value="submit"/>
Area Name	PVP	
Bunk Name	PVP Pvt Petroleum	

Deleted 1 rows successfully

Hyderguda
pvp
RailwayStation
asnagar
Gurjapur
asnagar

Petrol_price Table:

Insert Petrol Price:

City Name	Vijayawada
Area Name	Benz Circle
Date	03-sep-21
Price	57

submit

Inserted 1 rows successfully

Update Petrol Price:

City Name: Vijayawada

Area Name: Benz Circle

Date: 03-Sep-21

Price: 75

submit

Updated 1 rows successfully

city	area	date	price
Vijayawada	BenzCircle	03-Sep-01	1000
Hyderabad	Charminar	23-Aug-03	200
Vijayawada	BenzCircle	03-Dec-21	32
Vijayawada	Benz Circle	03-Sep-21	57
Vijayawada	Tata Nagar	03-Dec-21	56

Delete Petrol Price:

City Name: Vijayawada

Area Name: Benz Circle

Date: 03-Sep-21

Price: 75

submit

Deleted 1 rows successfully

city	area	date	price
Vijayawada	BenzCircle	03-Sep-01	1000
Hyderabad	Charminar	23-Aug-03	200
Vijayawada	BenzCircle	03-Dec-21	32
Vijayawada	Benz Circle	03-Sep-21	75
Vijayawada	Tata Nagar	03-Dec-21	56

Diesel_price Table:

Insert Diesel Price:

City Name	Warangal
Area Name	Rythu Bazar
Date	06-jul-20
Price	76

Inserted 1 rows successfully

Update Diesel Price:

City Name: Warangal

Area Name: Rythu Bazar

Date: 06-Jul-20

Price: 67

submit

Updated 1 rows successfully

city	area	date	price
Warangal	Rythu Bazar	06-Jul-20	76
Hyderabad	BenzCircle	03-Sep-01	100

Enter following details:

City Name: Warangal

Area Name: Rythu Bazar

Date: 06-Jul-20

Price: 67

submit

Deleted 1 rows successfully

city	area	date	price
Warangal	Rythu Bazar	06-Jul-20	67
Hyderabad	BenzCircle	03-Sep-01	100

Register Table:

Swetha Sai Dakupati
1602-19-737-093

Register:

Name	Sunny
Phone Number	9182996030
Enter pin	3654

submit

Inserted 1 rows successfully

Update Registered Account:

Name	Sunny	Update Name	Update Pin
Phone Number	9182996030		
Enter pin	3657		

Updated 1 rows successfully with new pin

6305424006
8185051364
9182996030

Update Registered Account:

Name	SunnyD	Update Name	Update Pin
Phone Number	9182996030		
Enter pin	3657		

Updated 1 rows successfully with new name

6305424006
8185051364
9182996030

Delete Registered Account:

Name	SunnyD	<input type="button" value="submit"/>
Phone Number	9182996030	
Enter pin	3657	

Deleted 1 rows successfully

6305424006

8185051364

9182996030

Update_balance Table:

Insert Into Update Balance:

Phone Number

6305424006

Enter amount

100

submit

Inserted 1 rows successfully

Update Balance:

Phone Number

8185051364

Enter amount

20

Submit

phno	amount
6305424006	100
8185051364	100

Delete the Balance: min max close

Phone Number Submit

6305424006

Enter amount

50|

phno	amount
6305424006	100
8185051364	120

Booking_petrol Table:

Insert into Petrol Booking:

Phone Number

6305424006

Enter amount

100|

submit

Update Booking Petrol: [-] [Maximize] [X]

Phone Number Submit

6305424006

Enter amount

50

Updated 1 rows successfully.

phno	amount
6305424006	100

Delete Boboking Petrol:

Phone Number

6305424006

Enter amount

50

Submit

Deleted 1 rows successfully.

phno	amount
6305424006	50

Booking_diesel Table:

Insert into Booking Diesel:

Phone Number

6305424006

Enter amount

100

submit

Inserted 1 rows successfully

Update Diesel Booking:

Phone Number

8185051364

Enter amount

20

Updated 1 rows successfully..

phno	amount
8185051364	100
6305424006	100

Submit

Delete Diesel Booking:

Phone Number

6305424006

Enter amount

100

Deleted 1 rows successfully..

phno	amount
8185051364	100
6305424006	100

Submit

Feedback Table:

Insert into Feedback:

Serial No:

2

Registered phone number

8185051364

Experience

ok

Number Of Stars

4

Any Complaints

no

Inserted 1 rows successfully

submit

sno	phno	Experience	Stars	Complaints
1	9632587412	lol	5	no

Update your Feedback:

Serial No:

2

Registered phone number

8185051364

Experience

nice

Number Of Stars

4

Any Complaints

no

updated your given experience successfully

UpadteMyExperience

UpdateMyStars

UpdateMyComplaint

sno	phno	Experience	Stars	Complaints
1	9632587412	lol	5	no
2	8185051364	nice	4	no

Delete your feedback

Serial No: 2

Registered phone number: 8185051364

Experience: ok

Number Of Stars: 4

Any Complaints: no

Deleted 1 rows successfully

Delete

sno	phno	Experience	Stars	Complaints
1	9632587412	lol	5	no
2	8185051364	ok	4	no

DISCUSSION AND FUTURE WORK

So far this project has helped the user to know the price of petrol and diesel per litre on a specific day. It can further be improvised so that all his transactions are tracked and even a message sending for each of his transaction.

On further, it should even help the petrol bunk management by helping them with tracking clients, details and transactions.

REFERENCES

1. Oracle Docs
2. Tutorialspoint
3. Geeksforgeeks
4. Youtube

THANK YOU