

**ELECTRICITY BILL MANAGEMENT SYSTEM**

**A MINI-PROJECT REPORT**

**Submitted by**

**B.NIKITHA 230701211**

**D.PRATHEBA 230701239**

**K.RAGAVI 230701249**

**In partial fulfillment of the award of the degree**

**of**

**BACHELOR OF ENGINEERING**

**IN**

**COMPUTER SCIENCE AND ENGINEERING**



**RAJALAKSHMI ENGINEERING COLLEGE, CHENNAI**

**An Autonomous Institute**

**CHENNAI**

**NOVEMBER 2023**

**BONAFIDE CERTIFICATE**

Certified that this project "ELECTRICITY BILL MANAGEMENT SYSTEM" is the bonafide work of "B NIKITHA, D PRATHEBA,K RAGAVI " who carried out the project work under my supervision.

SIGNATURE  
SIGNATURE

ASSISTANT PROFESSOR  
ASSISTANT PROFESSOR

Dept. of Computer Science and Engg,  
Science and Engg,

Rajalakshmi Engineering College  
Engineering College  
chennai

Dept. of Computer

Rajalakshmi

Chennai

**INTERNAL EXAMINER**

**EXTERNAL EXAMINER**

This mini project report is submitted for the viva voce examination to be held on

\_\_\_\_\_

# TABLE OF CONTENTS

S.NO		TITLE
		Abstract
1		Introduction
	05	
2		Scope of Project
	06	
3		UML Diagrams
	09	
4		Code Implementation
	22	
5		Conclusion
	31	

6	Future Work
33	
7	Reference
35	

## ABSTRACT

The Electricity Bill Management System represents an innovative solution aimed at revolutionizing the traditional approach to managing utility billing processes. This project presents a comprehensive analysis and design of a database-driven system tailored to streamline and enhance the efficiency of electricity billing operations for both customers and administrators. The system aims to cover key aspects of an electricity billing system through fundamental database concepts. These include facilitating user registration, secure authentication and also reminding

customers regarding due bills via an intuitive interface. Additionally, it involves designing a structured database schema capable of storing essential billing data such as customer information, meter readings, and bill amounts. The system also focuses on a straightforward bill generation process, whether through predefined rates or manual inputs. Further, it aims to incorporate basic payment tracking functionalities for record-keeping purposes. The focus of this project is to introduce fundamental concepts of database design, implementation, and management in the context of an electricity billing system.

# INTRODUCTION

'Electricity Bill Management System' is a one-step solution for streamlined and hassle-free online electricity bill payments. The Electricity Billing System is an automated system that is developed using JAVA, and MySQL database. This Electricity Billing System has two sections: the client section and the admin section. The admin plays a vital role in all of the management of the Electricity Billing System and this web application is tailored to redefine the experience in managing and settling electricity bills seamlessly. Paying bills such as electricity bills can take a lot of time. Most especially when people need to fall in a long line in order to pay their bill. Hence, it's an advantage to have an automated Electricity Billing System, it's not just very helpful but it will also lessen the stress of many people because they don't have to be there at the outlet just to pay the bill. All they have to do is just go to the website register for a client's account then they can have access into the system. As they logged in as a client , the client now has the ability to view their bill for the month then, they can also send a complaint regarding their bill. While in the admin section, they have the ability to issue a billing statement to the client and make a response whenever there are complaints, is meticulously designed to offer an intuitive and user-friendly interface.

Client Section:

Upon initiation, clients are ushered into an intuitive portal, providing an effortless avenue to peruse monthly bills and submit queries. Beyond being a mere transactional platform, it serves as a secure online gateway, simplifying the payment process with a user-friendly dashboard. This platform not only redefines bill payments but transforms them into a streamlined and secure experience.

#### Admin Section:

The administrative facet of the system empowers administrators to issue billing statements and promptly address client concerns. Through meticulous design and functionality, the admin interface becomes an integral component in ensuring the overall efficiency of the Electricity Billing System.

## **Key Features:**

**1. User Authentication System:** Elevating user experience, a robust authentication system prioritizes data confidentiality and security, guaranteeing a secure online environment.

**2. Comprehensive Dashboard:** Clients are greeted with a visually appealing and responsive

dashboard, ensuring a seamless experience for bill reviews and secure transactions.

**3. Smart Features:** Integration of intelligent features, including detailed bill history and personalized notifications, enhances user engagement and promotes proactive bill management.

**4. Database Efficiency:** Anchored in a well-structured database schema, the system establishes

efficient relationships between users, bills, and transactions, ensuring meticulous record-keeping and streamlined data retrieval.

The "Electricity Bill Management System" stands as a testament to the evolution of automated billing solutions. Beyond a mere platform for financial transactions, it embodies a commitment to user-centric design, data security, and streamlined management.



This innovation, complemented by comprehensive documentation and unwavering customer support, heralds a new era in electricity bill management, promising a future marked by efficiency, convenience, and security.

## SCOPE OF THE PROJECT

The management and payment of electricity bills constitute a fundamental aspect of modern living. In today's fast-paced world, traditional methods of bill payment are becoming increasingly inconvenient and time-consuming. Moreover, with the surge in digital transactions and the need for efficient utility management, there's a pressing need for a streamlined and user-friendly solution for handling electricity bills. The emergence of digital technologies has opened doors for innovative solutions to simplify the process of managing utility payments. Recognizing these

challenges and opportunities, the development of an "Electricity Bill Management System" becomes pivotal in addressing the shortcomings of traditional bill payment methods.

The primary goal behind the creation of the Electricity Bill Management System is to alleviate the complexities and inefficiencies associated with conventional bill payment processes. These challenges include long queues at payment centers, manual handling of bills, potential errors in data entry, and a lack of real-time information for consumers.

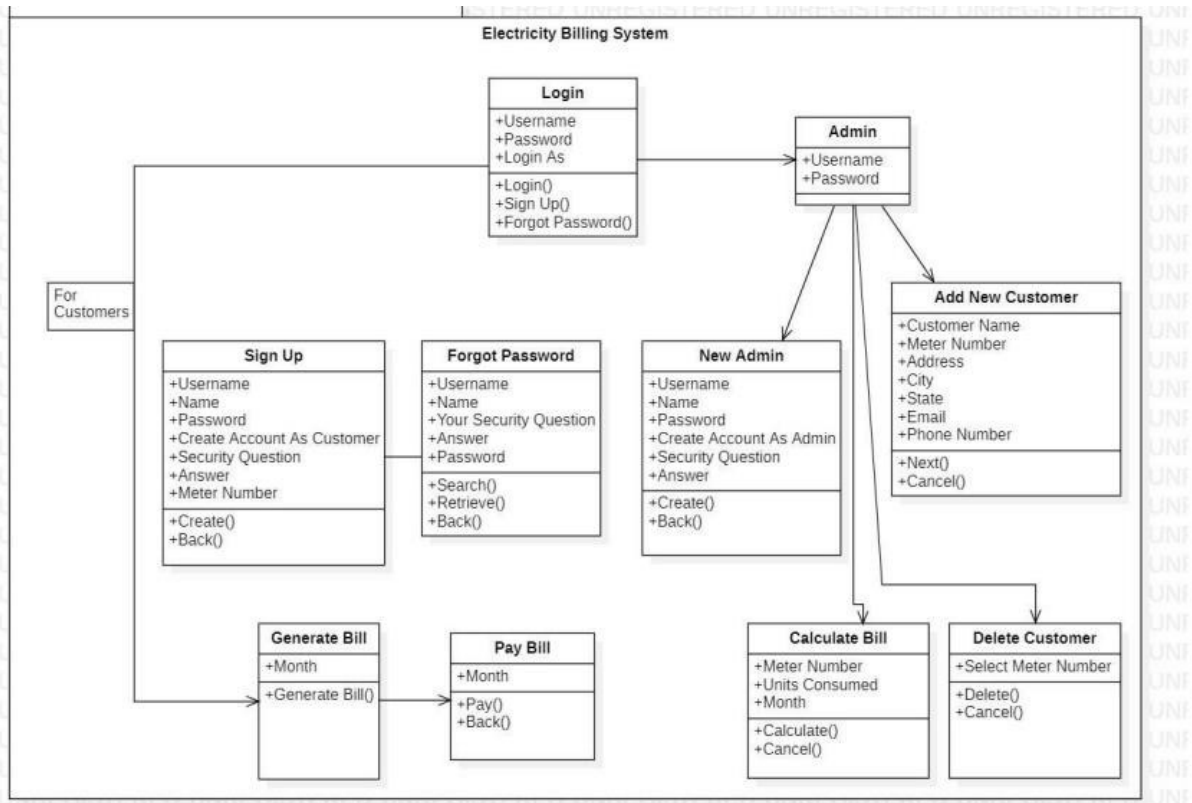
The scope section of an "Electricity Bill Management System" report outlines the boundaries, functionalities, and limitations of the system. It defines what the system encompasses and what aspects it does not cover.

1. **User Registration and Authentication:** The system includes provisions for users to register accounts securely and authenticate their identities to access the platform.

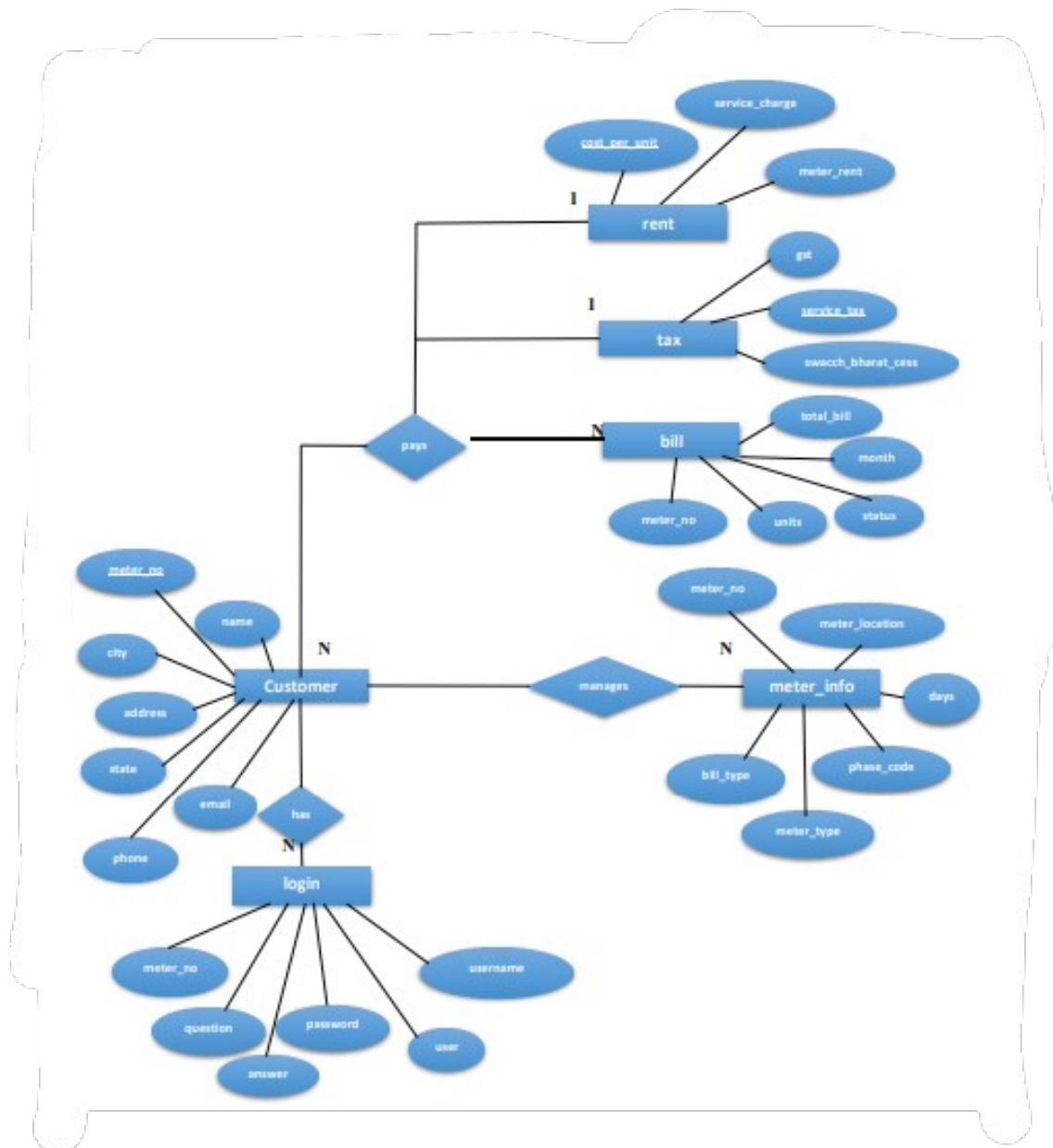
2. **Bill Viewing and Payment:** Users can view detailed billing information, including consumption history, due dates, and payment options. The system facilitates secure online payments for electricity bills.

3. Notification and Alert System: An integrated notification system alerts users about upcoming bill due dates, payment confirmations, and other relevant updates.
4. Secure Database Management: The system maintains a robust database structure to manage.
5. User-Friendly Interface: The platform boasts an intuitive and responsive user interface, ensuring ease of navigation and accessibility across various devices.
6. Security Measures: Stringent security protocols are implemented to safeguard user data and financial transactions, including encryption, secure authentication, and protection against common web vulnerabilities

## UML DIAGRAMS



## Entity Relationship Diagram



## Normalization:

Normalization is a process of organizing the data in database to avoid data redundancy, insertion anomaly, update anomaly & deletion anomaly.

Let's discuss about anomalies first then we will discuss normal forms with examples. Anomalies in DBMS There are three types of anomalies that occur when the database is not normalized. These are –Insertion, update and deletion anomaly.

\***First normal form(1NF)** As per the rule of first normal form, □ All rows must be unique (no duplicate rows).

- Each Cell must only contain a single value (not a list).
- Each value should be non-divisible (can't be split down further) \* **Second normal form(2NF)** As per the rule of second normal form, ✓ Database must be in First Normal Form.

✓ Non partial dependency-All non-prime attributes should be fully functionally dependent on the candidate key

\***Third normal form(3NF)** As per the rule of third normal form, ✓ Database must be in First and Second Normal Form.

✓ Non transitive dependency-All fields must only be determinable by the primary/composite key, not by other keys.

## **PROGRAM:-**

### **Bill details**

```
package electricity_bill.management.system;
```

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

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

```
import java.sql.*;
```

```
import net.proteanit.sql.DbUtils;
```

```

public class BillDetails extends JFrame{

    BillDetails(String meter) {

        setSize(700, 600);
        setLocation(400, 50);

        getContentPane().setBackground(Color.WHITE);

        JTable table = new JTable();

        try {
            Conn c = new Conn();
            String query = "select * from bill where meter_no =
""+meter+""";
            ResultSet rs = c.s.executeQuery(query);

            table.setModel(DbUtils.resultSetToTableModel(rs));
        } catch (Exception e) {
            e.printStackTrace();
        }

        JScrollPane sp = new JScrollPane(table);
        sp.setBounds(0, 0, 700, 650);
        add(sp);

        setVisible(true);
    }
}

```

```

    }

    public static void main(String[] args) {
        new BillDetails("");
    }
}

```

## **Calculate bill**

```
package electricity_bill.management.system;
```

```

import javax.swing.*;
import java.awt.*;
import java.util.*;
import java.awt.event.*;
import java.sql.*;

```

```
public class CalculateBill extends JFrame implements
ActionListener{
```

```

    JTextField tfname, tfaddress, tfstate, tfunits, tfemail, tfphone;
    JButton next, cancel;
    JLabel lblname, labeladdress;
    Choice meternumber, cmonth;
    CalculateBill() {
        setSize(700, 500);
        setLocation(400, 150);
    }
}

```



```
JPanel p = new JPanel();  
p.setLayout(null);  
p.setBackground(new Color(173, 216, 230));  
add(p);
```

```
JLabel heading = new JLabel("Calculate Electricity Bill");  
heading.setBounds(100, 10, 400, 25);  
heading.setFont(new Font("Tahoma", Font.PLAIN, 24));  
p.add(heading);
```

```
JLabel lblmeternumber = new JLabel("Meter Number");  
lblmeternumber.setBounds(100, 80, 100, 20);  
p.add(lblmeternumber);
```

```
meternumber = new Choice();
```

```
try {  
    Conn c = new Conn();  
    ResultSet rs = c.s.executeQuery("select * from  
customer");  
    while(rs.next()) {  
        meternumber.add(rs.getString("meter_no"));  
    }  
} catch (Exception e) {  
    e.printStackTrace();  
}
```

```
meternumber.setBounds(240, 80, 200, 20);
```

```
p.add(meternumber);
```

```
JLabel lblmeterno = new JLabel("Name");  
lblmeterno.setBounds(100, 120, 100, 20);  
p.add(lblmeterno);
```

```
lblname = new JLabel("");  
lblname.setBounds(240, 120, 100, 20);  
p.add(lblname);
```

```
JLabel lbladdress = new JLabel("Address");  
lbladdress.setBounds(100, 160, 100, 20);  
p.add(lbladdress);
```

```
labeladdress = new JLabel();  
labeladdress.setBounds(240, 160, 200, 20);  
p.add(labeladdress);
```

```
try {  
    Conn c = new Conn();  
    ResultSet rs = c.s.executeQuery("select * from  
customer          where          meter_no          =  
"+meternumber.getSelectedItem()+"");  
    while(rs.next()) {  
        lblname.setText(rs.getString("name"));  
        labeladdress.setText(rs.getString("address"));  
    }  
} catch (Exception e) {
```

```

        e.printStackTrace();
    }

    meternumber.addItemListener(new ItemListener() {
        public void itemStateChanged(ItemEvent ie) {
            try {
                Conn c = new Conn();
                ResultSet rs = c.s.executeQuery("select * from
customer          where          meter_no          =
"+meternumber.getSelectedItem()+"");
                while(rs.next()) {
                    lblname.setText(rs.getString("name"));
                    labeladdress.setText(rs.getString("address"));
                }
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    });

    JLabel lblcity = new JLabel("Units Consumed");
    lblcity.setBounds(100, 200, 100, 20);
    p.add(lblcity);

    tfunits = new JTextField();
    tfunits.setBounds(240, 200, 200, 20);
    p.add(tfunits);

```

```
JLabel lblstate = new JLabel("Month");  
lblstate.setBounds(100, 240, 100, 20);  
p.add(lblstate);
```

```
cmonth = new Choice();  
cmonth.setBounds(240, 240, 200, 20);  
cmonth.add("January");  
cmonth.add("February");  
cmonth.add("March");  
cmonth.add("April");  
cmonth.add("May");  
cmonth.add("June");  
cmonth.add("July");  
cmonth.add("August");  
cmonth.add("September");  
cmonth.add("October");  
cmonth.add("November");  
cmonth.add("December");  
p.add(cmonth);
```

```
next = new JButton("Submit");  
next.setBounds(120, 350, 100,25);  
next.setBackground(Color.BLACK);  
next.setForeground(Color.WHITE);  
next.addActionListener(this);  
p.add(next);
```

```

cancel = new JButton("Cancel");
cancel.setBounds(250, 350, 100,25);
cancel.setBackground(Color.BLACK);
cancel.setForeground(Color.WHITE);
cancel.addActionListener(this);
p.add(cancel);

setLayout(new BorderLayout());

add(p, "Center");

                                ImagemIcon    i1    =    new
ImagemIcon(ClassLoader.getResource("icon/hicon2.jpg"));
        Image i2 = i1.getImage().getScaledInstance(150, 300,
Image.SCALE_DEFAULT);
        ImagemIcon i3 = new ImagemIcon(i2);
        JLabel image = new JLabel(i3);
        add(image, "West");

getContentPane().setBackground(Color.WHITE);

setVisible(true);
}

public void actionPerformed(ActionEvent ae) {
    if (ae.getSource() == next) {
        String meter = meternumber.getSelectedItems();
        String units = tfunits.getText();

```

```

String month = cmonth.getSelectedItem();

int totalbill = 0;
int unit_consumed = Integer.parseInt(units);

String query = "select * from tax";

try {
    Conn c = new Conn();
    ResultSet rs = c.s.executeQuery(query);

    while(rs.next()) {
        totalbill += unit_consumed *
Integer.parseInt(rs.getString("cost_per_unit"));
        totalbill +=
Integer.parseInt(rs.getString("meter_rent"));
        totalbill +=
Integer.parseInt(rs.getString("service_charge"));
        totalbill +=
Integer.parseInt(rs.getString("service_tax"));
        totalbill +=
Integer.parseInt(rs.getString("swacch_bharat_cess"));
        totalbill +=
Integer.parseInt(rs.getString("fixed_tax"));
    }
} catch (Exception e) {
    e.printStackTrace();
}

```

```
String query2 = "insert into bill values('"+meter+"',  
"+month+"', '"+units+"', '"+totalbill+"', 'Not Paid')";
```

```
try {  
    Conn c = new Conn();  
    c.s.executeUpdate(query2);  
  
    JOptionPane.showMessageDialog(null, "Customer Bill  
Updated Successfully");  
    setVisible(false);  
} catch (Exception e) {  
    e.printStackTrace();  
}  
} else {  
    setVisible(false);  
}  
}
```

```
public static void main(String[] args) {  
    new CalculateBill();  
}  
}
```

## **Jdbc connection**

```
package electricity_bill.management.system;
```

```
import java.sql.*;  
public class Conn {  
    Connection c;
```

```

Statement s;
Conn() {
    try {
        c = DriverManager.getConnection("jdbc:mysql:///ebs",
"root", "Nikitha@220305");
        s = c.createStatement();
    } catch (Exception e) {
        e.printStackTrace();
    }
}
}

```

## **Customer details**

```
package electricity_bill.management.system;
```

```

import java.awt.*;
import javax.swing.*;
import java.sql.*;
import net.proteanit.sql.DbUtils;
import java.awt.event.*;

```

```

public class CuspackagetomerDetails extends JFrame
implements ActionListener{
    Choice meternumber, cmonth;
    JTable table;
    JButton search, print;
    CustomerDetails(){
        super("Customer Details")
    }
}

```



```

setSize(800, 500);
setLocation(200, 150);

table = new JTable();

try {
    Conn c = new Conn();
        ResultSet rs = c.s.executeQuery("select * from
customer");

        table.setModel(DbUtils.resultSetToTableModel(rs));
    } catch (Exception e) {
        e.printStackTrace();
    }

JScrollPane sp = new JScrollPane(table);
add(sp);

print = new JButton("Print");
print.addActionListener(this);
add(print, "South");

setVisible(true);
}

public void actionPerformed(ActionEvent ae) {
    try {

```

```

        table.print();
    } catch (Exception e) {
        e.printStackTrace();
    }
}

public static void main(String[] args) {
    new CustomerDetails();
}
}

```

## **Deposit details**

```
package electricity_bill.management.system;
```

```

import java.awt.*;
import javax.swing.*;
import java.sql.*;
import net.proteanit.sql.DbUtils;
import java.awt.event.*;

```

```
public class DepositDetails extends JFrame implements
ActionListener{
```

```
    Choice meternumber, cmonth;
```

```
    JTable table;
```

```
    JButton search, print;
```

```
    DepositDetails(){
```

```

super("Deposit Details");

setSize(700, 600);
setLocation(400, 20);

setLayout(null);
getContentPane().setBackground(Color.WHITE);

JLabel lblmeternumber = new JLabel("Search By Meter
Number");
lblmeternumber.setBounds(20, 20, 150, 20);
add(lblmeternumber);

meternumber = new Choice();
meternumber.setBounds(180, 20, 150, 20);
add(meternumber);

try {
    Conn c = new Conn();
    ResultSet rs = c.s.executeQuery("select * from
customer");
    while(rs.next()) {
        meternumber.add(rs.getString("meter_no"));
    }
} catch (Exception e) {
    e.printStackTrace();
}

```

```
JLabel lblmonth = new JLabel("Search By Month");  
lblmonth.setBounds(400, 20, 100, 20);  
add(lblmonth);
```

```
cmonth = new Choice();  
cmonth.setBounds(520, 20, 150, 20);  
cmonth.add("January");  
cmonth.add("February");  
cmonth.add("March");  
cmonth.add("April");  
cmonth.add("May");  
cmonth.add("June");  
cmonth.add("July");  
cmonth.add("August");  
cmonth.add("September");  
cmonth.add("October");  
cmonth.add("November");  
cmonth.add("December");  
add(cmonth);
```

```
table = new JTable();
```

```
try {  
    Conn c = new Conn();  
    ResultSet rs = c.s.executeQuery("select * from bill");  
  
    table.setModel(DbUtils.resultSetToTableModel(rs));  
} catch (Exception e) {
```

```
        e.printStackTrace();
    }
```

```
JScrollPane sp = new JScrollPane(table);
sp.setBounds(0, 100, 700, 600);
add(sp);
```

```
search = new JButton("Search");
search.setBounds(20, 70, 80, 20);
search.addActionListener(this);
add(search);
```

```
print = new JButton("Print");
print.setBounds(120, 70, 80, 20);
print.addActionListener(this);
add(print);
```

```
setVisible(true);
}
```

```
public void actionPerformed(ActionEvent ae) {
    if (ae.getSource() == search) {
        String query = "select * from bill where meter_no =
"+meternumber.getSelectedItem()+" and month =
"+cmonth.getSelectedItem()+"";

        try {
            Conn c = new Conn();
```

```

        ResultSet rs = c.s.executeQuery(query);
        table.setModel(DbUtils.resultSetToTableModel(rs));
    } catch (Exception e) {

    }

    } else {
        try {
            table.print();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

```

public static void main(String[] args) {
    new DepositDetails();
}
}

```

## **Generate bill**

```
package electricity_bill.management.system;
```

```

import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.sql.*;

```

```
public class GenerateBill extends JFrame implements  
ActionListener{
```

```
    String meter;
```

```
    JButton bill;
```

```
    Choice cmonth;
```

```
    JTextArea area;
```

```
    GenerateBill(String meter) {
```

```
        this.meter = meter;
```

```
        setSize(500, 600);
```

```
        setLocation(500, 30);
```

```
        setLayout(new BorderLayout());
```

```
        JPanel panel = new JPanel();
```

```
        JLabel heading = new JLabel("Generate Bill");
```

```
        JLabel meternumber = new JLabel(meter);
```

```
        cmonth = new Choice();
```

```
        cmonth.add("January");
```

```
        cmonth.add("February");
```

```
        cmonth.add("March");
```

```
        cmonth.add("April");
```

```
        cmonth.add("May");
```

```
        cmonth.add("June");
```

```

cmonth.add("July");
cmonth.add("August");
cmonth.add("September");
cmonth.add("October");
cmonth.add("November");
cmonth.add("December");

area = new JTextArea(50, 15);
    area.setText("\n\n\t-----Click on the-----\n\t Generate
Bill Button to get\n\tthe bill of the Selected Month");
    area.setFont(new Font("Senserif", Font.ITALIC, 18));

JScrollPane pane = new JScrollPane(area);

bill = new JButton("Generate Bill");
bill.addActionListener(this);

panel.add(heading);
panel.add(meternumber);
panel.add(cmonth);
add(panel, "North");

add(pane, "Center");
add(bill, "South");

setVisible(true);
}

```



```

public void actionPerformed(ActionEvent ae) {
    try {
        Conn c = new Conn();

        String month = cmonth.getSelectedItem();

        area.setText("\tReliance Power Limited\nELECTRICITY
BILL GENERATED FOR THE MONTH\n\tOF "+month+", 2022\n\n\
n");

        ResultSet rs = c.s.executeQuery("select * from
customer where meter_no = '"+meter+"'");

        if(rs.next()) {
            area.append("\n      Customer Name: " +
rs.getString("name"));
            area.append("\n      Meter Number   : " +
rs.getString("meter_no"));
            area.append("\n      Address                : " +
rs.getString("address"));
            area.append("\n      City                  : " +
rs.getString("city"));
            area.append("\n      State                  : " +
rs.getString("state"));
            area.append("\n      Email                  : " +
rs.getString("email"));
            area.append("\n      Phone                  : " +
rs.getString("phone"));

            area.append("\
n-----");
            area.append("\n");

```

```
}
```

```
rs = c.s.executeQuery("select * from meter_info where  
meter_no = '"+meter+"'");
```

```
if(rs.next()) {  
    area.append("\n      Meter Location: " +  
rs.getString("meter_location"));  
    area.append("\n      Meter Type:      " +  
rs.getString("meter_type"));  
    area.append("\n      Phase Code:      " +  
rs.getString("phase_code"));  
    area.append("\n      Bill Type:      " +  
rs.getString("bill_type"));  
    area.append("\n      Days:          " +  
rs.getString("days"));  
    area.append("\n-----");  
    area.append("\n");  
}  
  
rs = c.s.executeQuery("select * from tax");  
  
if(rs.next()) {  
    area.append("\n");  
    area.append("\n      Cost Per Unit: " +  
rs.getString("cost_per_unit"));  
    area.append("\n      Meter Rent:    " +  
rs.getString("cost_per_unit"));  
    area.append("\n      Service Charge:  " +  
rs.getString("service_charge"));
```

```

        area.append("\n    Service Tax: " +
rs.getString("service_charge"));

        area.append("\n    Swacch Bharat Cess: " +
rs.getString("swacch_bharat_cess"));

        area.append("\n    Fixed Tax: " +
rs.getString("fixed_tax"));

        area.append("\n");
    }

    rs = c.s.executeQuery("select * from bill where
meter_no = '"+meter+"' and month='"+month+"'");

    if(rs.next()) {
        area.append("\n");

        area.append("\n    Current Month: " +
rs.getString("month"));

        area.append("\n    Units Consumed: " +
rs.getString("units"));

        area.append("\n    Total Charges: " +
rs.getString("totalbill"));

        area.append("\n-----");

        area.append("\n    Total Payable: " +
rs.getString("totalbill"));

        area.append("\n");
    }
} catch (Exception e) {
    e.printStackTrace();
}
}

```

```

        public static void main(String[] args) {
            new GenerateBill("");
        }
    }
}

```

## **Login page**

```

package electricity_bill.management.system;

```

```

import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.sql.*;

```

```

public class Login extends JFrame implements ActionListener{

```

```

    JButton login, cancel, signup;
    JTextField username, password;
    Choice logginin;
    Login() {
        super("Login Page");
        getContentPane().setBackground(Color.WHITE);
        setLayout(null);

        JLabel lblusername = new JLabel("Username");
        lblusername.setBounds(300, 20, 100, 20);
        add(lblusername);
    }
}

```

```
username = new JTextField();  
username.setBounds(400, 20, 150, 20);  
add(username);
```

```
JLabel lblpassword = new JLabel("Password");  
lblpassword.setBounds(300, 60, 100, 20);  
add(lblpassword);
```

```
password = new JTextField();  
password.setBounds(400, 60, 150, 20);  
add(password);
```

```
JLabel loggininas = new JLabel("Loggin in as");  
loggininas.setBounds(300, 100, 100, 20);  
add(loggininas);
```

```
logginin = new Choice();  
logginin.add("Admin");  
logginin.add("Customer");  
logginin.setBounds(400, 100, 150, 20);  
add(logginin);
```

```
ImageIcon i1 = new  
ImageIcon(ClassLoader.getResource("icon/login.png"));  
Image i2 = i1.getImage().getScaledInstance(16, 16,  
Image.SCALE_DEFAULT);  
login = new JButton("Login", new ImageIcon(i2));  
login.setBounds(330, 160, 100, 20);
```

```

login.addActionListener(this);
add(login);

        ImageIcon i3 = new
ImageIcon(ClassLoader.getResource("icon/cancel.jpg"));
        Image i4 = i3.getImage().getScaledInstance(16, 16,
Image.SCALE_DEFAULT);
        cancel = new JButton("Cancel", new ImageIcon(i4));
        cancel.setBounds(450, 160, 100, 20);
        cancel.addActionListener(this);
        add(cancel);

        ImageIcon i5 = new
ImageIcon(ClassLoader.getResource("icon/signup.png"))
;
        Image i6 = i5.getImage().getScaledInstance(16, 16,
Image.SCALE_DEFAULT);
        signup = new JButton("Signup", new ImageIcon(i6));
        signup.setBounds(380, 200, 100, 20);
        signup.addActionListener(this);
        add(signup);

        ImageIcon i7 = new
ImageIcon(ClassLoader.getResource("icon/second.jpg"))
;
        Image i8 = i7.getImage().getScaledInstance(250, 250,
Image.SCALE_DEFAULT);
        ImageIcon i9 = new ImageIcon(i8);
        JLabel image = new JLabel(i9);
        image.setBounds(0, 0, 250, 250);

```

```

add(image);

setSize(640, 300);
setLocation(400, 200);
setVisible(true);
}

public void actionPerformed(ActionEvent ae) {
    if (ae.getSource() == login) {
        String susername = username.getText();
        String spassword = password.getText();
        String user = logginin.getSelectedItem();

        try {
            Conn c = new Conn();

            String query = "select * from login where username
= '"+susername+"' and password = '"+spassword+"' and user
= '"+user+"'";

            ResultSet rs = c.s.executeQuery(query);

            if (rs.next()) {
                String meter = rs.getString("meter_no");
                setVisible(false);
                new Project(user, meter);
            } else {
                JOptionPane.showMessageDialog(null, "Invalid
Login");
                username.setText("");
            }
        }
    }
}

```

```

        password.setText("");
    }

    } catch (Exception e) {
        e.printStackTrace();
    }
} else if (ae.getSource() == cancel) {
    setVisible(false);
} else if (ae.getSource() == signup) {
    setVisible(false);

    new Signup();
}
}

public static void main(String[] args) {
    new Login();
}
}

```

## **Meter information**

```
package electricity_bill.management.system;
```

```

import javax.swing.*;
import java.awt.*;
import java.util.*;
import java.awt.event.*;

```



```
public class MeterInfo extends JFrame implements  
ActionListener{
```

```
    JTextField tfname, tfaddress, tfstate, tfcity, tfemail, tfphone;
```

```
    JButton next, cancel;
```

```
    JLabel lblmeter;
```

```
    Choice meterlocation, metertype, phasecode, billtype;
```

```
    String meternumber;
```

```
    MeterInfo(String meternumber) {
```

```
        this.meternumber = meternumber;
```

```
        setSize(700, 500);
```

```
        setLocation(400, 100);
```

```
        JPanel p = new JPanel();
```

```
        p.setLayout(null);
```

```
        p.setBackground(new Color(173, 216, 230));
```

```
        add(p);
```

```
        JLabel heading = new JLabel("Meter Information");
```

```
        heading.setBounds(180, 10, 200, 25);
```

```
        heading.setFont(new Font("Tahoma", Font.PLAIN, 24));
```

```
        p.add(heading);
```

```
        JLabel lblname = new JLabel("Meter Number");
```

```
        lblname.setBounds(100, 80, 100, 20);
```

```
        p.add(lblname);
```

```
JLabel lblmeternumber = new JLabel(meternumber);  
lblmeternumber.setBounds(240, 80, 100, 20);  
p.add(lblmeternumber);
```

```
JLabel lblmeterno = new JLabel("Meter Location");  
lblmeterno.setBounds(100, 120, 100, 20);  
p.add(lblmeterno);
```

```
meterlocation = new Choice();  
meterlocation.add("Outside");  
meterlocation.add("Inside");  
meterlocation.setBounds(240, 120, 200, 20);  
p.add(meterlocation);
```

```
JLabel lbladdress = new JLabel("Meter Type");  
lbladdress.setBounds(100, 160, 100, 20);  
p.add(lbladdress);
```

```
metertype = new Choice();  
metertype.add("Electric Meter");  
metertype.add("Solar Meter");  
metertype.add("Smart Meter");  
metertype.setBounds(240, 160, 200, 20);  
p.add(metertype);
```

```
JLabel lblcity = new JLabel("Phase Code");  
lblcity.setBounds(100, 200, 100, 20);
```

```
p.add(lblcity);
```

```
phasecode = new Choice();  
phasecode.add("011");  
phasecode.add("022");  
phasecode.add("033");  
phasecode.add("044");  
phasecode.add("055");  
phasecode.add("066");  
phasecode.add("077");  
phasecode.add("088");  
phasecode.add("099");  
phasecode.setBounds(240, 200, 200, 20);  
p.add(phasecode);
```

```
JLabel lblstate = new JLabel("Bill Type");  
lblstate.setBounds(100, 240, 100, 20);  
p.add(lblstate);
```

```
billtype = new Choice();  
billtype.add("Normal");  
billtype.add("Industial");  
billtype.setBounds(240, 240, 200, 20);  
p.add(billtype);
```

```
JLabel lblemail = new JLabel("Days");  
lblemail.setBounds(100, 280, 100, 20);  
p.add(lblemail);
```

```
JLabel lblemails = new JLabel("30 Days");  
lblemails.setBounds(240, 280, 100, 20);  
p.add(lblemails);
```

```
JLabel lblphone = new JLabel("Note");  
lblphone.setBounds(100, 320, 100, 20);  
p.add(lblphone);
```

```
JLabel lblphones = new JLabel("By Default Bill is calculated  
for 30 days only");  
lblphones.setBounds(240, 320, 500, 20);  
p.add(lblphones);
```

```
next = new JButton("Submit");  
next.setBounds(220, 390, 100, 25);  
next.setBackground(Color.BLACK);  
next.setForeground(Color.WHITE);  
next.addActionListener(this);  
p.add(next);
```

```
setLayout(new BorderLayout());
```

```
add(p, "Center");
```

```
ImageIcon i1 = new  
ImageIcon(ClassLoader.getResource("icon/hicon1.jpg"));
```

```
        Image i2 = i1.getImage().getScaledInstance(150, 300,
Image.SCALE_DEFAULT);
```

```
        ImageIcon i3 = new ImageIcon(i2);
```

```
        JLabel image = new JLabel(i3);
```

```
        add(image, "West");
```

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

```
        setVisible(true);
```

```
    }
```

```
    public void actionPerformed(ActionEvent ae) {
```

```
        if (ae.getSource() == next) {
```

```
            String meter = meternumber;
```

```
            String location = meterlocation.getSelectedItem();
```

```
            String type = metertype.getSelectedItem();
```

```
            String code = phasecode.getSelectedItem();
```

```
            String typebill = billtype.getSelectedItem();
```

```
            String days = "30";
```

```
                String query = "insert into meter_info
values('"+meter+"', '"+location+"', '"+type+"', '"+code+"',
 '"+typebill+"', '"+days+"')";
```

```
        try {
```

```
            Conn c = new Conn();
```

```
            c.s.executeUpdate(query);
```

```
        JOptionPane.showMessageDialog(null, "Meter  
Information Added Successfully");
```

```
        setVisible(false);
```

```
    } catch (Exception e) {
```

```
        e.printStackTrace();
```

```
    }
```

```
    } else {
```

```
        setVisible(false);
```

```
    }
```

```
}
```

```
public static void main(String[] args) {
```

```
    new MeterInfo("");
```

```
}
```

```
}
```

## **New customer**

```
package electricity_bill.management.system;
```

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

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

```
import java.util.*;
```

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

```
public class NewCustomer extends JFrame implements  
ActionListener{
```

```

JTextField tfname, tfaddress, tfstate, tfcity, tfemail, tfphone;
JButton next, cancel;
JLabel lblmeter;
NewCustomer() {
    setSize(700, 500);
    setLocation(400, 100);

    JPanel p = new JPanel();
    p.setLayout(null);
    p.setBackground(new Color(173, 216, 230));
    add(p);

    JLabel heading = new JLabel("New Customer");
    heading.setBounds(180, 10, 200, 25);
    heading.setFont(new Font("Tahoma", Font.PLAIN, 24));
    p.add(heading);

    JLabel lblname = new JLabel("Customer Name");
    lblname.setBounds(100, 80, 100, 20);
    p.add(lblname);

    tfname = new JTextField();
    tfname.setBounds(240, 80, 200, 20);
    p.add(tfname);

    JLabel lblmeterno = new JLabel("Meter Number");
    lblmeterno.setBounds(100, 120, 100, 20);
    p.add(lblmeterno);

```

```
lblmeter = new JLabel("");  
lblmeter.setBounds(240, 120, 100, 20);  
p.add(lblmeter);
```

```
Random ran = new Random();  
long number = ran.nextLong() % 1000000;  
lblmeter.setText("" + Math.abs(number));
```

```
JLabel lbladdress = new JLabel("Address");  
lbladdress.setBounds(100, 160, 100, 20);  
p.add(lbladdress);
```

```
tfaddress = new JTextField();  
tfaddress.setBounds(240, 160, 200, 20);  
p.add(tfaddress);
```

```
JLabel lblcity = new JLabel("City");  
lblcity.setBounds(100, 200, 100, 20);  
p.add(lblcity);
```

```
tfcity = new JTextField();  
tfcity.setBounds(240, 200, 200, 20);  
p.add(tfcity);
```

```
JLabel lblstate = new JLabel("State");  
lblstate.setBounds(100, 240, 100, 20);  
p.add(lblstate);
```



```
tfstate = new JTextField();  
tfstate.setBounds(240, 240, 200, 20);  
p.add(tfstate);
```

```
JLabel lblemail = new JLabel("Email");  
lblemail.setBounds(100, 280, 100, 20);  
p.add(lblemail);
```

```
tfemail = new JTextField();  
tfemail.setBounds(240, 280, 200, 20);  
p.add(tfemail);
```

```
JLabel lblphone = new JLabel("Phone Number");  
lblphone.setBounds(100, 320, 100, 20);  
p.add(lblphone);
```

```
tfphone = new JTextField();  
tfphone.setBounds(240, 320, 200, 20);  
p.add(tfphone);
```

```
next = new JButton("Next");  
next.setBounds(120, 390, 100, 25);  
next.setBackground(Color.BLACK);  
next.setForeground(Color.WHITE);  
next.addActionListener(this);  
p.add(next);
```

```

cancel = new JButton("Cancel");
cancel.setBounds(250, 390, 100,25);
cancel.setBackground(Color.BLACK);
cancel.setForeground(Color.WHITE);
cancel.addActionListener(this);
p.add(cancel);

setLayout(new BorderLayout());

add(p, "Center");

                                ImagemIcon    i1    =    new
ImagemIcon(ClassLoader.getResource("icon/hicon1.jpg"));
        Image i2 = i1.getImage().getScaledInstance(150, 300,
Image.SCALE_DEFAULT);
        ImagemIcon i3 = new ImagemIcon(i2);
        JLabel image = new JLabel(i3);
        add(image, "West");

getContentPane().setBackground(Color.WHITE);

setVisible(true);
}

public void actionPerformed(ActionEvent ae) {
    if (ae.getSource() == next) {
        String name = tfname.getText();
        String meter = lblmeter.getText();

```

```

String address = tfaddress.getText();
String city = tfcity.getText();
String state = tfstate.getText();
String email = tfemail.getText();
String phone = tfphone.getText();

String query1 = "insert into customer
values('"+name+"', '"+meter+"', '"+address+"', '"+city+"',
 '"+state+"', '"+email+"', '"+phone+"')";

String query2 = "insert into login values('"+meter+"', ",
 '"+name+"', ", ")";

try {
    Conn c = new Conn();
    c.s.executeUpdate(query1);
    c.s.executeUpdate(query2);

    JOptionPane.showMessageDialog(null, "Customer
Details Added Successfully");
    setVisible(false);

    // new frame
    new MeterInfo(meter);
} catch (Exception e) {
    e.printStackTrace();
}
} else {
    setVisible(false);
}

```

```

    }

    public static void main(String[] args) {
        new NewCustomer();
    }
}

```

## **Main program implementation**

```
package electricity_bill.management.system;
```

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
```

```
public class Project extends JFrame implements ActionListener{
```

```
    String atype, meter;
```

```
    Project(String atype, String meter) {
```

```
        this.atype = atype;
```

```
        this.meter = meter;
```

```
        setExtendedState(JFrame.MAXIMIZED_BOTH);
```

```
        ImageIcon i1 = new
        ImageIcon(ClassLoader.getResource("icon/elect1.jpg"));
```

```
        Image i2 = i1.getImage().getScaledInstance(1550, 850,
        Image.SCALE_DEFAULT);
```

```
        ImageIcon i3 = new ImageIcon(i2);
```

```
        JLabel image = new JLabel(i3);
```

```
        add(image);
```

```

JMenuBar mb = new JMenuBar();
setJMenuBar(mb);

JMenu master = new JMenu("Master");
master.setForeground(Color.BLUE);

JMenuItem newcustomer = new JMenuItem("New
Customer");
        newcustomer.setFont(new Font("monospaced",
Font.PLAIN, 12));
        newcustomer.setBackground(Color.WHITE);
        ImageIcon icon1 = new
ImageIcon(ClassLoader.getResource("icon/icon1.png"));
        Image image1 = icon1.getImage().getScaledInstance(20,
20, Image.SCALE_DEFAULT);
        newcustomer.setIcon(new ImageIcon(image1));
        newcustomer.setMnemonic('D');
        newcustomer.addActionListener(this);
        newcustomer.setAccelerator(KeyStroke.getKeyStroke(Key
Event.VK_D, ActionEvent.CTRL_MASK));
        master.add(newcustomer);

JMenuItem customerdetails = new JMenuItem("Customer
Details");
        customerdetails.setFont(new Font("monospaced",
Font.PLAIN, 12));
        customerdetails.setBackground(Color.WHITE);

```

```

        ImageIcon icon2 = new
        ImageIcon(ClassLoader.getResource("icon/icon2.png"));
        Image image2 = icon2.getImage().getScaledInstance(20,
        20, Image.SCALE_DEFAULT);
        customerdetails.setIcon(new ImageIcon(image2));
        customerdetails.setMnemonic('M');
        customerdetails.addActionListener(this);
        customerdetails.setAccelerator(KeyStroke.getKeyStroke(K
        eyEvent.VK_M, ActionEvent.CTRL_MASK));
        master.add(customerdetails);

        JMenuItem depositdetails = new JMenuItem("Deposit
        Details");
        depositdetails.setFont(new Font("monospaced",
        Font.PLAIN, 12));
        depositdetails.setBackground(Color.WHITE);
        ImageIcon icon3 = new
        ImageIcon(ClassLoader.getResource("icon/icon3.png"));
        Image image3 = icon3.getImage().getScaledInstance(20,
        20, Image.SCALE_DEFAULT);
        depositdetails.setIcon(new ImageIcon(image3));
        depositdetails.setMnemonic('N');
        depositdetails.addActionListener(this);
        depositdetails.setAccelerator(KeyStroke.getKeyStroke(Key
        Event.VK_N, ActionEvent.CTRL_MASK));
        master.add(depositdetails);

        JMenuItem calculatebill = new JMenuItem("Calculate Bill");
        calculatebill.setFont(new Font("monospaced", Font.PLAIN,
        12));

```

```

        calculatebill.setBackground(Color.WHITE);

        ImageIcon icon4 = new
        ImageIcon(ClassLoader.getResource("icon/icon5.png"));
        Image image4 = icon4.getImage().getScaledInstance(20,
        20, Image.SCALE_DEFAULT);
        calculatebill.setIcon(new ImageIcon(image4));
        calculatebill.setMnemonic('B');
        calculatebill.addActionListener(this);
        calculatebill.setAccelerator(KeyStroke.getKeyStroke(KeyEv
        ent.VK_B, ActionEvent.CTRL_MASK));
        master.add(calculatebill);

        JMenu info = new JMenu("Information");
        info.setForeground(Color.RED);

```

```

        JMenuItem updateinformation = new JMenuItem("Update
        Information");
        updateinformation.setFont(new Font("monospaced",
        Font.PLAIN, 12));
        updateinformation.setBackground(Color.WHITE);

        ImageIcon icon5 = new
        ImageIcon(ClassLoader.getResource("icon/icon4.png"));
        Image image5 = icon5.getImage().getScaledInstance(20,
        20, Image.SCALE_DEFAULT);
        updateinformation.setIcon(new ImageIcon(image5));
        updateinformation.setMnemonic('P');
        updateinformation.addActionListener(this);

```

```
        updateinformation.setAccelerator(KeyStroke.getKeyStroke  
(KeyEvent.VK_P, ActionEvent.CTRL_MASK));  
        info.add(updateinformation);
```

```
        JMenuItem viewinformation = new JMenuItem("View  
Information");
```

```
        viewinformation.setFont(new Font("monospaced",  
Font.PLAIN, 12));
```

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

```
                ImageIcon icon6 = new  
ImageIcon(ClassLoader.getResource("icon/icon6.png"));
```

```
        Image image6 = icon6.getImage().getScaledInstance(20,  
20, Image.SCALE_DEFAULT);
```

```
        viewinformation.setIcon(new ImageIcon(image6));
```

```
        viewinformation.setMnemonic('L');
```

```
        viewinformation.addActionListener(this);
```

```
        viewinformation.setAccelerator(KeyStroke.getKeyStroke(K  
eyEvent.VK_L, ActionEvent.CTRL_MASK));
```

```
        info.add(viewinformation);
```

```
JMenu user = new JMenu("User");
```

```
user.setForeground(Color.BLUE);
```

```
JMenuItem paybill = new JMenuItem("Pay Bill");
```

```
paybill.setFont(new Font("monospaced", Font.PLAIN, 12));
```

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

```
                ImageIcon icon7 = new  
ImageIcon(ClassLoader.getResource("icon/icon4.png"));
```



```
        Image image7 = icon7.getImage().getScaledInstance(20,
20, Image.SCALE_DEFAULT);
        paybill.setIcon(new ImageIcon(image7));
        paybill.setMnemonic('R');
        paybill.addActionListener(this);
        paybill.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_R,
ActionEvent.CTRL_MASK));
        user.add(paybill);
```

```
        JMenuItem billdetails = new JMenuItem("Bill Details");
        billdetails.setFont(new Font("monospaced", Font.PLAIN,
12));
        billdetails.setBackground(Color.WHITE);
```

```
                ImageIcon icon8 = new
ImageIcon(ClassLoader.getResource("icon/icon6.png"));
```

```
        Image image8 = icon8.getImage().getScaledInstance(20,
20, Image.SCALE_DEFAULT);
```

```
        billdetails.setIcon(new ImageIcon(image8));
        billdetails.setMnemonic('B');
        billdetails.addActionListener(this);
        billdetails.setAccelerator(KeyStroke.getKeyStroke(KeyEven
t.VK_B, KeyEvent.CTRL_MASK));
        user.add(billdetails);
```

```
JMenu report = new JMenu("Report");
report.setForeground(Color.RED);
```

```
JMenuItem generatebill = new JMenuItem("Generate Bill");
```

```

        generatebill.setFont(new Font("monospaced", Font.PLAIN,
12));
        generatebill.setBackground(Color.WHITE);

        ImageIcon icon9 = new
ImageIcon(ClassLoader.getResource("icon/icon7.png"));
        Image image9 = icon9.getImage().getScaledInstance(20,
20, Image.SCALE_DEFAULT);
        generatebill.setIcon(new ImageIcon(image9));
        generatebill.setMnemonic('G');
        generatebill.addActionListener(this);
        generatebill.setAccelerator(KeyStroke.getKeyStroke(KeyEv
ent.VK_G, ActionEvent.CTRL_MASK));
        report.add(generatebill);

```

```

JMenu utility = new JMenu("Utility");
utility.setForeground(Color.BLUE);

```

```

JMenuItem notepad = new JMenuItem("Notepad");
        notepad.setFont(new Font("monospaced", Font.PLAIN,
12));
        notepad.setBackground(Color.WHITE);

        ImageIcon icon10 = new
ImageIcon(ClassLoader.getResource("icon/icon12.png"))
;
        Image image10 =
icon10.getImage().getScaledInstance(20, 20,
Image.SCALE_DEFAULT);
        notepad.setIcon(new ImageIcon(image10));

```

```

notepad.setMnemonic('N');
notepad.addActionListener(this);
    notepad.setAccelerator(KeyStroke.getKeyStroke(KeyEvent
.VK_N, ActionEvent.CTRL_MASK));
utility.add(notepad);

JMenuItem calculator = new JMenuItem("Calculator");
    calculator.setFont(new Font("monospaced", Font.PLAIN,
12));
    calculator.setBackground(Color.WHITE);

                                ImageIcon icon11 = new
ImageIcon(ClassLoader.getResource("icon/icon9.png"));
                                Image image11 =
icon11.getImage().getScaledInstance(20, 20,
Image.SCALE_DEFAULT);
    calculator.setIcon(new ImageIcon(image10));
    calculator.setMnemonic('C');
    calculator.addActionListener(this);
    calculator.setAccelerator(KeyStroke.getKeyStroke(KeyEve
nt.VK_C, ActionEvent.CTRL_MASK));
utility.add(calculator);

JMenu mexit = new JMenu("Exit");
mexit.setForeground(Color.RED);

JMenuItem exit = new JMenuItem("Exit");
exit.setFont(new Font("monospaced", Font.PLAIN, 12));
exit.setBackground(Color.WHITE);

```

```

        ImageIcon icon12 = new
        ImageIcon(ClassLoader.getResource("icon/icon11.png"))
        ;

        Image image12 =
        icon12.getImage().getScaledInstance(20, 20,
        Image.SCALE_DEFAULT);

        exit.setIcon(new ImageIcon(image12));
        exit.setMnemonic('W');
        exit.addActionListener(this);
        exit.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_
        W, ActionEvent.CTRL_MASK));
        mexit.add(exit);

        if (atype.equals("Admin")) {
            mb.add(master);
        } else {
            mb.add(info);
            mb.add(user);
            mb.add(report);
        }

        mb.add(utility);
        mb.add(mexit);

        setLayout(new FlowLayout());

        setVisible(true);
    }

```

```

public void actionPerformed(ActionEvent ae) {
    String msg = ae.getActionCommand();
    if (msg.equals("New Customer")) {
        new NewCustomer();
    } else if (msg.equals("Customer Details")) {
        new CustomerDetails();
    } else if (msg.equals("Deposit Details")) {
        new DepositDetails();
    } else if (msg.equals("Calculate Bill")) {
        new CalculateBill();
    } else if (msg.equals("View Information")) {
        new ViewInformation(meter);
    } else if (msg.equals("Update Information")) {
        new UpdateInformation(meter);
    } else if (msg.equals("Bill Details")) {
        new BillDetails(meter);
    } else if (msg.equals("Notepad")) {
        try {
            Runtime.getRuntime().exec("notepad.exe");
        } catch (Exception e) {
            e.printStackTrace();
        }
    } else if (msg.equals("Calculator")) {
        try {
            Runtime.getRuntime().exec("calc.exe");
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

```

    }
    } else if (msg.equals("Exit")) {
        setVisible(false);
        new Login();
    } else if (msg.equals("Pay Bill")) {
        new PayBill(meter);
    } else if (msg.equals("Generate Bill")) {
        new GenerateBill(meter);
    }
}

```

```

public static void main(String[] args) {
    new Project("", "");
}
}

```

## **Signup page**

```
package electricity_bill.management.system;
```

```

import javax.swing.*;
import javax.swing.border.*;
import java.awt.*;
import java.awt.event.*;
import java.sql.*;

```

```
public class Signup extends JFrame implements ActionListener{
```

```
    JButton create, back;
```

```

Choice accountType;
JTextField meter, username, name, password;
Signup(){

    setBounds(450, 150, 700, 400);
    getContentPane().setBackground(Color.WHITE);
    setLayout(null);

    JPanel panel = new JPanel();
    panel.setBounds(30, 30, 650, 300);
    panel.setBorder(new TitledBorder(new LineBorder(new
Color(173, 216, 230), 2), "Create-Account",
TitledBorder.LEADING, TitledBorder.TOP, null, new Color(172,
216, 230)));
    panel.setBackground(Color.WHITE);
    panel.setLayout(null);
    panel.setForeground(new Color(34, 139, 34));
    add(panel);

    JLabel heading = new JLabel("Create Account As");
    heading.setBounds(100, 50, 140, 20);
    heading.setForeground(Color.GRAY);
    heading.setFont(new Font("Tahoma", Font.BOLD, 14));
    panel.add(heading);

    accountType = new Choice();
    accountType.add("Admin");
    accountType.add("Customer");
    accountType.setBounds(260, 50, 150, 20);

```

```
panel.add(accountType);
```

```
JLabel lblmeter = new JLabel("Meter Number");  
lblmeter.setBounds(100, 90, 140, 20);  
lblmeter.setForeground(Color.GRAY);  
lblmeter.setFont(new Font("Tahoma", Font.BOLD, 14));  
panel.add(lblmeter);
```

```
meter = new JTextField();  
meter.setBounds(260, 90, 150, 20);  
meter.setVisible(false);  
panel.add(meter);
```

```
JLabel lblusername = new JLabel("Username");  
lblusername.setBounds(100, 130, 140, 20);  
lblusername.setForeground(Color.GRAY);  
lblusername.setFont(new Font("Tahoma", Font.BOLD, 14));  
panel.add(lblusername);
```

```
username = new JTextField();  
username.setBounds(260, 130, 150, 20);  
panel.add(username);
```

```
JLabel lblname = new JLabel("Name");  
lblname.setBounds(100, 170, 140, 20);  
lblname.setForeground(Color.GRAY);  
lblname.setFont(new Font("Tahoma", Font.BOLD, 14));  
panel.add(lblname);
```



```

name = new JTextField();
name.setBounds(260, 170, 150, 20);
panel.add(name);

meter.addFocusListener(new FocusListener() {
    @Override
    public void focusGained(FocusEvent fe) {}

    @Override
    public void focusLost(FocusEvent fe) {
        try {
            Conn c = new Conn();
            ResultSet rs = c.s.executeQuery("select * from
login where meter_no = '"+meter.getText()+"'");
            while(rs.next()) {
                name.setText(rs.getString("name"));
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
});

```

```

JLabel lblpassword = new JLabel("Password");
lblpassword.setBounds(100, 210, 140, 20);
lblpassword.setForeground(Color.GRAY);

```

```
lblpassword.setFont(new Font("Tahoma", Font.BOLD, 14));  
panel.add(lblpassword);
```

```
password = new JTextField();  
password.setBounds(260, 210, 150, 20);  
panel.add(password);
```

```
accountType.addItemListener(new ItemListener() {  
    public void itemStateChanged(ItemEvent ae) {  
        String user = accountType.getSelectedItem();  
        if (user.equals("Customer")) {  
            lblmeter.setVisible(true);  
            meter.setVisible(true);  
            name.setEditable(false);  
        } else {  
            lblmeter.setVisible(false);  
            meter.setVisible(false);  
            name.setEditable(true);  
        }  
    }  
});
```

```
create = new JButton("Create");  
create.setBackground(Color.BLACK);  
create.setForeground(Color.WHITE);  
create.setBounds(140, 260, 120, 25);  
create.addActionListener(this);
```

```
panel.add(create);
```

```
back = new JButton("Back");
```

```
back.setBackground(Color.BLACK);
```

```
back.setForeground(Color.WHITE);
```

```
back.setBounds(300, 260, 120, 25);
```

```
back.addActionListener(this);
```

```
panel.add(back);
```

```
ImageIcon i1 = new  
ImageIcon(ClassLoader.getResource("icon/signupImage.  
png"));
```

```
Image i2 = i1.getImage().getScaledInstance(250, 250,  
Image.SCALE_DEFAULT);
```

```
ImageIcon i3 = new ImageIcon(i2);
```

```
JLabel image = new JLabel(i3);
```

```
image.setBounds(415, 30, 250, 250);
```

```
panel.add(image);
```

```
setVisible(true);
```

```
}
```

```
public void actionPerformed(ActionEvent ae) {
```

```
    if (ae.getSource() == create) {
```

```
        String atype = accountType.getSelectedItem();
```

```
        String susername = username.getText();
```

```
        String sname = name.getText();
```

```
        String spassword = password.getText();
```

```

String smeter = meter.getText();

try {
    Conn c = new Conn();

    String query = null;
    if (atype.equals("Admin")) {
        query = "insert into login values('"+smeter+"',
"+susername+"',      '"+sname+"',      '"+spassword+"',
"+atype+"')";
    } else {
        query = "update login set username =
"+susername+"', password = '"+spassword+"', user =
"+atype+" where meter_no = '"+smeter+"'";
    }
    c.s.executeUpdate(query);

    JOptionPane.showMessageDialog(null, "Account
Created Successfully");

    setVisible(false);
    new Login();
} catch (Exception e) {
    e.printStackTrace();
}
} else if (ae.getSource() == back) {
    setVisible(false);

    new Login();
}

```

```

    }
}

public static void main(String[] args) {
    new Signup();
}
}


```

## Output

The image displays two screenshots of a Java Swing application. The top window, titled "Create-Account", features a light blue header bar with standard window controls. The main content area has a light gray background. On the left, there are five labels: "Create Account As", "Meter Number", "Username", "Name", and "Password". The "Create Account As" label is followed by a dropdown menu showing "Admin". The other four labels are followed by empty text input fields. To the right of these fields is a large blue silhouette of a person with a red circular arrow pointing right. At the bottom, there are two black buttons labeled "Create" and "Back".

The bottom window, titled "Login Page", also has a light blue header bar. It features a black silhouette of a person in a suit on the left. To the right, there are three labels: "Username", "Password", and "Login in as". The "Username" and "Password" labels are followed by empty text input fields. The "Login in as" label is followed by a dropdown menu showing "Admin". At the bottom, there are three buttons: "Login" (with a yellow arrow icon), "Cancel" (with a red X icon), and "Signup" (with a blue document icon).

Master Utility Exit



### New Customer

Customer Name

Meter Number 393805

Address

City

State

Email

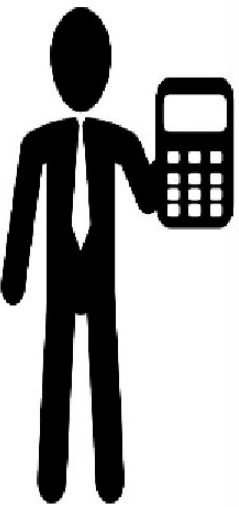
Phone Number

Next Cancel

Customer Details

name	meter_no	address	city	state	email	phone
hulasi	298286	anna nagar food s...	chennai	tamil nadu	thulasi@gmail.com	2345678901
ratheba	817916	19/2 manjakuppam	cuddalore	tamil nadu	sree#15@gmail.com	9876543210
idhanya	977025	ambur	vellore	tn	ridhan@1234	9173395051
iswanalal	268567	asdf	chennai	tn	asdfghJ@Kl	12345667
oshika	647005	mmc	kanchipuram	tn	asdfgghjkl	1234567890
agavi	81170	kanchipuram	manimangalam	tn	ragavi220304	1234567890
osh	18061	no.1	kpm	tamil nadu	joshm30@gmail.c...	6385908456

Print



## Calculate Electricity Bill

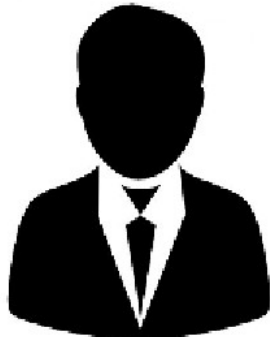
Meter Number

Name

Address

Units Consumed

Month



## Login Page

Username

Password

Loggin in as

VIEW CUSTOMER INFORMATION

Name

thulasi

State

Meter Number

298286

Email

Address


anna nagar foo...

Phone

City

chennai

Cancel



Information User Report Utility Exit

Electricity Bill

Meter Number

298286

Name

thulasi

Month

January

Units

118

Total Bill

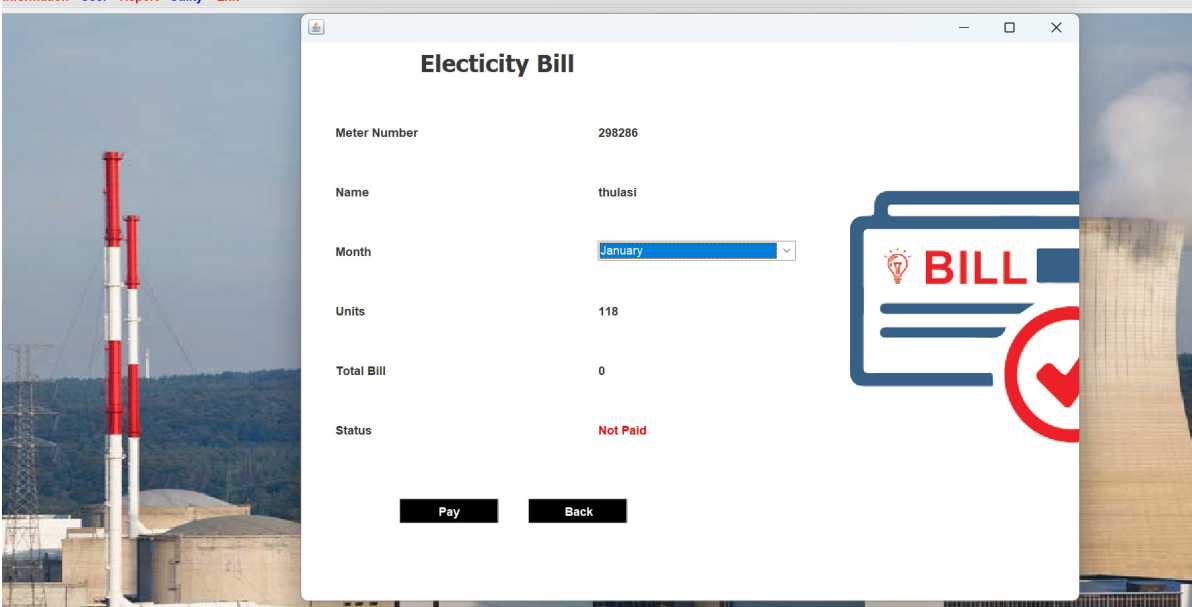
0

Status

Not Paid

Pay

Back





# CONCLUSION

The Electricity Bill Management System efficiently automates the entire billing process, making it easier for administrators to manage large amounts of data and for customers to access their billing information. The project achieves the core objectives of reducing errors, ensuring timely bill generation, and providing secure payment processing. Future improvements could include integration with online payment gateways, mobile support, and enhanced data visualization features for usage tracking

# REFERENCE

**REFERENCES ADD LINKS ,BOOKS,REFERED TO  
DEVELOP THIS PROJECT**

- 1) Silberschatz, Abraham, et al. "Database System Concepts." McGraw-Hill.
- 2) Database Systems: <https://www.geeksforgeeks.org/dbms/>
- 3) **IEEE Xplore** – Search for related articles on "**electricity billing systems**" or "**database management in utility billing**" on IEEE Xplore for detailed case studies and system architectures.

