A report on
“Electricity billing Management System”
Using OOPJ

Submitted to- Dr. Mayank Suhani

Submitted by-
Charul Deshpande (N242)
Gautam Kundalia(N243)
Aditi Tiwari(N234)

ACKNOWLEDGEMENT

We would like to express our sincere appreciation and gratitude to our esteemed faculty, Dr. Mayank Suhani, for his invaluable guidance, support, and encouragement throughout this project. Without his expertise and unwavering dedication, this project would not have been possible.

We would also like to thank each other for our tireless efforts and contributions towards the development of this object-oriented program in Java. It has been a privilege to work alongside Charul Deshpande, Gautam Kundalia, and Aditi Tiwari, whose exceptional skills, creativity, and dedication have been the driving force behind this project's success.

Together, we have accomplished something truly remarkable, and we look forward to applying the knowledge and experience gained from this project in our future endeavors.

INTRODUCTION

Electricity Billing System is a software-based application.

i. This project aims at serving the department of electricity by computerizing the billing system.
ii. It mainly focuses on the calculation of units consumed during the specified time and the money to be charged by the electricity offices.
iii. This computerized system will make the overall billing system easy, accessible, comfortable, and effective for consumers.

Problem Statement

The manual system is suffering from a series of drawbacks. Since whole of the bills is to be maintained with hands the process of keeping and maintaining the information is very tedious and lengthy to customer. It is very time consuming and laborious process because, staff need to be visited the customers place every month to give the bills and to receive the payments. For this reason, we have provided features Present system is partially automated(computerized), existing system is quite laborious as one must enter same information at different places.

ANALYSIS AND SYSTEM REQUIREMENT

Software & Hardware Requirements

Hardware Requirements:

- Hardware Specification: -Processor Intel Pentium V or higher
- Clock Speed: -1.7 GHz or more
- System Bus: -64 bits
- RAM: -16GB
- HDD: -2TB
- Monitor: -LCD Monitor
- Keyboard: -Standard keyboard
- Mouse: -Compatible mouse

Software Requirements:

- Operating System: -Windows 10
- Software: -Microsoft SQL Server
- Front End: -Java core/swings (NetBeans)
- Back End: -My SQL

IMPLIMENTATION

Implementation of operations

- **Adding Customer:** Here admin can add new customer to the customer list who started using electricity bill system.
- **Searching Deposit Details:** Here admin can search according to meter number and month to view deposit details.
- **Viewing Details**: Here admin and user can view customer details and about details.
- **Adding Tax:** Here admin can add tax details.
- **Updating Customer:** Here customer can update his/her details by using meter_no of the customer.
- **Delete Customer:** Here admin can delete details based on meter number.

Implementation of SQL statement

```
 show databases;

create database ebs;
use ebs;
create table login(meter_no varchar(20), username varchar(30), name varchar(30), password
varchar(20), user varchar(20));
create table customer(name varchar(20), meter_no varchar(20), address varchar(50), city
varchar(30),  state varchar(30), email varchar(40), phone varchar(20));
create table meter_info(meter_no varchar(20), meter_location varchar(20), meter_type
varchar(20), phase_code varchar(20), bill_type varchar(20), days varchar(20));
create table tax(cost_per_unit varchar(20), meter_rent varchar(20), service_charge
varchar(20), service_tax varchar(20), swacch_bharat_cess varchar(20), fixed_tax
varchar(20));
insert into tax values('9', '47', '22', '57', '6', '18');


create table bill(meter_no varchar(20), month varchar(30), units varchar(20), totalbill
varchar(20), status varchar(20));
select * from bill;
select * from customer;
select * from login;
```

Algorithm or pseudocode of implementation

Explanation of Algorithm or pseudocode of system:

- ✓ Start system
- ✓ Enter login name and password
- ✓ On clicking the login button
- ✓ Connect to database
- ✓ Query database to know whether user credentials are correct
- ✓ If not, deny access and return login page with an error message
- ✓ If correct, check if credentials for administrator
- ✓ If yes, allow login
- ✓ Set admin session, re-direct administrator to admin login page
- ✓ If no, allow login set user session
- ✓ Re-direct user to user home page

Algorithm or pseudocode of admin:

Login:

- This program will allow the admin to enter the username and password.
- If the entered credentials are correct, then the login will be successful otherwise need to be signup.
- If admin forgets password, it can be retrieved by giving username and answer for security question.
- After successful login the admin will be redirected to admin portal page where he/she can do following activities.

NewCustomer:

- This program will allow the admin to enter the customer details and automatically generates unique meter number.
- If customer name, address, city, state, email and phone number is entered, insert the values into customer
  else print error
  while next=true
  enter the meter_info details
  else print meter_info error
  Submit the details of customer that has been entered by clicking onto next button.
- If we need to cancel the particulars that has been entered click onto cancel option.

- If we need to submit the particulars that has been entered click onto submit option.

CustomerDetails:

- This program will allow the admin to view customer details.
- If we need to print the particulars that has been viewed click onto print option.

DepositDetails:

- This program will allow the admin to view bill details. If we need to sort the particulars based on meter_no and month.
- If we need to search the particulars that has been viewed click onto search option.
- If we need to print the particulars that has been viewed click onto print option.

TaxDetails:

- This program will allow the admin to add tax details.
  insert the values into tax
  else print error
  Submit the details of tax that has been entered by clicking onto submit button.
- If we need to cancel the particulars that has been entered click onto cancel option.

CalculateBill:

- This program will allow the admin to calculate total_bill when units consumed are inserted where meter_no and month is selected.
  insert the values into bill
  else print error
  Submit the details of tax that has been entered by clicking onto submit button.
- If we need to cancel the particulars that has been entered click onto cancel option.

DeleteBill:

- This Program will allow the admin to delete the customer info when meter_no is selected.
- If we need to delete the particulars that has been saved click onto delete option.
- If we need to cancel the particulars that has been entered click onto back option.

About:

- This program will allow the admin to view details of the project in short.
- If we need to exit the particulars that has been viewed click onto exit option.

Algorithm or pseudocode of Customer:

Login:

- This program will allow the customer to enter the username and password. If the entered credentials are correct, then the login will be successful otherwise need to be signup with the meter_no which is given by admin.
- If customer forgets password, it can be retrieved by giving username and answer for security question. After successful login the customer will be redirected to customer portal page where he/she can do following activities.

UpdateInfo1:

- This program will allow the customer to update the customer details. If customer address, city, state, email and phone number is updated,
update the values into customer
else print error
update the details of customer that has been updated by clicking onto update button.
- If we need to cancel the particulars that has been updated, click onto back option.

ViewInfo:

- This program will allow the customer to view his/her own details.
- If we need to go back from the particulars that has been viewed click onto back option.

PayBill:

- This program will allow the customer to view bill details and redirects to pay
- the bill where status will be updated.
- If we need to cancel the particulars that has been viewed click onto back option.
- If we need to pay the bill amount that has been viewed click onto pay option.

BillDetails:

- This program will allow the customer to view bill details.
- If we need to print the particulars that has been viewed click onto print option.

GenerateBill:

- This program will allow the customer to generate bill when meter_no and month is selected.
- Generate the details by clicking on generatebill button.

About:

- This program will allow the customer to view details of the project in short.
- If we need to exit the particulars that has been viewed click onto exit option.

NOTE: Utility (notepad, browser, calculator),query and logout is given to both customer and admin portals.

Sample Codes:

#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("");
    }
}
```

#Login Details:

```
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("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.getSystemResource("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.getSystemResource("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.getSystemResource("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.getSystemResource("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);
```

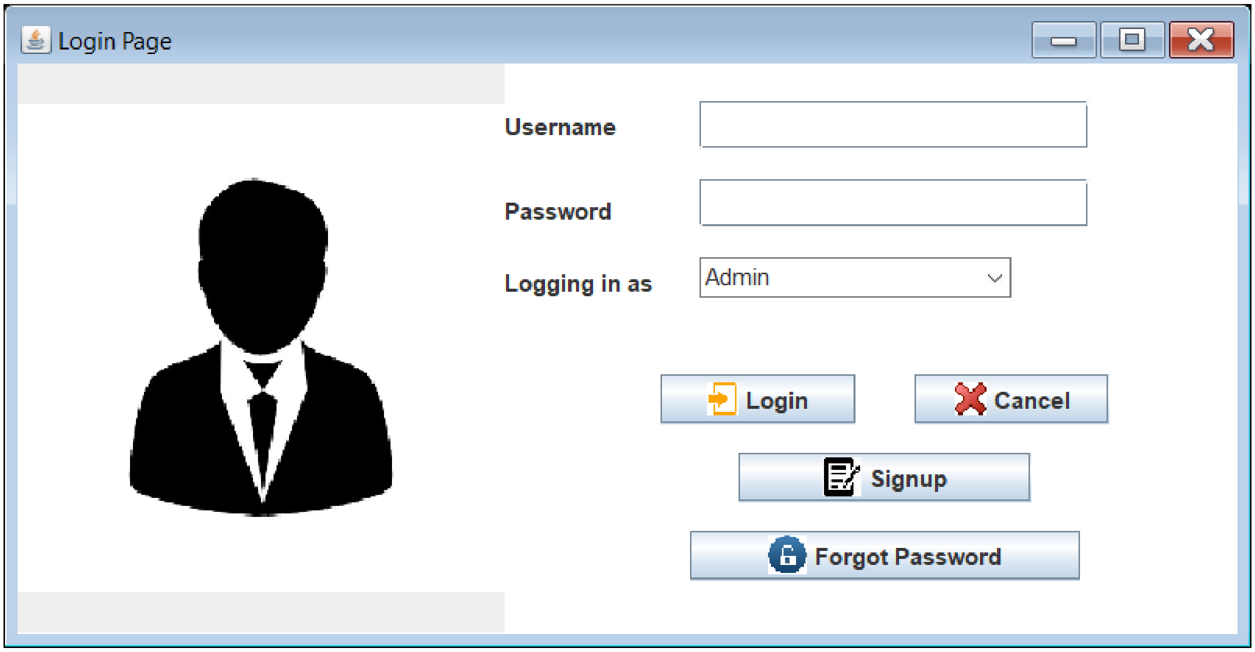
```
        add(image);

        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();
    }
}
```

#Generate 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();
        }
```

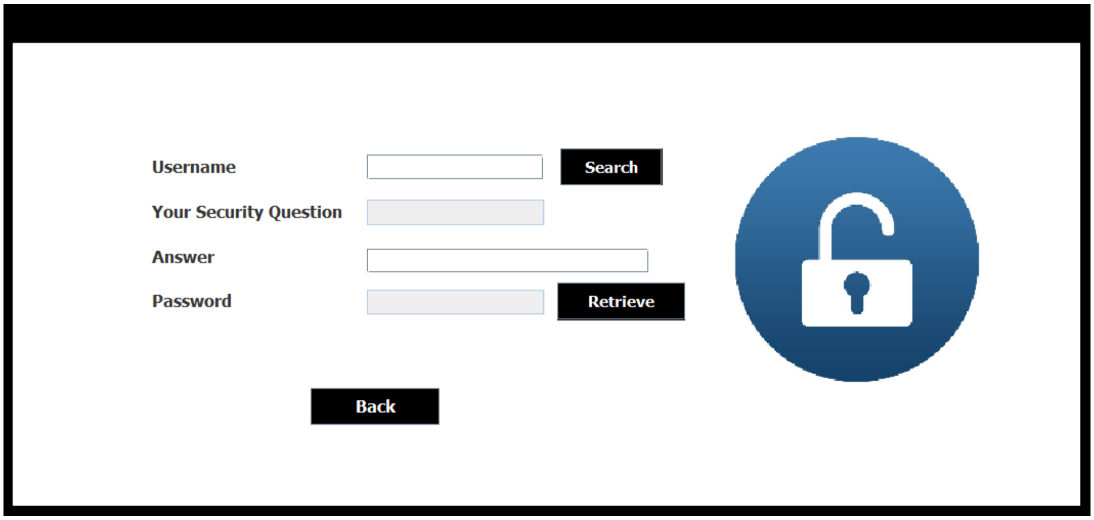
```
    }

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

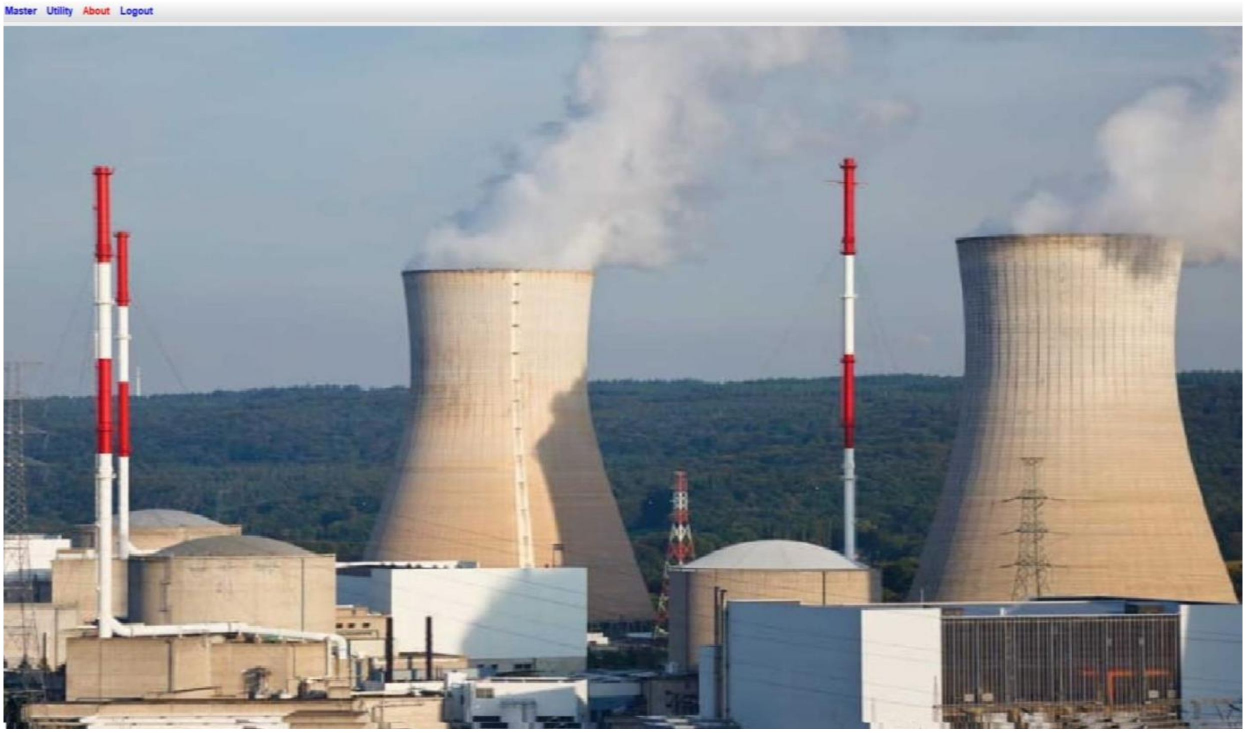
SNAPSHOTS:

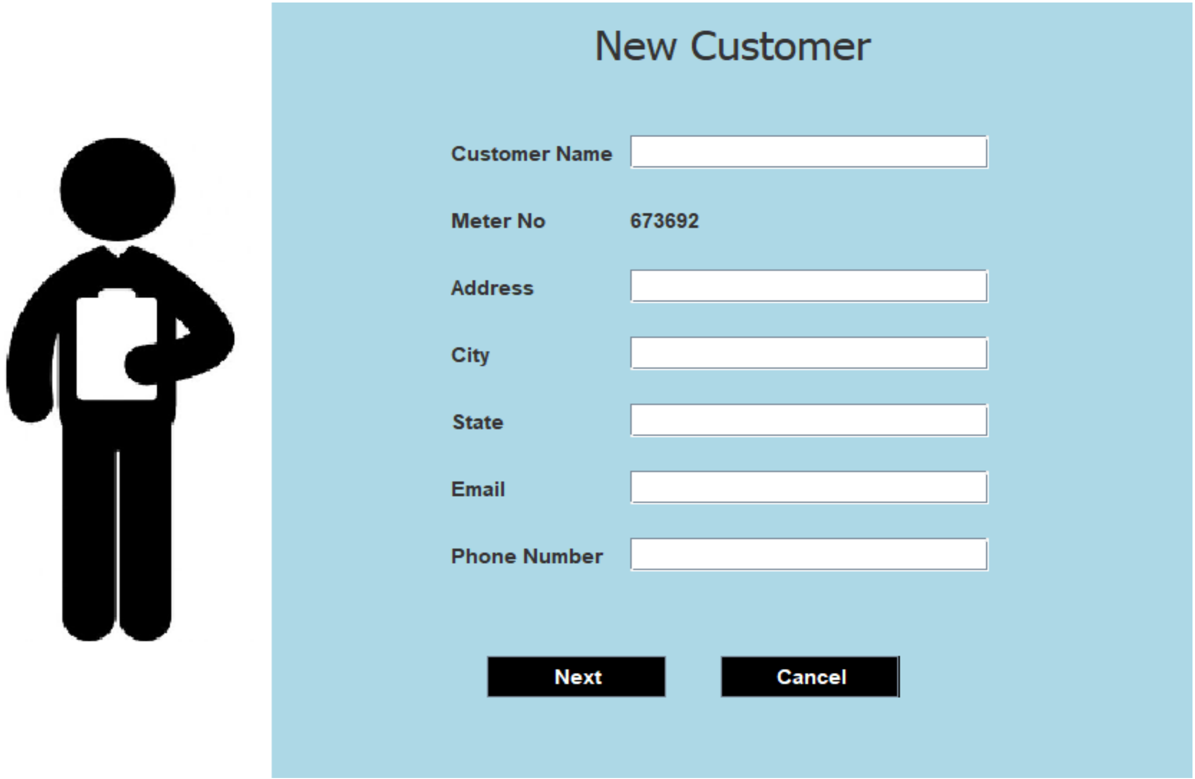


Login Page

Username	
Password	
Logging in as	Admin


Username
Search
Your Security Question
Answer
Password
Retrieve
Back


Master
Utility
About
Logout



New Customer

Customer Name

Meter No 673692

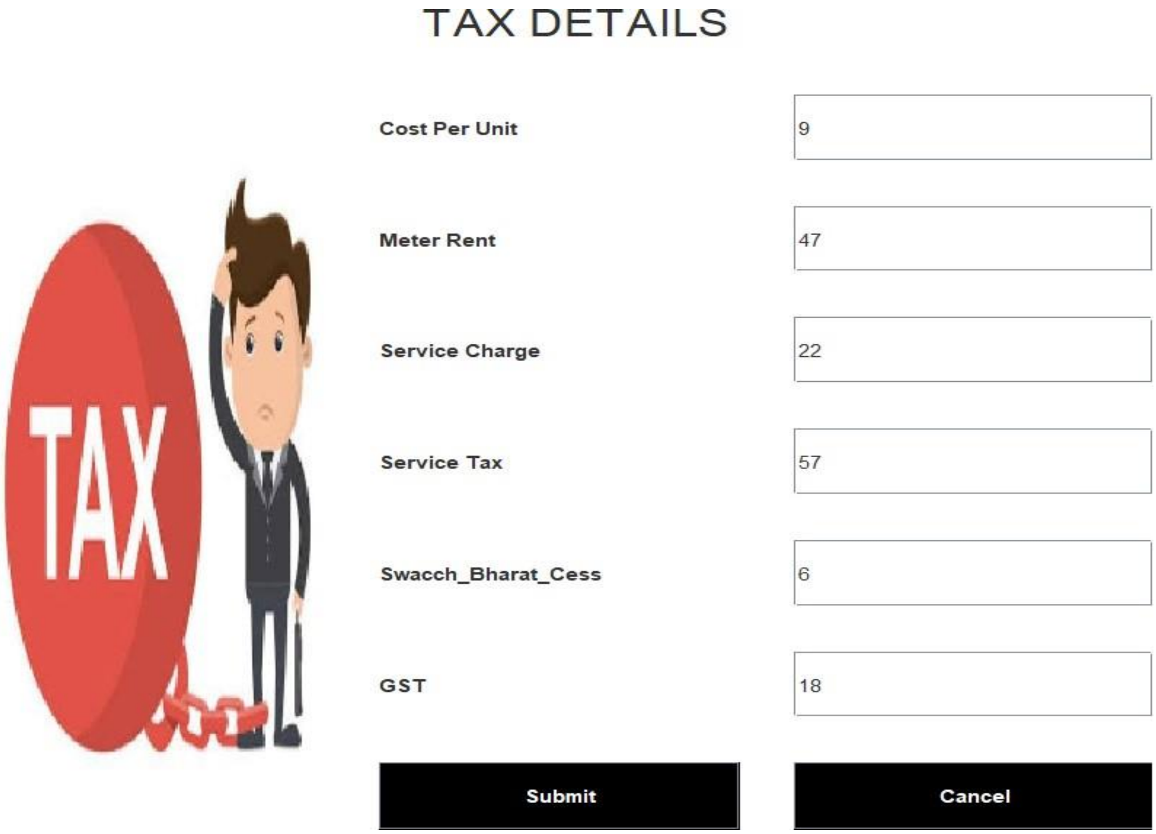
Address

City

State

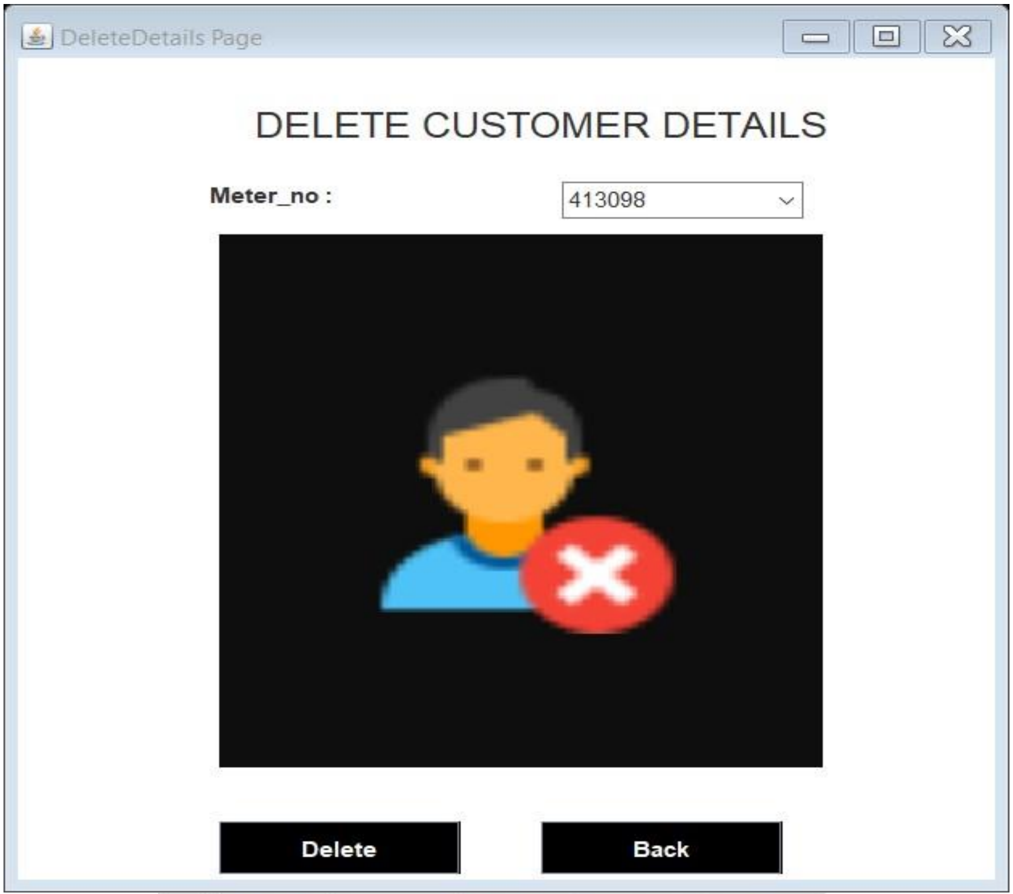
Email

Phone Number

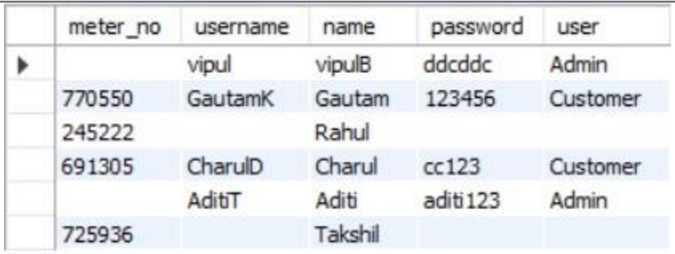


TAX DETAILS

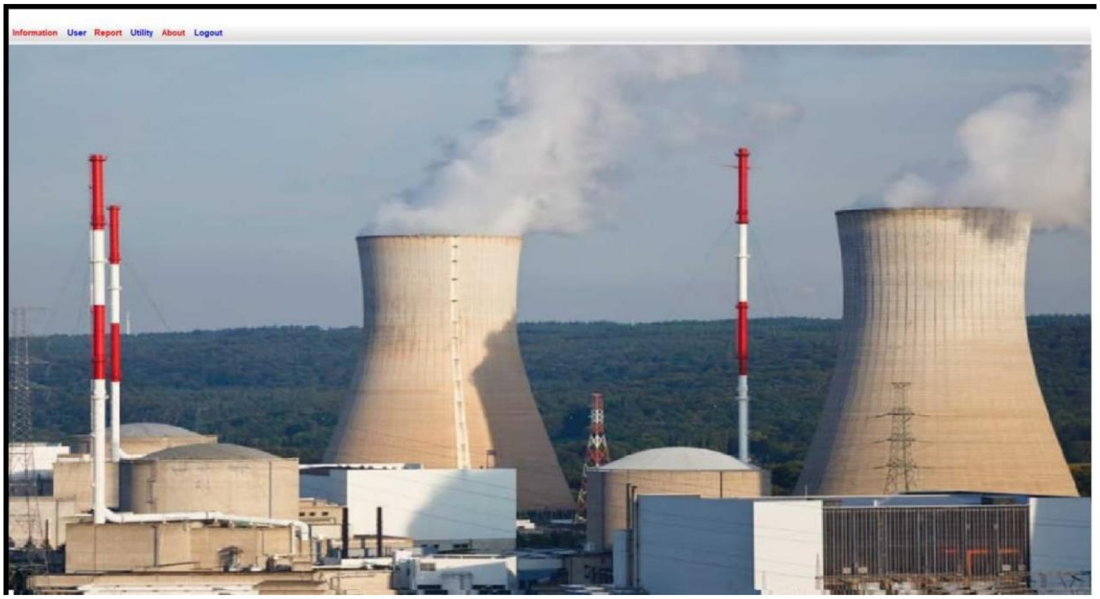
Cost Per Unit	9
Meter Rent	47
Service Charge	22
Service Tax	57
Swacch_Bharat_Cess	6
GST	18

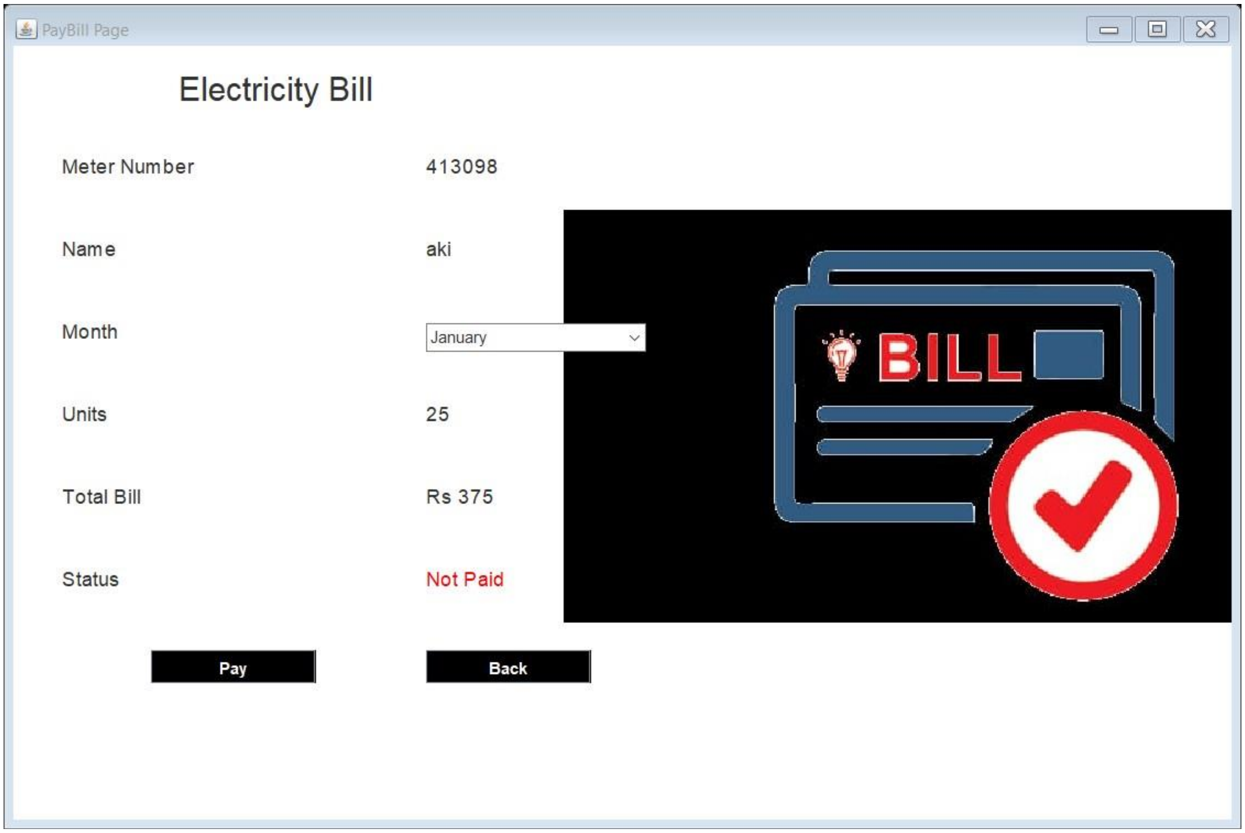


Meter_no : 413098



	meter_no	username	name	password	user
▶		vipul	vipulB	ddcddc	Admin
	770550	GautamK	Gautam	123456	Customer
	245222		Rahul		
	691305	CharulD	Charul	cc123	Customer
		AditiT	Aditi	aditi123	Admin
	725936		Takshil		

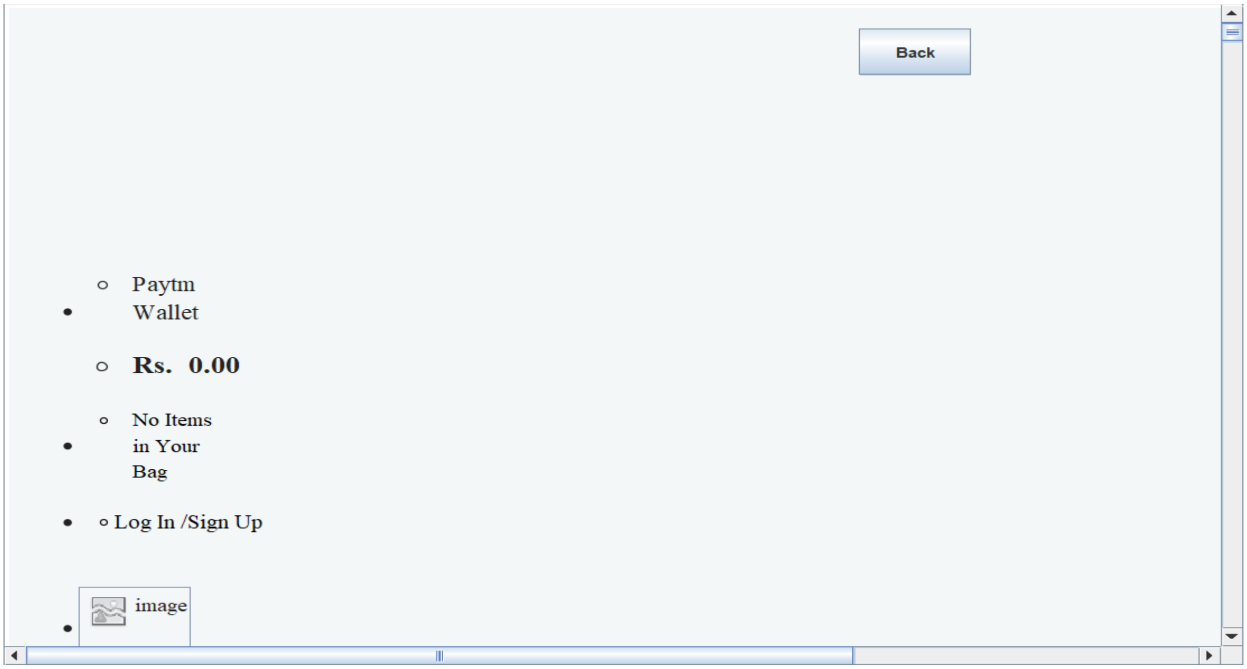

Information
User
Report
Utility
About
Logout



Electricity Bill

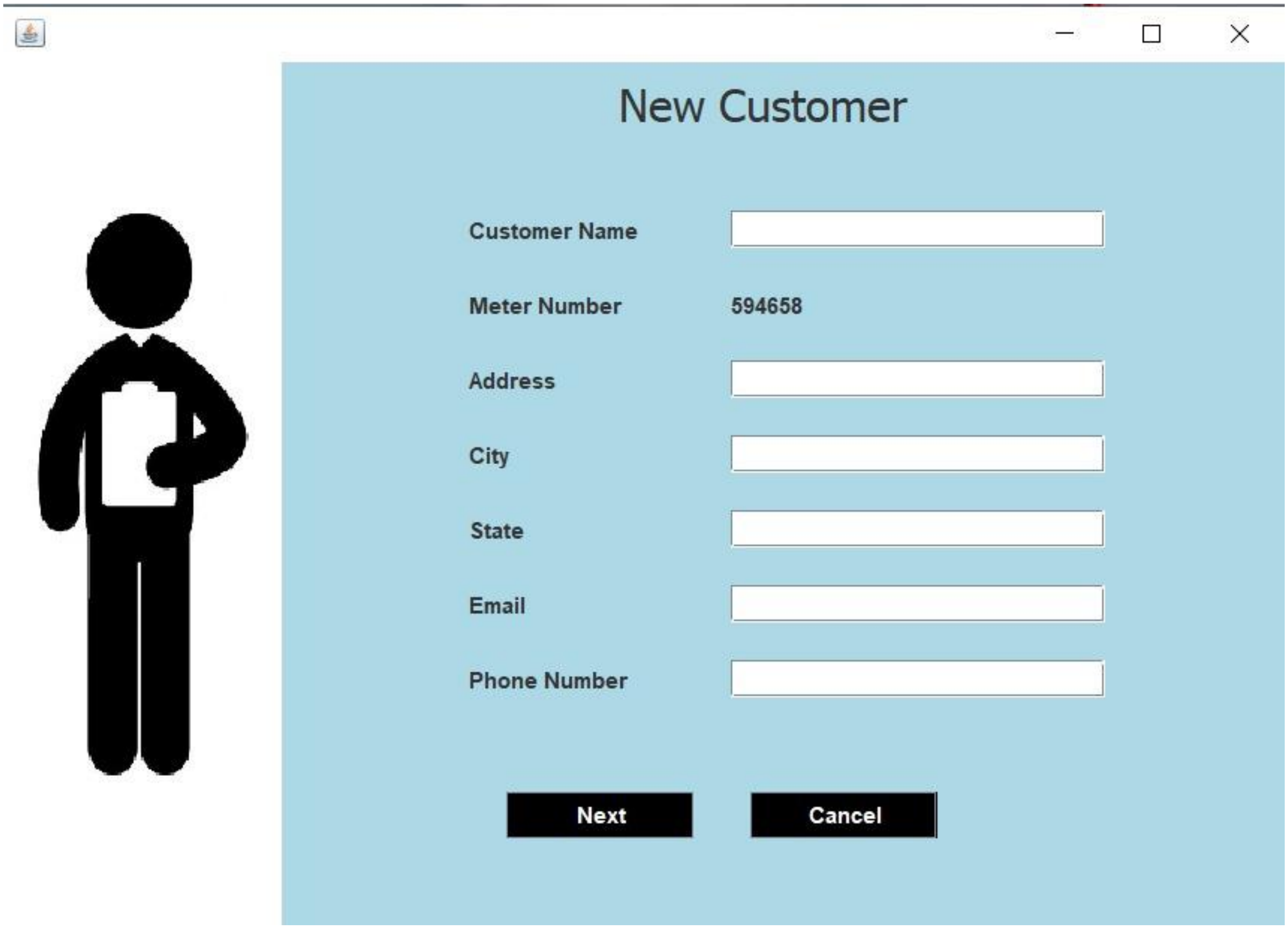
Meter Number	413098
Name	aki
Month	January
Units	25
Total Bill	Rs 375
Status	Not Paid

Pay Bill page



- ○ Paytm Wallet
- ○ **Rs. 0.00**
- ○ No Items in Your Bag
- ○ Log In /Sign Up
- image

Paytm page



New Customer

Customer Name

Meter Number 594658

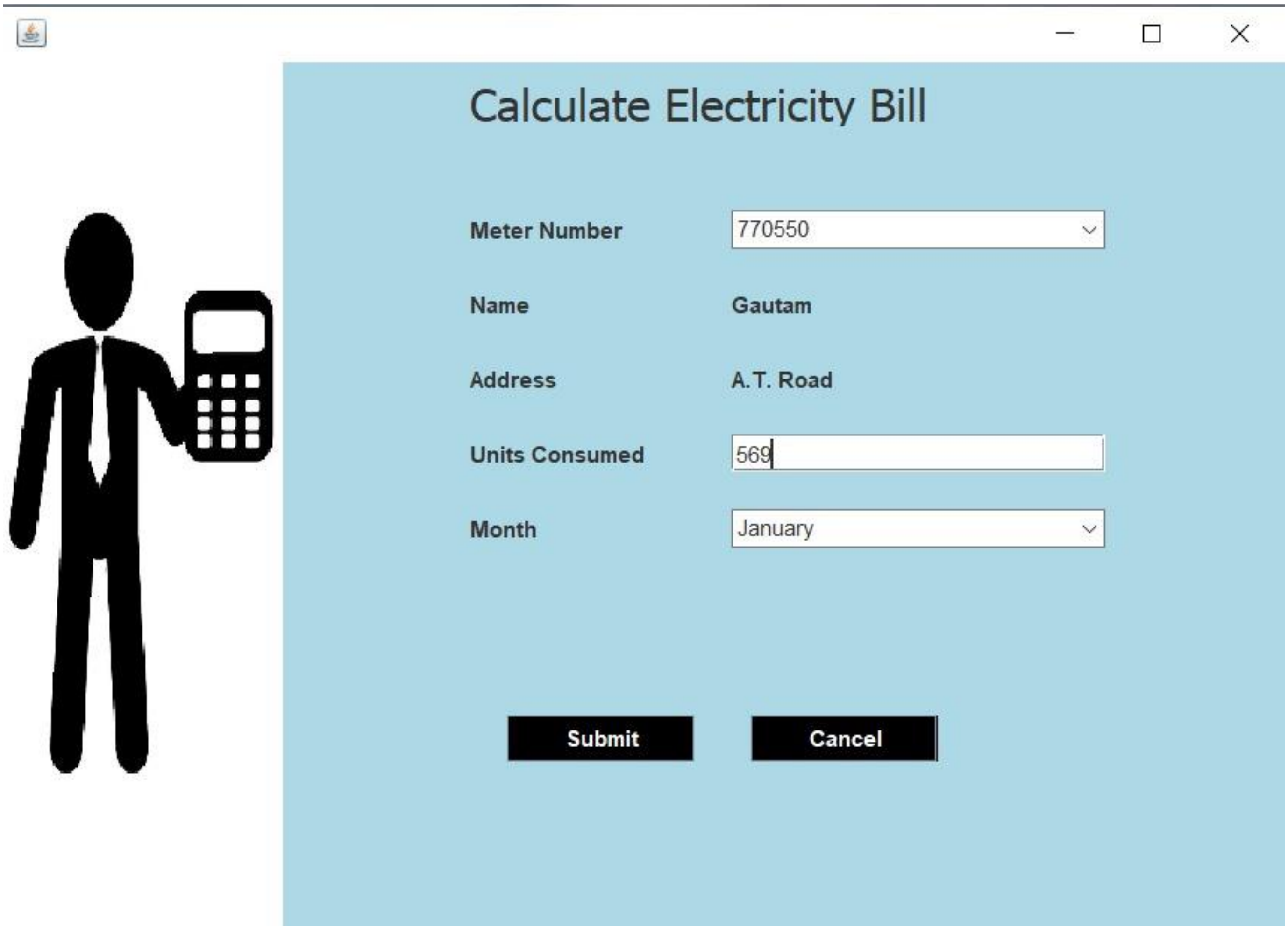
Address

City

State

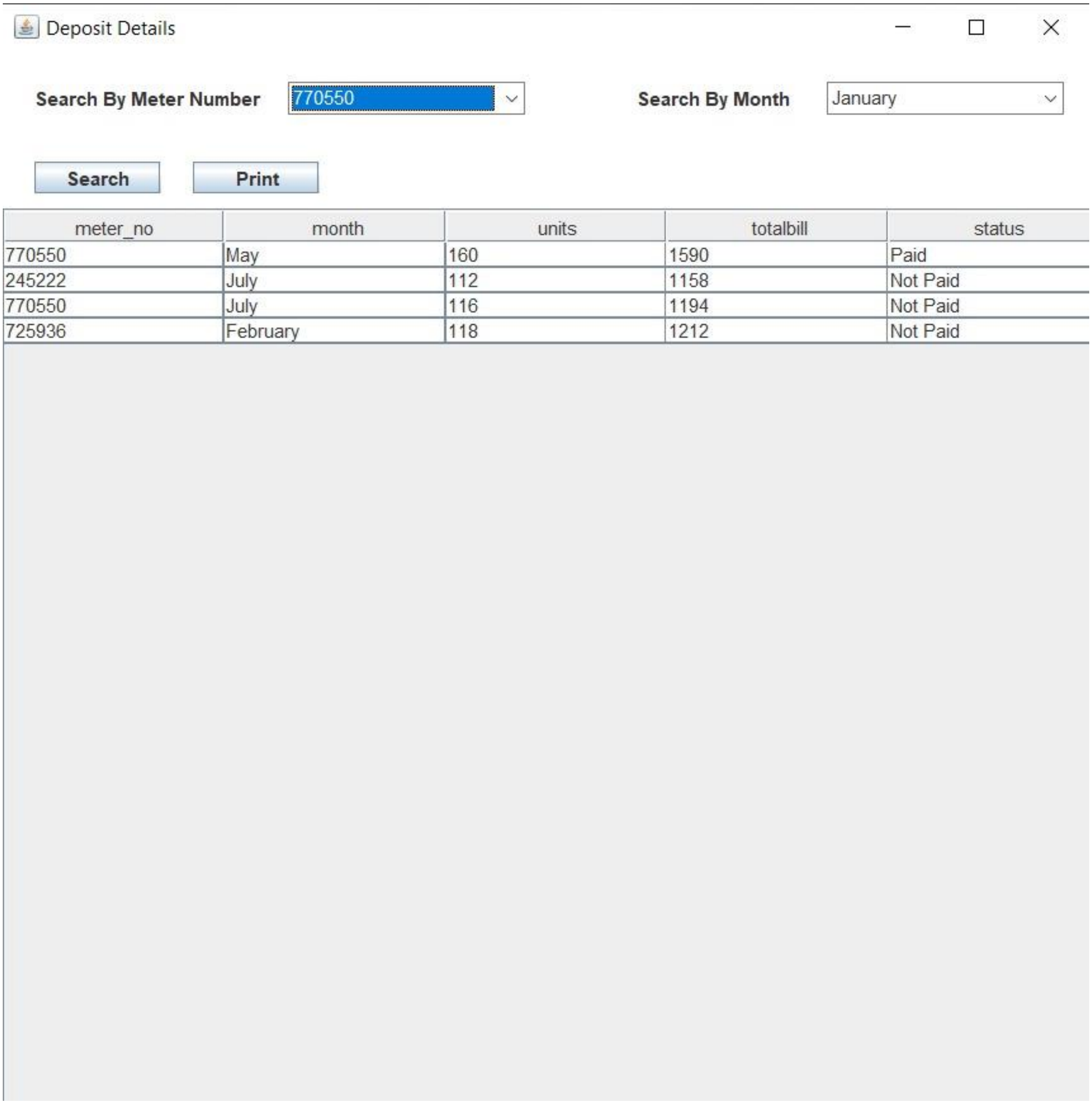
Email

Phone Number



Calculate Electricity Bill

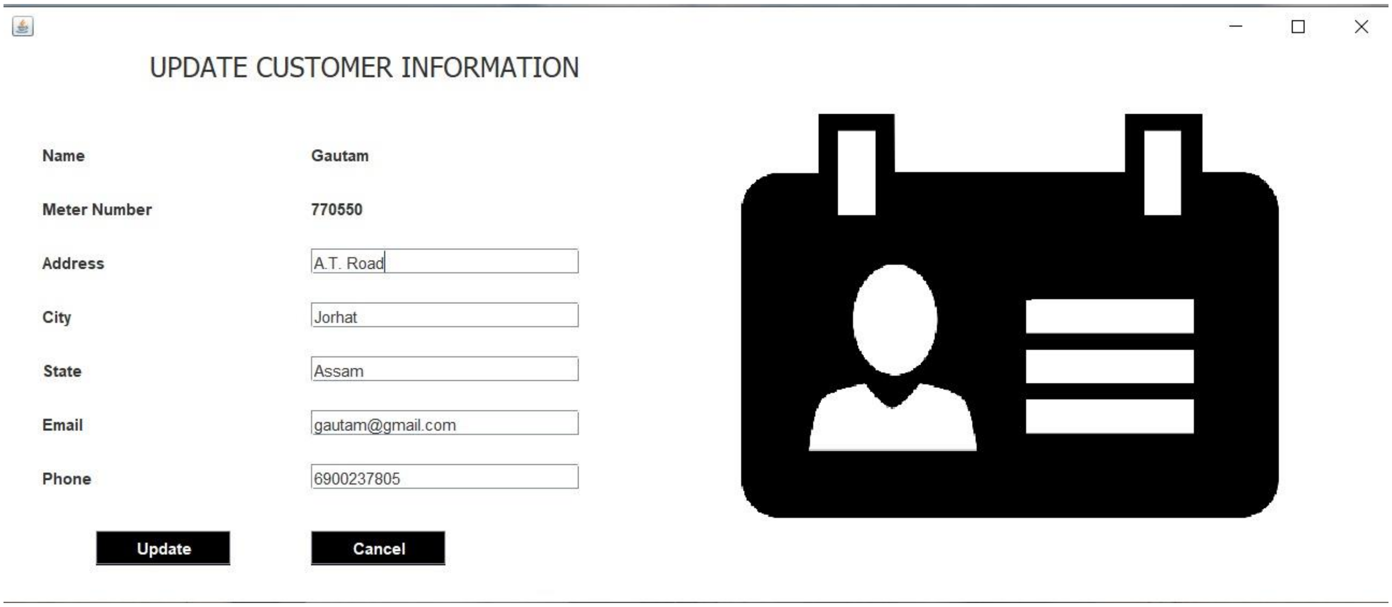
Meter Number	770550
Name	Gautam
Address	A.T. Road
Units Consumed	569
Month	January



Search By Meter Number: 770550

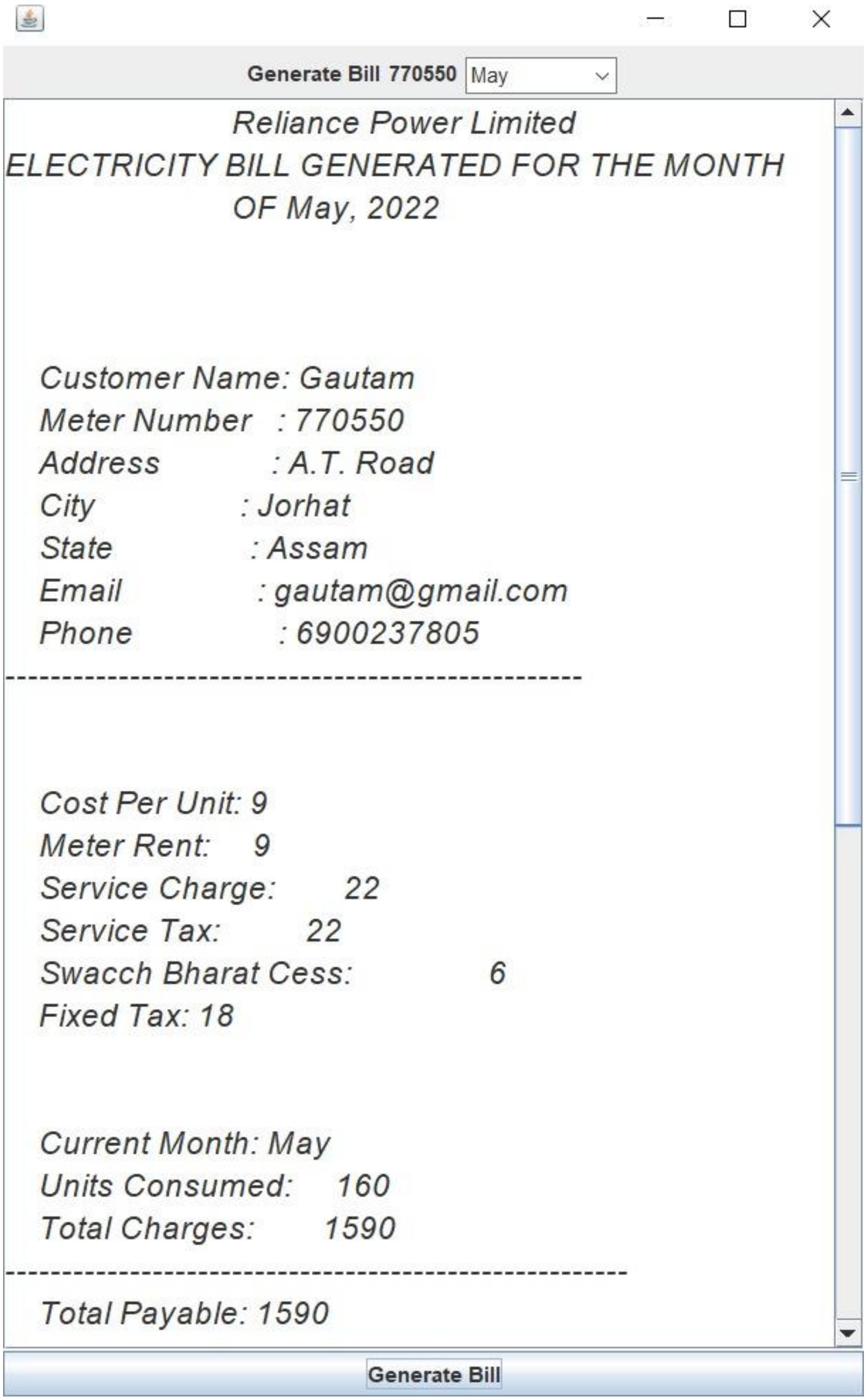
Search By Month: January

meter_no	month	units	totalbill	status
770550	May	160	1590	Paid
245222	July	112	1158	Not Paid
770550	July	116	1194	Not Paid
725936	February	118	1212	Not Paid



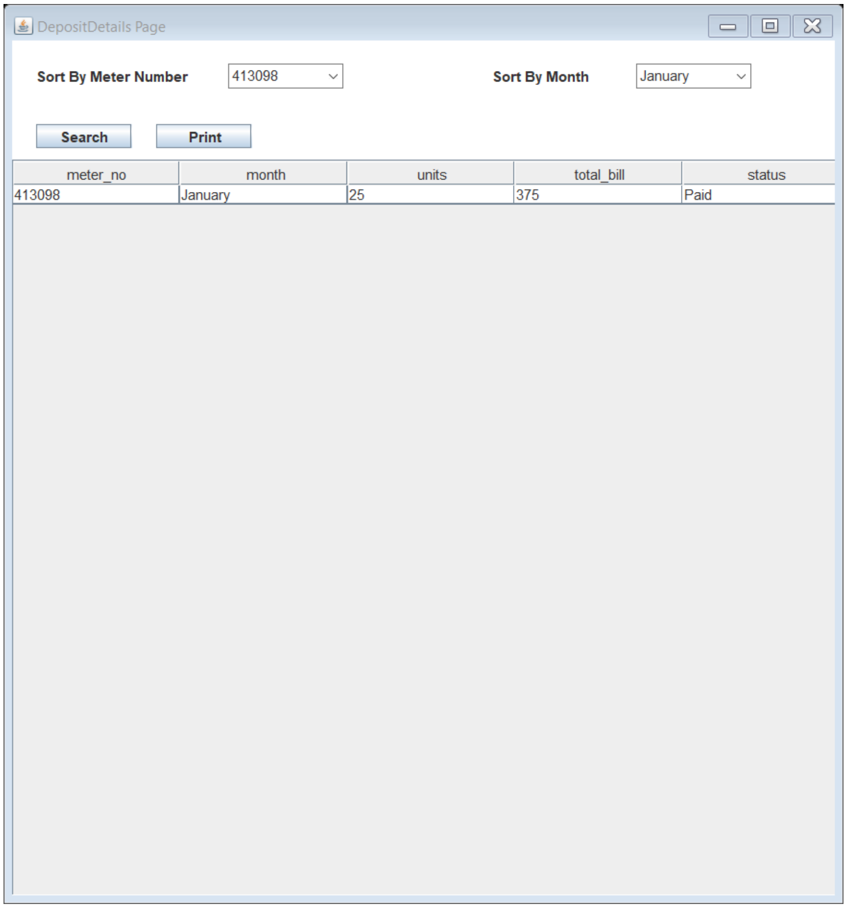
UPDATE CUSTOMER INFORMATION

Name	Gautam
Meter Number	770550
Address	A.T. Road
City	Jorhat
State	Assam
Email	gautam@gmail.com
Phone	6900237805

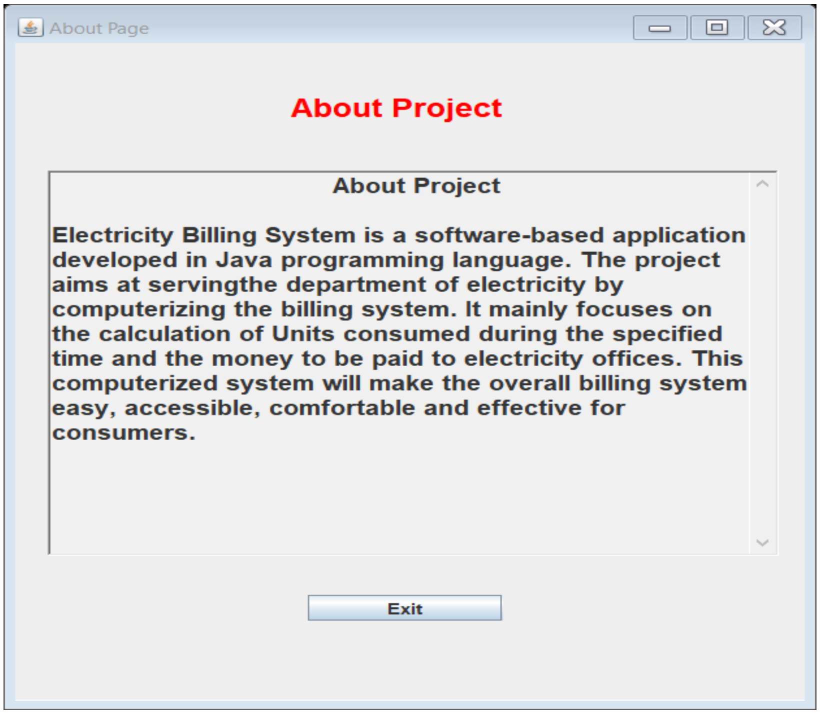


```
                      Reliance Power Limited
ELECTRICITY BILL GENERATED FOR THE MONTH
                      OF May, 2022



   Customer Name: Gautam
   Meter Number   : 770550
   Address           : A.T. Road
   City               : Jorhat
   State              : Assam
   Email              : gautam@gmail.com
   Phone               : 6900237805
----------------------------------------------------------



   Cost Per Unit: 9
   Meter Rent:     9
   Service Charge:       22
   Service Tax:         22
   Swacch Bharat Cess:              6
   Fixed Tax: 18


   Current Month: May
   Units Consumed:    160
   Total Charges:       1590
---------------------------------------------------------------
   Total Payable: 1590
```

FIG 6.26: Generate Bill page



meter_no	month	units	total_bill	status
413098	January	25	375	Paid



About Project

About Project

Electricity Billing System is a software-based application developed in Java programming language. The project aims at servingthe department of electricity by computerizing the billing system. It mainly focuses on the calculation of Units consumed during the specified time and the money to be paid to electricity offices. This computerized system will make the overall billing system easy, accessible, comfortable and effective for consumers.

CONCLUSION

The project aims to provide a convenient and user-friendly platform for customers to pay their electricity bills online, without the need for physically visiting the payment center.

We have utilized various Java programming concepts and technologies to develop this project, including object-oriented programming, database connectivity using JDBC, user interface development using Swing, and more.

Our project provides various features such as user registration, login, viewing and paying bills, updating user information, and viewing payment history. With the help of these features, customers can easily manage their electricity bills from the comfort of their homes, saving time and effort.

Through the development of this project, we have gained valuable insights and experience in Java programming and database connectivity. We believe that our project can serve as a useful reference for future developers looking to develop similar projects.

In conclusion, we are proud of the work we have done, and we hope that our project will prove to be useful and beneficial to its users.

BIBLIOGRAPHY

REFERENCES

Book Reference

Core Java An Integrated Approach.(TEXTBOOK).

Websites

- http://www.github.com
- www.stackoverflow.com
- www.google.com
- http://www.javatpoint.com/