

ELECTRICITY BILL MANAGEMENT SYSTEM

A MINI PROJECT REPORT

Submitted BY:

MONIC AUDITYA.A230701194

MANICKA MEENAKSHI.S230701173

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-602105

NOVEMBER 2024

BONAFIDE CERTIFICATE

Certified that this project report “**ELECTRICITY BILL MANAGEMENT**” is the Bonafide work of “**MONIC AUDITYA.A AND MANICKA MEENAKSHI.S**” who carried out the project work under my supervision.

Submitted for the practical examination held on _____

SIGNATURE

Mrs.K.MAHESMEENNA

Assistant Professor,
Computer Science and Engineering,
Rajalakshmi Engineering College, Thandalam,
Chennai : 602 105

INTERNAL EXAMINER

EXTERNAL EXAMINER

ABSTRACT:

The **Electricity Bill Management System** is a comprehensive software application designed to facilitate the efficient management of electricity billing and payment processing. Utilizing SQL for data storage, the system ensures data protection, stability, and flexibility, enabling reliable and secure handling of customer and billing information. The system consists of two major interfaces: the user interface for customers and the administrator interface for system management.

The user interface provides customers with a simple and intuitive way to manage their electricity accounts. Customers can view their electricity bills, check the status of payments, and access their personal meter readings. Additionally, the system allows users to print or download their bills and make secure payments via integrated online payment systems, enhancing convenience and ensuring secure transactions.

On the other hand, the administrator interface offers powerful control tools for managing customer accounts and overseeing the entire billing process. Administrators can monitor and allocate meter readings, generate bills based on consumption data, and track payment statuses. This functionality ensures transparency in the billing process and helps identify any overdue or failed payments, promoting efficiency and accountability.

Data security is a top priority in the system, with authentication protocols and data encryption safeguarding sensitive customer information. Advanced SQL queries enable real-time updates on bill generation, payment statuses, and detailed reporting. These features streamline financial transactions, meter reading management, and customer data, resulting in improved operational efficiency and enhanced customer satisfaction.

TABLE OF CONTENTS

1 INTRODUCTION

1.1 OVERVIEW

1.2 TECHNOLOGY STACK

1.3 PROJECT SCOPE

1.4 FEATURES

2 SYSTEM REQUIREMENTS

2.1 HARDWARE

2.2 SOFTWARE

3 E R DIAGRAM

4 SAMPLE CODE & NORMALIZATION

5. VISUAL REPRESENTATIONS

5.1 ADMIN INTERFACE

5.2 USER INTERFACE

6 CONCLUSIONS

7 REFERENCES

INTRODUCTION

1.1 OVERVIEW

The Electricity Bill Management System is a comprehensive software solution, designed to simplify electricity billing and payment processing. It uses SQL for secure data storage. The system consists of two interfaces: user and admin interfaces. Users can get their bill, know the payment status, and view their meter reading in the most secure way.

Administrators will be able to manage customer accounts, generate bills, track meter readings, and track statuses of payments. The system implements data safety with encryption and authentication. Real-time SQL queries allow for efficient billing, payment tracking, and report generation, which shall help improve both operational efficiency and customer satisfaction.

1.2 TECHNOLOGY STACK

- **JavaFX:** Creates an interactive and user-friendly interface.
- **MySQL:** Stores game data and player records.
- **JDBC (Java Database Connectivity):** Facilitates communication between the Java application and the database.

1.3 PROJECT SCOPE

1.The system enables customers to view electricity bills, monitor their payments, check meter readings, and make secure online payments.

2. Administrator Control : Administrators can manage customer accounts, monitor and allocate meter readings, generate bills on consumption basis, and track payment status of each customer.

3.Real-Time Billing and Payment Tracking : The system allows for real-time updates on bill generation, payment status, discrepancies, or overdue payments to promote transparency and accuracy.

4.Data Security and Reporting : The system ensures proper protection of data through encryption and authentication protocols, advancing reporting capacity on billing, payments, and operational performance.

1.4 FEATURES

- Automated Bill Generation
- Produces bills based on meter readings and facilitates secure online payments.
- Customer Account Access
- Enables the access to bill viewings, payment tracking, and meter reading check.
- Admin Control Panel
- Advances account managing, bills generating, and payment status tracking of the administrator.
- Data Security & Real-time Updates
- Holds and processes the data in a secured manner with real-time updates of the billing and payments.

SYSTEM REQUIREMENTS

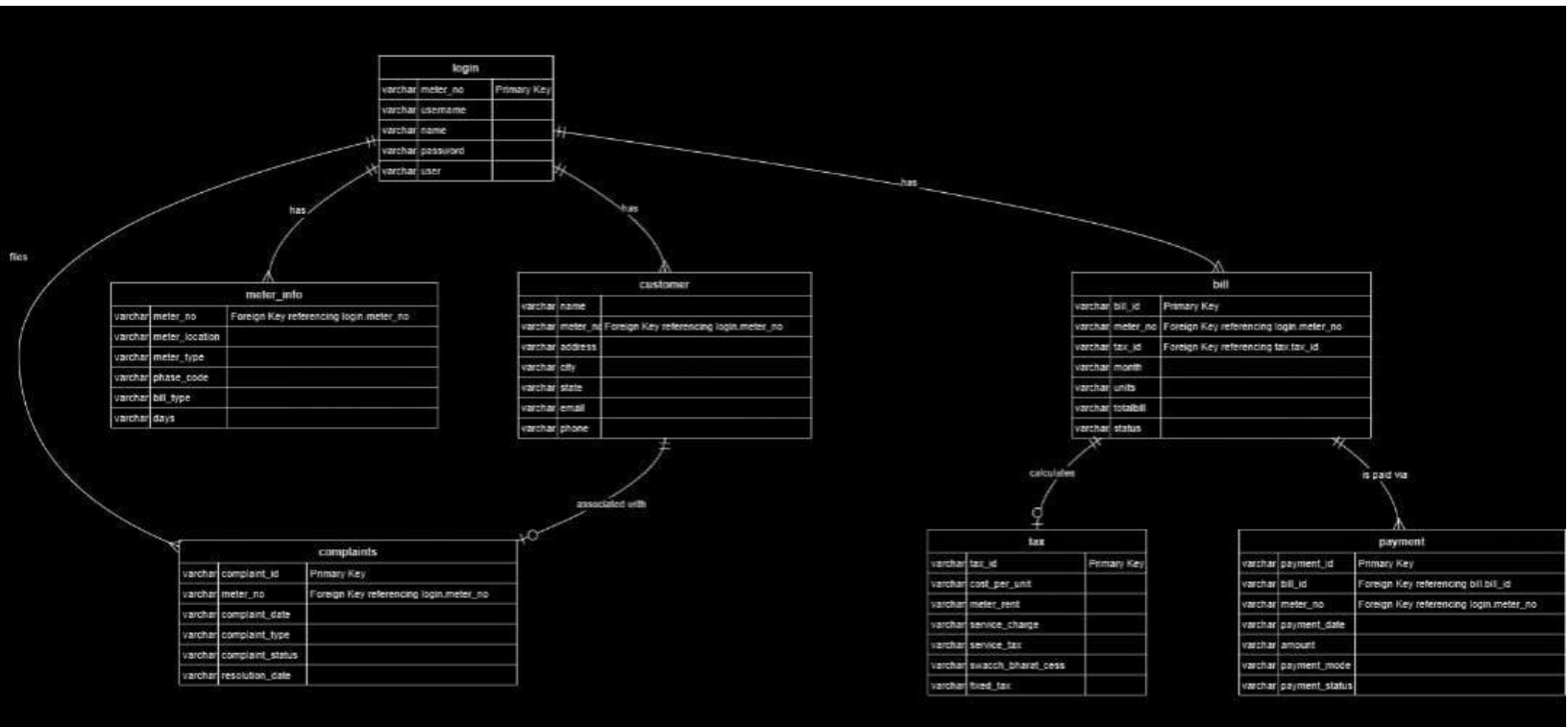
2.1 HARDWARE

- Processor: Intel i5 or equivalent
- Memory: Minimum 4GB RAM
- Hard Disk: 500 GB of free space
- Graphics Card: Capable of supporting JavaFX graphics

2.2 SOFTWARE

- Programming Language: Java
- GUI Library: JavaFX
- Database: MySQL
- Database Connectivity: JDBC (Java Database Connectivity)
- Operating System: Windows 11
- IDE (Integrated Development Environment): NtBeans

ER DIAGRAM



SAMPLE CODE

4.1 Login.

```
package electricity.billing.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("Login 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 = login.getSelectedText();

        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();
}
}

```

4.2 Bill details

```

package electricity.billing.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, 650);
        setLocation(400, 150);

        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("");
    }
}

```

4.3 Calculate Bill Java

```

package electricity.billing.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");

ImageIcon i1 = new ImageIcon(ClassLoader.getResource("icon/hicon2.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.getSelectedItemAt();
        String units = tfunits.getText();
        String month = cmonth.getSelectedItemAt();

        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();
}
}

```

4.4 CONNECTION WITH JAVA

```

package electricity.billing.system;

import java.sql.*;

public class Conn {

    Connection c;
    Statement s;
    Conn() {
        try {
            c = DriverManager.getConnection("jdbc:mysql:///ebs", "root", "12345");
            s = c.createStatement();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

4.5 Customer Details

```

package electricity.billing.system;

import java.awt.*;

```

```

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

public class CustomerDetails extends JFrame implements ActionListener{

    Choice meternumber, cmonth;
    JTable table;
    JButton search, print;

    CustomerDetails(){
        super("Customer Details");

        setSize(1200, 650);
        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();
    }
}

```

4.6 Deposit details

```

package electricity.billing.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, 700);
        setLocation(400, 100);

        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();
}
}

```

4.7Generate Bill

```
package electricity.billing.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, 800);
        setLocation(550, 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("");
    }
}

```

4.8 Meter Information

```

package electricity.billing.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, 200);

        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("Industrial");  
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("");
}
}

```

4.9 New User

```

package electricity.billing.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, 200);  
  
    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);  
  
    JTextField 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);  
  
    JTextField 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);  
  
    JTextField 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);  
  
    JTextField 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");

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 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+"', '1', '"+name+"', '1')";

        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();
}
}

```

4.10 Pay bill

```
package electricity.billing.system;
```

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

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

```

    Choice cmonth;
    JButton pay, back;
    String meter;
    PayBill(String meter) {
        this.meter = meter;
        setLayout(null);
        setBounds(300, 150, 900, 600);

```

```

        JLabel heading = new JLabel("Electricity Bill");
        heading.setFont(new Font("Tahoma", Font.BOLD, 24));
        heading.setBounds(120, 5, 400, 30);
        add(heading);

```

```

        JLabel lblmeternumber = new JLabel("Meter Number");
        lblmeternumber.setBounds(35, 80, 200, 20);
        add(lblmeternumber);

```

```

        JLabel meternumber = new JLabel("");
        meternumber.setBounds(300, 80, 200, 20);
        add(meternumber);

```

```

        JLabel lblname = new JLabel("Name");
        lblname.setBounds(35, 140, 200, 20);
        add(lblname);

```

```

        JLabel labelname = new JLabel("");
        labelname.setBounds(300, 140, 200, 20);

```

```
add(labelname);
```

```
JLabel lblmonth = new JLabel("Month");  
lblmonth.setBounds(35, 200, 200, 20);  
add(lblmonth);
```

```
cmonth = new Choice();  
cmonth.setBounds(300, 200, 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");  
add(cmonth);
```

```
JLabel lblunits = new JLabel("Units");  
lblunits.setBounds(35, 260, 200, 20);  
add(lblunits);
```

```
JLabel labelunits = new JLabel("");  
labelunits.setBounds(300, 260, 200, 20);  
add(labelunits);
```

```
JLabel lbltotalbill = new JLabel("Total Bill");  
lbltotalbill.setBounds(35, 320, 200, 20);  
add(lbltotalbill);
```

```
JLabel labeltotalbill = new JLabel("");  
labeltotalbill.setBounds(300, 320, 200, 20);  
add(labeltotalbill);
```

```
JLabel lblstatus = new JLabel("Status");  
lblstatus.setBounds(35, 380, 200, 20);  
add(lblstatus);
```

```
JLabel labelstatus = new JLabel("");  
labelstatus.setBounds(300, 380, 200, 20);  
labelstatus.setForeground(Color.RED);  
add(labelstatus);
```

```
try {  
    Conn c = new Conn();  
    ResultSet rs = c.s.executeQuery("select * from customer where meter_no = '"+meter+"'");  
    while(rs.next()) {
```

```

        meternumber.setText(meter);
        labelname.setText(rs.getString("name"));
    }

    rs = c.s.executeQuery("select * from bill where meter_no = '"+meter+"' AND month = 'January'");
    while(rs.next()) {
        labelunits.setText(rs.getString("units"));
        labeltotalbill.setText(rs.getString("totalbill"));
        labelstatus.setText(rs.getString("status"));
    }
} catch (Exception e) {
    e.printStackTrace();
}

```

```

cmonth.addItemListener(new ItemListener(){
    @Override
    public void itemStateChanged(ItemEvent ae) {
        try {
            Conn c = new Conn();
            ResultSet rs = c.s.executeQuery("select * from bill where meter_no = '"+meter+"' AND
month = '"+cmonth.getSelectedItem()+"'");
            while(rs.next()) {
                labelunits.setText(rs.getString("units"));
                labeltotalbill.setText(rs.getString("totalbill"));
                labelstatus.setText(rs.getString("status"));
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
});

```

```

pay = new JButton("Pay");
pay.setBackground(Color.BLACK);
pay.setForeground(Color.WHITE);
pay.setBounds(100, 460, 100, 25);
pay.addActionListener(this);
add(pay);

```

```

back = new JButton("Back");
back.setBackground(Color.BLACK);
back.setForeground(Color.WHITE);
back.setBounds(230, 460, 100, 25);
back.addActionListener(this);
add(back);

```

```

getContentPane().setBackground(Color.WHITE);

```

```

ImageIcon i1 = new ImageIcon(ClassLoader.getResource("icon/bill.png"));
Image i2 = i1.getImage().getScaledInstance(600, 300, Image.SCALE_DEFAULT);

```

```

        ImageIcon i3 = new ImageIcon(i2);
        JLabel image = new JLabel(i3);
        image.setBounds(400, 120, 600, 300);
        add(image);

        setVisible(true);
    }

    public void actionPerformed(ActionEvent ae) {
        if (ae.getSource() == pay) {
            try {
                Conn c = new Conn();
                c.s.executeUpdate("update bill set status = 'Paid' where meter_no = '"+meter+"' AND
month='"+cmonth.getSelectedItem()+"'");
            } catch (Exception e) {
                e.printStackTrace();
            }
            setVisible(false);
            new Paytm(meter);
        } else {
            setVisible(false);
        }
    }

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

```

4.11 Paytm

```

package electricity.billing.system;

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

public class Paytm extends JFrame implements ActionListener{

    String meter;
    JButton back;
    Paytm(String meter) {
        this.meter = meter;

        JEditorPane j = new JEditorPane();
        j.setEditable(false);

        try {
            j.setPage("https://paytm.com/online-payments");
        } catch (Exception e) {
            j.setContentType("text/html");
            j.setText("<html>Could not load<html>");
        }
    }
}

```

```

    }

    JScrollPane pane = new JScrollPane(j);
    add(pane);

    back = new JButton("Back");
    back.setBounds(640, 20, 80, 30);
    back.addActionListener(this);
    j.add(back);

    setSize(800, 600);
    setLocation(400, 150);
    setVisible(true);

}

public void actionPerformed(ActionEvent ae) {
    setVisible(false);
    new PayBill(meter);
}

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

```

4.12 Project

```

package electricity.billing.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(KeyEvent.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(KeyEvent.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(KeyEvent.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(KeyEvent.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(KeyEvent.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, KeyEvent.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(KeyEvent.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(KeyEvent.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(image11));
calculator.setMnemonic('C');
calculator.addActionListener(this);
calculator.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.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("", "");
}
}

```

4.13 Signup

```

package electricity.billing.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));
lblmeter.setVisible(false);
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();
    }
}

```

4.14 Splash

```

package electricity.billing.system;
import javax.swing.*.*;
import java.awt.*.*;

```

```

public class Splash extends JFrame implements Runnable {
    Thread t;
    Splash() {

        ImageIcon i1 = new ImageIcon(ClassLoader.getResource("icon/elect.jpg"));
        Image i2 = i1.getImage().getScaledInstance(730, 550, Image.SCALE_DEFAULT);
        ImageIcon i3 = new ImageIcon(i2);
        JLabel image = new JLabel(i3);
        add(image);

        setVisible(true);

        int x = 1;
        for (int i = 2; i < 600; i+=4, x+=1) {
            setSize(i + x, i);
            setLocation(700 - ((i + x)/2), 400 - (i/2));
            try {
                Thread.sleep(5);
            } catch (Exception e) {
                e.printStackTrace();
            }
        }

        t = new Thread(this);
        t.start();

        setVisible(true);
    }

    public void run() {
        try {
            Thread.sleep(7000);
            setVisible(false);

            // login frame
            new Login();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

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

```

4.15 Update Information

```

package electricity.billing.system;
import javax.swing.*;
import java.awt.*;
import java.sql.*;

```

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

public class UpdateInformation extends JFrame implements ActionListener{

    JTextField tfaddress, tfstate, tfcity, tfemail, tfphone;

    JButton update, cancel;

    String meter;

    JLabel name;

    UpdateInformation(String meter) {

        this.meter = meter;

        setBounds(300, 150, 1050, 450);

        getContentPane().setBackground(Color.WHITE);

        setLayout(null);


        JLabel heading = new JLabel("UPDATE CUSTOMER INFORMATION");

        heading.setBounds(110, 0, 400, 30);

        heading.setFont(new Font("Tahoma", Font.PLAIN, 20));

        add(heading);


        JLabel lblname = new JLabel("Name");

        lblname.setBounds(30, 70, 100, 20);

        add(lblname);


        name = new JLabel("");

        name.setBounds(230, 70, 200, 20);

        add(name);


        JLabel lblmeternumber = new JLabel("Meter Number");

        lblmeternumber.setBounds(30, 110, 100, 20);

        add(lblmeternumber);


        JLabel meternumber = new JLabel("");

        meternumber.setBounds(230, 110, 200, 20);
```

```
add(meternumber);
```

```
JLabel lbladdress = new JLabel("Address");  
lbladdress.setBounds(30, 150, 100, 20);  
add(lbladdress);
```

```
tfaddress = new JTextField();  
tfaddress.setBounds(230, 150, 200, 20);  
add(tfaddress);
```

```
JLabel lblcity = new JLabel("City");  
lblcity.setBounds(30, 190, 100, 20);  
add(lblcity);
```

```
tfcity = new JTextField();  
tfcity.setBounds(230, 190, 200, 20);  
add(tfcity);
```

```
JLabel lblstate = new JLabel("State");  
lblstate.setBounds(30, 230, 100, 20);  
add(lblstate);
```

```
tfstate = new JTextField();  
tfstate.setBounds(230, 230, 200, 20);  
add(tfstate);
```

```
JLabel lblemail = new JLabel("Email");  
lblemail.setBounds(30, 270, 100, 20);  
add(lblemail);
```

```
tfemail = new JTextField();
```



```
tfemail.setBounds(230, 270, 200, 20);  
add(tfemail);
```

```
JLabel lblphone = new JLabel("Phone");  
lblphone.setBounds(30, 310, 100, 20);  
add(lblphone);
```

```
tfphone = new JTextField();  
tfphone.setBounds(230, 310, 200, 20);  
add(tfphone);
```

```
try {  
    Conn c = new Conn();  
    ResultSet rs = c.s.executeQuery("select * from customer where meter_no = '"+meter+"'");  
    while(rs.next()) {  
        name.setText(rs.getString("name"));  
        tfaddress.setText(rs.getString("address"));  
        tfcity.setText(rs.getString("city"));  
        tfstate.setText(rs.getString("state"));  
        tfemail.setText(rs.getString("email"));  
        tfphone.setText(rs.getString("phone"));  
        meternumber.setText(rs.getString("meter_no"));  
    }  
} catch (Exception e) {  
    e.printStackTrace();  
}
```

```
update = new JButton("Update");  
update.setBackground(Color.BLACK);  
update.setForeground(Color.WHITE);  
update.setBounds(70, 360, 100, 25);
```

```

add(update);
update.addActionListener(this);

cancel = new JButton("Cancel");
cancel.setBackground(Color.BLACK);
cancel.setForeground(Color.WHITE);
cancel.setBounds(230, 360, 100, 25);
add(cancel);
cancel.addActionListener(this);

ImageIcon i1 = new ImageIcon(ClassLoader.getResource("icon/update.jpg"));
Image i2 = i1.getImage().getScaledInstance(400, 300, Image.SCALE_DEFAULT);
ImageIcon i3 = new ImageIcon(i2);
JLabel image = new JLabel(i3);
image.setBounds(550, 50, 400, 300);
add(image);

setVisible(true);
}

public void actionPerformed(ActionEvent ae) {
    if (ae.getSource() == update) {
        String address = tfaddress.getText();
        String city = tfcity.getText();
        String state = tfstate.getText();
        String email = tfemail.getText();
        String phone = tfphone.getText();

        try {
            Conn c = new Conn();

```

```
        c.s.executeUpdate("update customer set address = '"+address+"', city = '"+city+"', state = '"+state+"', email = '"+email+"', phone = '"+phone+" where meter_no = '"+meter+"'");
```

```
        JOptionPane.showMessageDialog(null, "User Information Updated Successfully");
        setVisible(false);
    } catch (Exception e) {
        e.printStackTrace();
    }
} else {
    setVisible(false);
}
}
```

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

4.16 View Information

```
package electricity.billing.system;
```

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

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

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

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

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

```
    JButton cancel;
```

```
    ViewInformation(String meter) {
```

```
        setBounds(350, 150, 850, 650);
```

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

```
        setLayout(null);
```

```
JLabel heading = new JLabel("VIEW CUSTOMER INFORMATION");  
heading.setBounds(250, 0, 500, 40);  
heading.setFont(new Font("Tahoma", Font.PLAIN, 20));  
add(heading);
```

```
JLabel lblname = new JLabel("Name");  
lblname.setBounds(70, 80, 100, 20);  
add(lblname);
```

```
JLabel name = new JLabel("");  
name.setBounds(250, 80, 100, 20);  
add(name);
```

```
JLabel lblmeternumber = new JLabel("Meter Number");  
lblmeternumber.setBounds(70, 140, 100, 20);  
add(lblmeternumber);
```

```
JLabel meternumber = new JLabel("");  
meternumber.setBounds(250, 140, 100, 20);  
add(meternumber);
```

```
JLabel lbladdress = new JLabel("Address");  
lbladdress.setBounds(70, 200, 100, 20);  
add(lbladdress);
```

```
JLabel address = new JLabel("");  
address.setBounds(250, 200, 100, 20);  
add(address);
```

```
JLabel lblcity = new JLabel("City");  
lblcity.setBounds(70, 260, 100, 20);  
add(lblcity);
```

```
JLabel city = new JLabel("");  
city.setBounds(250, 260, 100, 20);  
add(city);
```

```
JLabel lblstate = new JLabel("State");  
lblstate.setBounds(500, 80, 100, 20);  
add(lblstate);
```

```
JLabel state = new JLabel("");  
state.setBounds(650, 80, 100, 20);  
add(state);
```

```
JLabel lblemail = new JLabel("Email");  
lblemail.setBounds(500, 140, 100, 20);  
add(lblemail);
```

```
JLabel email = new JLabel("");  
email.setBounds(650, 140, 100, 20);  
add(email);
```

```
JLabel lblphone = new JLabel("Phone");  
lblphone.setBounds(500, 200, 100, 20);  
add(lblphone);
```

```
JLabel phone = new JLabel("");  
phone.setBounds(650, 200, 100, 20);  
add(phone);
```

```

try {
    Conn c = new Conn();
    ResultSet rs = c.s.executeQuery("select * from customer where meter_no = '"+meter+"'");
    while(rs.next()) {
        name.setText(rs.getString("name"));
        address.setText(rs.getString("address"));
        city.setText(rs.getString("city"));
        state.setText(rs.getString("state"));
        email.setText(rs.getString("email"));
        phone.setText(rs.getString("phone"));
        meternumber.setText(rs.getString("meter_no"));
    }
} catch (Exception e) {
    e.printStackTrace();
}

cancel = new JButton("Cancel");
cancel.setBackground(Color.BLACK);
cancel.setForeground(Color.WHITE);
cancel.setBounds(350, 340, 100, 25);
add(cancel);
cancel.addActionListener(this);

ImageIcon i1 = new ImageIcon(ClassLoader.getResource("icon/viewcustomer.jpg"));
Image i2 = i1.getImage().getScaledInstance(600, 300, Image.SCALE_DEFAULT);
ImageIcon i3 = new ImageIcon(i2);
JLabel image = new JLabel(i3);
image.setBounds(20, 350, 600, 300);
add(image);

```

```
        setVisible(true);
    }

    public void actionPerformed(ActionEvent ae) {
        setVisible(false);
    }

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

Normalization of columns

First Normal Form (1NF)

In this stage, we ensure that the table has atomic values. Each column contains unique and indivisible data, and there are no repeating groups or arrays. Additionally, every record in the table has a unique identifier (Primary Key).

1NF - Bill Table

meter_no	month	units	totalbill	status
699906	January	450	4200	Paid
36510	January	600	5550	Paid

Second Normal Form (2NF)

To achieve 2NF, the table must first be in 1NF. Then, we remove partial dependencies by ensuring that all non-key attributes depend on the entire primary key. If the primary key consists of multiple columns, each non-key attribute must depend on the entire primary key. In our case, the table already satisfies 2NF without any partial dependencies.

2NF - Separate Customer Table

customer_id	customer_name	customer_address
1	John Doe	123 Elm St.
2	Jane Smith	456 Oak Rd.

Third Normal Form (3NF)

For 3NF, the table must first be in 2NF. We then eliminate transitive dependencies, meaning non-key attributes should not depend on other non-key attributes. In this case, there are no transitive dependencies, so the table is already in 3NF.

3NF - Remove Transitive Dependency (if needed)

customer_id	meter_no	month	units	totalbill	status
1	699906	January	450	4200	Paid
2	36510	January	600	5550	Paid

Fourth Normal Form (4NF)

To achieve 4NF, the table must first be in 3NF. We eliminate multivalued dependencies, ensuring that an attribute does not depend on more than one independent attribute. In this case, we separate multi-valued attributes, such as customer phone numbers and emails, into separate tables.

4NF - Separate Multi-valued Attributes (if needed)

Customer Phone Table:

customer_id	phone_number
1	123-456-7890
2	987-654-3210

Customer Email Table:

customer_id	email
1	johndoe@example.com
2	janesmith@example.com

Fifth Normal Form (5NF)

For 5NF, the table must be in 4NF. We eliminate any join dependencies, meaning we should ensure that data redundancy caused by joining multiple tables is removed. This is typically done by further splitting tables to ensure that each table contains data that can be retrieved without redundant joins.

5NF - No Join Dependencies (Ensure no data redundancy)

We ensure that many-to-many relationships are broken into separate tables to prevent data redundancy, allowing each table to store unique and non-redundant data.

VISUAL REPRESENTATION:

5.1 ADMIN INTERFACE

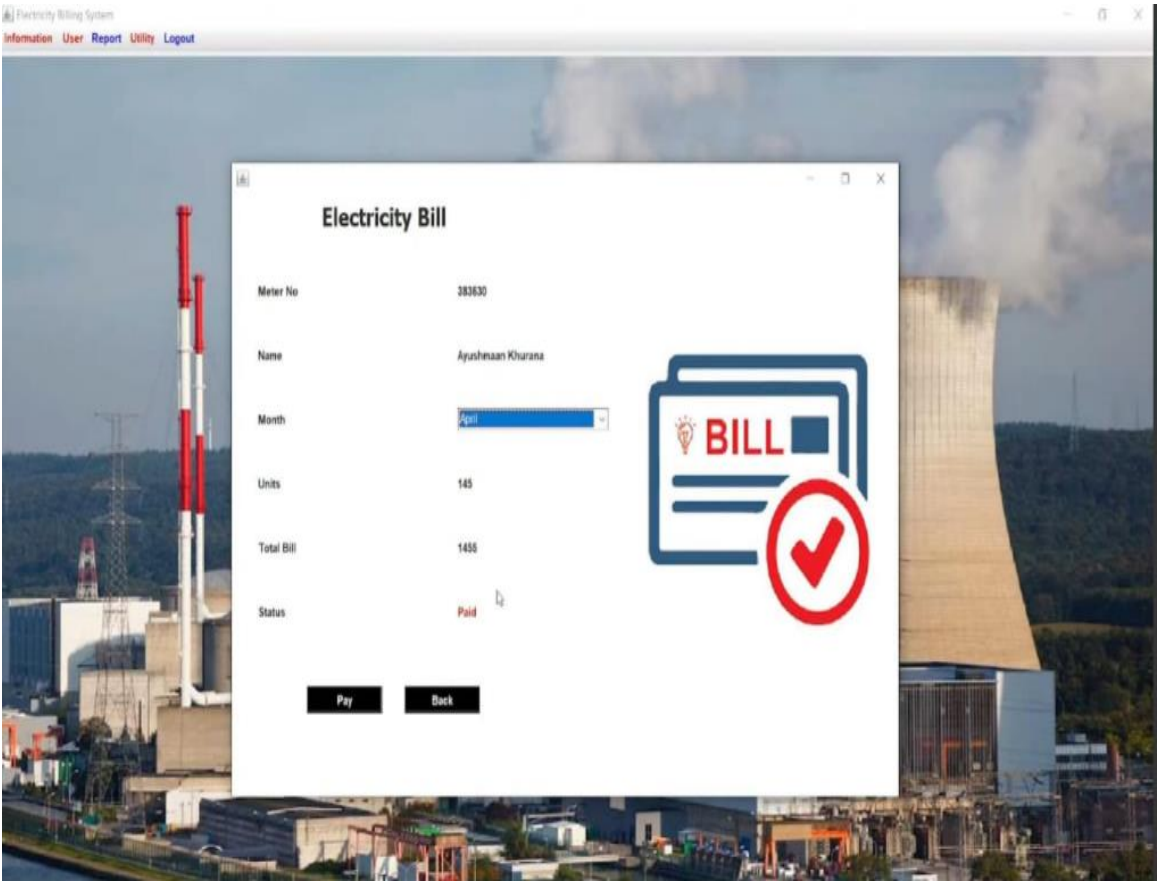
The screenshot shows the 'Meter Information' form in the 'Metering Systems' application. The form is displayed over a background image of a power plant. The form contains the following fields and values:

- Meter Number:** 383630
- Meter Location:** Outside
- Meter Type:** Electric Meter
- Phase Code:** 033
- Bill Type:** Normal
- Days:** 70 Days
- Note:** By Default Bill is calculated for 30 days only

At the bottom of the form, there are two buttons: **Submit** and **Cancel**. On the left side of the form, there is a black silhouette of a person holding a clipboard.

[illegible]

5.2 USER INTERFACE



Electricity Billing System
Information User Report Utility Logout

Generate Bill: 383630 April

Meter Number: 383630
Address: Delhi Road
State: UP
City: Meerut
Email: ayushmaan@gmail.com
Phone Number: 5569855632

Meter Location: Outside
Meter Type: Electric Meter
Phase Code: 033
Bill Type: Normal
Days: 30

Cost per Unit: Rs 9
Meter Rent: Rs 47
Service Charge: Rs 22
Service Tax: Rs 57
Swachh Bharat Cess: Rs 6
Fixed Tax: Rs 18

Current Month: April
Units Consumed: 145
Total Charges: 1455


TOTAL PAYABLE: 1455

Electricity Billing System
Information User Report Utility Logout

VIEW CUSTOMER INFORMATION

Name	Ayushmaan Kha...	State	Delhi
Meter Number	383630	Email	ayushmaan@gmail.com
Address	Delhi Road	Phone	5569855632
City	Delhi		

Back



CONCLUSIONS

The Electricity Bill Management System serves as a high-level solution in the management of billing and payment by using SQL, which securely and stably provides data storage with room for flexibility to ensure easy access. On the interface available, customers can view and download their bills as well as make payments. Administrators, on the other hand, manage accounts, track the process of reading meters and issuing bills, and so forth. Apart from that, it has real-time status of payments and features encryption and authentication as security measures. It boosts the level of operational efficiency and also user experience through the automation of bill generation and payment tracking. The system optimizes both electricity bill management for providers and customers by giving secure payment options and transparency.

REFERENCES

1. MySQL Documentation : <https://dev.mysql.com/doc/>
2. Java JDBC Tutorial : <https://www.javatpoint.com/java-jdbc>
3. Java Swing Official documentation :
<https://docs.oracle.com/javase/8/docs/api/javax/swing/package-summary.html>
4. Java Swing Tutorial : <https://www.javatpoint.com/java-swing>