Name:- Amruta S. Bagde
Email:- amrutabagde7@gmail.com

Task – 2: Online Survey system

Table of Contents



Acknowledgement

I would like to express my sincere gratitude to Veritech Software Services for providing me with the opportunity to work on this project and gain valuable hands-on experience in Java development.

Introduction:-

The introduction of an online survey system provides a comprehensive overview of the purpose, scope, and objectives of the system. It serves as a foundational component that sets the stage for understanding the significance and benefits of implementing such a system.

1. **Purpose**: The introduction elucidates the primary purpose of the online survey system, which typically revolves around gathering feedback, opinions, or data from a targeted audience. It may highlight the need for efficient data collection methods to inform decision-making processes, improve products or services, conduct research, or gauge customer satisfaction.
2. **Scope**: The introduction delineates the scope of the online survey system, elucidating the breadth and depth of its functionalities and features. It may outline the types of surveys supported (e.g., customer satisfaction surveys, market research surveys, employee feedback surveys) and the intended audience (e.g., customers, employees, stakeholders, general public).
3. **Objectives**: The introduction articulates the specific objectives or goals that the online survey system aims to achieve. This may include enhancing data collection efficiency,

improving response rates, ensuring data accuracy and integrity, facilitating data analysis and reporting, and ultimately deriving actionable insights to drive informed decision-making.

4. **Significance**: The introduction underscores the significance and relevance of implementing an online survey system in today's digital landscape. It may emphasize the transformative impact of leveraging technology to streamline survey administration, reach a broader audience, reduce costs, and enhance the overall user experience. Additionally, it may highlight the competitive advantage gained by organizations that embrace data-driven approaches to decision-making and stakeholder engagement.
5. **Benefits**: The introduction enumerates the potential benefits and advantages associated with using an online survey system. This may include increased accessibility and convenience for respondents, faster turnaround times for data collection and analysis, greater flexibility in survey design and deployment, real-time reporting and analytics, and scalability to accommodate varying survey sizes and complexities.

User Authentication:-

Login Credentials
Authentication Mechanism
Role-based Access Control
Secure Transmission
Session Management
Account Management
Audit Trails
Compliance Standards

Technologies

1.JAVA:-

Java is a widely-used programming language for coding web applications. It has been a popular choice among developers for over two decades, with millions of Java applications in use today. Java is a multi-platform, object-oriented, and network-centric language that can be used as a platform in itself. It is a fast, secure, reliable programming language for coding everything from mobile apps and enterprise software to big data applications and server-side technologies.

The Java program was the first language to combine both methods above using a Java Virtual Machine (JVM). The Java code compiler is called the Java Virtual Machine. Any Java file is first compiled into bytecode. Java bytecode can only run in the JVM. The JVM then interprets the bytecode to run it on the underlying hardware platform. So if the application is running on a Windows

machine, the JVM will interpret it for Windows. But if it is running on an open-source platform like Linux, the JVM will interpret  it for Linux.

2.Visual Studio:-

Visual Studio is a powerful integrated development environment (IDE) widely used by software developers to build various types of applications, including web, mobile, desktop, cloud, and game development projects. Developed by Microsoft, Visual Studio provides a comprehensive suite of tools, services, and features that streamline the development process and enhance productivity. It offers robust code editing capabilities with features like syntax highlighting, IntelliSense code completion, and refactoring tools. Visual Studio also includes built-in debugging, profiling, and testing tools to help developers identify and fix issues quickly. Additionally, it supports collaboration through version control systems like Git and integration with Azure DevOps for continuous integration and deployment. With its extensive support for multiple programming languages and platforms, Visual Studio remains a preferred choice for developers seeking a feature-rich and efficient development environment.

Libraries/Framework:-

SWING LIBRARY:- The Swing library is a part of Java's standard GUI toolkit, providing developers with a rich set of components to create interactive and visually appealing graphical user interfaces (GUIs) for their applications. Swing offers a wide range of components such as JFrame for creating windows, JPanel for organizing and grouping components, JLabel for displaying text or images, JTextField for accepting user input, JPasswordField for password input fields, and JButton for clickable buttons. These components can be customized with various properties and behaviors to suit the application's requirements.

Goals in creation of JAVA, HTML/CSS Javascript:-

The creation of the Java language aimed to achieve five key objectives:

- Ease of Use and Object-Oriented Design:
- Reliability and Safety:
- Cross-Platform Compatibility and Portability:
- Optimized Performance:
- Dynamic and Interpreted Execution with Threading Support:

Software Specifications:-

Operating System:-Linux
Coding Language:- JAVA
IDE and Code Editor:-visual studio

Hardware Specification:-

System:-Intel core i5
RAM:-7.7GB

Functionality:-
This code is to create a graphical user interface (GUI) for guests to participate in an online survey.

```
package survey;

import java.awt.Font;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.ResultSet;
import java.sql.SQLException;

import javax.swing.ButtonGroup;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JRadioButton;

public class Guest {

	SQLManage manage;
	int[] opt;
	int k;
```

```
public void guestView(String surveyCode) throws SQLException {

        manage = new SQLManage();
        ResultSet rst = manage.getQuestions(surveyCode);
        opt = new int[50];

        Font options = new Font("Times New Roman", Font.BOLD, 15);

        JFrame frame = new JFrame();
        frame.setSize(800, 600);
        frame.setLayout(null);
        frame.setLocationRelativeTo(null);

        JLabel start = new JLabel("ATTENDING THE SURVEY");
        start.setBounds(0, 50, 800, 50);
        start.setHorizontalAlignment(JLabel.CENTER);
        start.setFont(new Font("Times New Roman", Font.BOLD,
        40)); frame.add(start);

        JLabel ques = new JLabel("Question Here!!!");
        ques.setBounds(80, 200, 500, 30);
        ques.setFont(new Font("Times New Roman", Font.BOLD,
        18)); frame.add(ques);

        JRadioButton op1 = new JRadioButton("Option1");
        JRadioButton op2 = new JRadioButton("Option2");
        JRadioButton op3 = new JRadioButton("Opyion3");
        JRadioButton op4 = new JRadioButton("Option4");

        ButtonGroup bgroup = new ButtonGroup();
        bgroup.add(op1);
        bgroup.add(op2);
        bgroup.add(op3);
        bgroup.add(op4);

        op1.setBounds(100, 250, 500, 30);
        op2.setBounds(100, 300, 500, 30);
        op3.setBounds(100, 350, 500, 30);
        op4.setBounds(100, 400, 500, 30);

        op1.setFont(options);
        op2.setFont(options);
        op3.setFont(options);
        op4.setFont(options);
```

```
if(rst.next()) {
	ques.setText(rst.getString("question"));
	op1.setText(rst.getString("option1"));
	op2.setText(rst.getString("option2"));
	op3.setText(rst.getString("option3"));
	op4.setText(rst.getString("option4"));
}

frame.add(op1);
frame.add(op2);
frame.add(op3);
frame.add(op4);
k=0;

JButton nextButton = new JButton("NEXT");
nextButton.setBounds(100, 470, 600, 50);
frame.add(nextButton);
nextButton.addActionListener(new ActionListener() {
	@Override
	public void actionPerformed(ActionEvent e) {
		int x;
		if(op1.isSelected()) {
			x=1;
		}
		else if(op2.isSelected()) {
			x=2;
		}
		else if(op3.isSelected()) {
			x=3;
		}
		else if(op4.isSelected()) {
			x=4;
		}
		else
			x=0;

		if(x!=0) {
			opt[k] = x;
			k++;
			try {
				if(rst.next()) {
					ques.setText(rst.getString("question"));
					op1.setText(rst.getString("option1"));
```

```
                                        op2.setText(rst.getString("option2"));
                                        op3.setText(rst.getString("option3"));
                                        op4.setText(rst.getString("option4"));
                                    }
                                    else {
                                        for(int j=0; j<k; j++) {
                                            manage.answerUpdt(surveyCode,
j+1, opt[j]);
                                        }
                                        JOptionPane.showMessageDialog(frame,
"Survey Completed. Thank You.", "Congratulations", JOptionPane.PLAIN_MESSAGE);
                                        manage.addTotal();
                                        frame.dispose();
                                    }
                                } catch (SQLException e1) {
                                    e1.printStackTrace();
                                }
                            }
                            else {
                                JOptionPane.showMessageDialog(frame, "Select an
option!", "Warning", JOptionPane.WARNING_MESSAGE);
                            }
                            bgroup.clearSelection();
                        }
                    });


                    frame.setVisible(true);

                }
            }
```

The main purpose of this Java code is to create a graphical user interface (GUI) for user authentication and survey access in an online survey system. It allows users to log in securely, sign up for new accounts, and participate in surveys either as registered users or guests. The code interacts with a SQL database to manage user authentication and survey data.

```
package survey;

import java.awt.Font;
import java.awt.event.ActionEvent;
```

```
import java.awt.event.ActionListener;
import java.sql.SQLException;
import java.util.logging.Level;
import java.util.logging.Logger;

import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPasswordField;
import javax.swing.JTextField;

public class Login {

	int id;

	public void loginView() throws SQLException {
		SQLManage manage = new SQLManage();
		JFrame frame = new JFrame();
		frame.setSize(450, 450);
		frame.setLayout(null);
		frame.setLocationRelativeTo(null);

		JLabel heading = new JLabel("SURVEY SYSTEM");
		heading.setBounds(0, 50, 450, 50);
		heading.setHorizontalAlignment(JLabel.CENTER);
		heading.setFont(new Font("Times New Roman", Font.BOLD,
		40)); frame.add(heading);

		JLabel uname = new JLabel("Username : ");
		uname.setBounds(50, 130, 150, 50);
		frame.add(uname);

		JTextField name = new JTextField();
		name.setBounds(50, 170, 350, 30);
		frame.add(name);

		JLabel upass = new JLabel("Password : ");
		upass.setBounds(50, 200, 150, 50);
		frame.add(upass);

		JPasswordField pass = new JPasswordField();
		pass.setBounds(50, 240, 350, 30);
		frame.add(pass);
```

```
JButton login = new JButton("LOGIN");
login.setBounds(100, 300, 100, 40);
frame.add(login);
login.addActionListener(new ActionListener() {
	@Override
	public void actionPerformed(ActionEvent e) { String
		username = name.getText(); String password
		= pass.getText(); if(username.isEmpty() ||
		password.isEmpty()) {
			JOptionPane.showMessageDialog(frame, "Please Enter
All Details!!!", "Warning Message", JOptionPane.WARNING_MESSAGE);
		}
		else {
	SQLManage manage = null;
	try {
	   manage = new SQLManage();
	} catch (SQLException ex) {
	   Logger.getLogger(Login.class.getName()).log(Level.SEVERE, null,
ex);
	}
	try {
	   id = manage.authUser(username, password);
	} catch (SQLException ex) {
	   Logger.getLogger(Login.class.getName()).log(Level.SEVERE, null,
ex);
	}
			if (id == -1) {
				JOptionPane.showMessageDialog(frame, "No User
Found!!!", "Warning Message", JOptionPane.WARNING_MESSAGE);
			}
			else if(id == 0) {
				JOptionPane.showMessageDialog(frame, "Wrong
Password!!!", "Warning Message", JOptionPane.WARNING_MESSAGE);
			}
			else {
				MainPage mainPage = new MainPage();
				try {
					mainPage.mainPageView(id);
				} catch (SQLException e1) {
					e1.printStackTrace();
				}
				frame.dispose();
			}
```

```
                        }
                    }
                });

                JButton signUp = new JButton("SIGNUP");
                signUp.setBounds(250, 300, 100, 40);
                frame.add(signUp);
                signUp.addActionListener(new ActionListener() {
                    @Override
                    public void actionPerformed(ActionEvent e) {
                        SignUp signup = new SignUp();
                        signup.signUpView();
                    }
                });

                JButton attend = new JButton("ATTEND A SURVEY
                (GUEST)"); attend.setBounds(100, 350, 250, 40);
                frame.add(attend);
                attend.addActionListener(new ActionListener() {
                    @Override
                    public void actionPerformed(ActionEvent e) {
                        String surveyCode = JOptionPane.showInputDialog("Enter the
Survey Code : ");
                        try {
                            if(!surveyCode.isEmpty() && surveyCode.length() == 5) {
                                if(manage.check(surveyCode)) {
                                    Guest guest = new Guest();
                                    guest.guestView(surveyCode);
                                }
                                else {
                                    JOptionPane.showMessageDialog(frame,
"No Survey Available!!!", "Warning Message", JOptionPane.WARNING_MESSAGE);
                                }
                            }
                        }
                        catch(Exception e2) {

                        }
                    }
                });

                frame.setVisible(true);
        }
}
```

The main purpose of this Java code is to serve as the entry point for the online survey system application by initiating the login interface for users. It invokes the `loginView()` method from the `Login` class to display the login interface to users.

```
package survey;

import java.sql.SQLException;

public class Main {
	public static void main(String args[]) throws SQLException {
		Login login = new Login();
		login.loginView();
	}
}
```

The main purpose of this Java code is to create the main page interface for the online survey system application, allowing users to add and view surveys, as well as navigate through survey questions. It also facilitates survey creation, deletion, and user authentication functionalities.

```
package survey;

import java.awt.Color;
import java.awt.Font;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.KeyEvent;
import java.awt.event.KeyListener;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
import java.awt.event.MouseListener;
import java.nio.charset.Charset;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.Random;

import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JScrollPane;
import javax.swing.JTable;
```

```
import javax.swing.JTextField;
import javax.swing.table.DefaultTableModel;

public class MainPage {

	SQLManage manage;
	JButton submit;
	String[] questionsArray, option1Array, option2Array, option3Array, option4Array;
	static DefaultTableModel model;
	String cd;

	int i=0, h=0;
	String[] queStr = new String[50];
	String[] op1Str = new String[50];
	String[] op2Str = new String[50];
	String[] op3Str = new String[50];
	String[] op4Str = new String[50];
	int id;
	public void mainPageView(int id) throws SQLException
		{ this.id=id;
		questionsArray = new String[25];
		option1Array = new String[25];
		option2Array = new String[25];
		option3Array = new String[25];
		option4Array = new String[25];

		manage = new SQLManage();

		JFrame frame = new JFrame();
		frame.setSize(800, 600);
		frame.setLayout(null);
		frame.setLocationRelativeTo(null);

		//-------------------------------------ADD PANEL----------------------------------------------
		JPanel addPanel = new JPanel();
		addPanel.setBounds(250, 0, 550, 600);
		addPanel.setLayout(null);

		JLabel question = new JLabel("Question : ");
		question.setBounds(50, 100, 100, 20);
		addPanel.add(question);
		JTextField questionField = new JTextField();
		questionField.setBounds(50, 125, 450, 30);
		addPanel.add(questionField);
```

```
JLabel option1 = new JLabel("Option 1 : ");
option1.setBounds(50, 165, 100, 20);
addPanel.add(option1);
JTextField option1Field = new JTextField();
option1Field.setBounds(50, 190, 200, 30);
addPanel.add(option1Field);

JLabel option2 = new JLabel("Option 2 : ");
option2.setBounds(50, 230, 100, 20);
addPanel.add(option2);
JTextField option2Field = new JTextField();
option2Field.setBounds(50, 255, 200, 30);
addPanel.add(option2Field);

JLabel option3 = new JLabel("Option 3 : ");
option3.setBounds(50, 295, 100, 20);
addPanel.add(option3);
JTextField option3Field = new JTextField();
option3Field.setBounds(50, 320, 200, 30);
addPanel.add(option3Field);

JLabel option4 = new JLabel("Option 4 : ");
option4.setBounds(50, 360, 100, 20);
addPanel.add(option4);
JTextField option4Field = new JTextField();
option4Field.setBounds(50, 385, 200, 30);
addPanel.add(option4Field);

JLabel start = new JLabel("CREATE A SURVEY");
start.setBounds(0, 10, 550, 50);
start.setHorizontalAlignment(JLabel.CENTER);
start.setFont(new Font("Times New Roman", Font.BOLD,
40)); addPanel.add(start);

JButton next = new JButton("ADD QUESTION >");
next.setBounds(50, 440, 450, 35);
addPanel.add(next);
next.addActionListener(new ActionListener( ) {
        @Override
        public void actionPerformed(ActionEvent e) {
                String qn = questionField.getText();
                String op1 = option1Field.getText();
                String op2 = option2Field.getText();;
```

```
                                String op3 = option3Field.getText();;
                                String op4 = option4Field.getText();;
                                if(qn.equals("") || op1.equals("") || op2.equals("") || op3.equals("") ||
op4.equals("")) {
                                        JOptionPane.showMessageDialog(frame, "Please Enter
All Details!!!", "Warning Message", JOptionPane.WARNING_MESSAGE);
                                }
                                else {
                                        questionField.setText("");
                                        option1Field.setText("");
                                        option2Field.setText("");
                                        option3Field.setText("");
                                        option4Field.setText("");
                                        queStr[i] = qn;
                                        op1Str[i] = op1;
                                        op2Str[i] = op2;
                                        op3Str[i] = op3;
                                        op4Str[i] = op4;
                                        i++;
                                        submit.setEnabled(true);
                                }
                        }
                });

                submit = new JButton("SUBMIT");
                submit.setBounds(50, 490, 200, 50);
                submit.setEnabled(false);
                addPanel.add(submit);
                submit.addActionListener(new ActionListener( ) {
                        @Override
                        public void actionPerformed(ActionEvent e) {
                                String code = stringGenerator();
                                String qn = questionField.getText();
                                String op1 = option1Field.getText();
                                String op2 = option2Field.getText();;
                                String op3 = option3Field.getText();;
                                String op4 = option4Field.getText();;
                                if(!(qn.equals("") && op1.equals("") && op2.equals("") &&
op3.equals("") && op4.equals(""))) {
                                        JOptionPane.showMessageDialog(frame, "Last details are
not added. If not required, Please clear the fields!!!", "Warning
Message", JOptionPane.WARNING_MESSAGE);
                                }
                                else {
```

```
                                        if(i==0) {
                                                JOptionPane.showMessageDialog(frame, "No
Questions Added", "Warning Message", JOptionPane.WARNING_MESSAGE);
                                        }
                                        else {
                                                try {
                                                        manage.userQuestionAdd(id, code);
                                                        for(int j=0; j<i; j++) {
                                                                manage.newQuestion(code,
queStr[j], op1Str[j], op2Str[j], op3Str[j], op4Str[j]);
                                                        }

JOptionPane.showMessageDialog(frame, "Survey Added. Survey Code : " +
code, "Congradulations", JOptionPane.PLAIN_MESSAGE);
                                                }
                                                catch (SQLException e1) {
                                                        e1.printStackTrace();
                                                }
                                        }
                                }
                                submit.setEnabled(false);
                        }
                });

                JButton cancel = new JButton("CANCEL");
                cancel.setBounds(300, 490, 200, 50);
                addPanel.add(cancel);
                cancel.addActionListener(new ActionListener( ) {
                        @Override
                        public void actionPerformed(ActionEvent e) {
                                questionField.setText("");
                                option1Field.setText("");
                                option2Field.setText("");
                                option3Field.setText("");
                                option4Field.setText("");
                                i=0;
                        }
                });

                frame.add(addPanel);
                //-----------------------------------------------------------------------------------------------

                //-------------------------------------------------VIEW
PANEL-----------------------------------------
```

```
JPanel viewPanel = new JPanel();
viewPanel.setBounds(250, 0, 550, 600);
viewPanel.setLayout(null);
JLabel searchLabel = new JLabel("Search : ");
searchLabel.setBounds(100, 20, 100, 50);
viewPanel.add(searchLabel);

JTextField search = new JTextField();
search.setBounds(160, 30, 290, 30);
viewPanel.add(search);
search.addKeyListener(new KeyListener() {
	@Override
	public void keyReleased(KeyEvent e) {
		tblupdt(search.getText());
	}

	@Override
	public void keyTyped(KeyEvent e) {
		//To avoid errors.
	}

	@Override
	public void keyPressed(KeyEvent e) {
		//To avoid errors.
	}
});

JTable table=new JTable(){
	public boolean isCellEditable(int row,int
		column){ return false;
	}
};
model = (DefaultTableModel)table.getModel();
table.setBackground(Color.decode("#f9d6c4"));
model.addColumn("Your Surveys");
tblupdt("");
JScrollPane scPane=new JScrollPane(table);
scPane.setBounds(100, 70, 350, 225);
viewPanel.add(scPane);

JLabel quesView = new JLabel();
quesView.setBounds(50, 340, 150, 50);
viewPanel.add(quesView);
```

```
JLabel op1View = new JLabel();
op1View.setBounds(70, 380, 150, 50);
viewPanel.add(op1View);

JLabel op2View = new JLabel();
op2View.setBounds(70, 420, 150, 50);
viewPanel.add(op2View);

JLabel op3View = new JLabel();
op3View.setBounds(70, 460, 150, 50);
viewPanel.add(op3View);

JLabel op4View = new JLabel();
op4View.setBounds(70, 500, 150, 50);
viewPanel.add(op4View);

JLabel op1Select = new JLabel();
op1Select.setBounds(100, 400, 150, 50);
viewPanel.add(op1Select);

JLabel op2Select = new JLabel();
op2Select.setBounds(100, 440, 150, 50);
viewPanel.add(op2Select);

JLabel op3Select = new JLabel();
op3Select.setBounds(100, 480, 150, 50);
viewPanel.add(op3Select);

JLabel op4Select = new JLabel();
op4Select.setBounds(100, 520, 150, 50);
viewPanel.add(op4Select);

JButton delete = new JButton("DELETE");
delete.setBounds(210, 300, 130, 50);
delete.setEnabled(false);
viewPanel.add(delete);

JButton viewPrev = new JButton("PREVIOUS");
viewPrev.setBounds(100, 300, 100, 50);
viewPrev.setEnabled(false);
viewPanel.add(viewPrev);
viewPrev.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
```

```
                        if(h>=0) {
                                h--;
                                quesView.setText(questionsArray[h]);
                                op1View.setText(option1Array[h]);
                                op2View.setText(option2Array[h]);
                                op3View.setText(option3Array[h]);
                                op4View.setText(option4Array[h]);
                                try {

op1Select.setText(String.valueOf(manage.getCount(cd, h, 1)));

op2Select.setText(String.valueOf(manage.getCount(cd, h, 2)));

op3Select.setText(String.valueOf(manage.getCount(cd, h, 3)));

op4Select.setText(String.valueOf(manage.getCount(cd, h, 4)));
                                } catch (SQLException e1) {
                                        e1.printStackTrace();
                                }
                        }
                }
        });

        delete.addActionListener(new ActionListener() {
                @Override
                public void actionPerformed(ActionEvent e) {
                        try {
                                manage.removeSurvey(cd);
                          } catch (SQLException e1) {
                                e1.printStackTrace();
                        }
                        tblupdt(search.getText());
                }
        });

        JButton viewNext = new JButton("NEXT");
        viewNext.setBounds(350, 300, 100, 50);
        viewNext.setEnabled(false);
        viewPanel.add(viewNext);
        viewNext.addActionListener(new ActionListener() {
                @Override
                public void actionPerformed(ActionEvent e) {
                        h++;
                        quesView.setText(questionsArray[h]);
```

```
                        op1View.setText(option1Array[h]);
                        op2View.setText(option2Array[h]);
                        op3View.setText(option3Array[h]);
                        op4View.setText(option4Array[h]);
                        try {
                                op1Select.setText(String.valueOf(manage.getCount(cd, h,
1)));
                                op2Select.setText(String.valueOf(manage.getCount(cd, h,
2)));
                                op3Select.setText(String.valueOf(manage.getCount(cd, h,
3)));
                                op4Select.setText(String.valueOf(manage.getCount(cd, h,
4)));
                        } catch (SQLException e1) {
                                e1.printStackTrace();
                        }
                }
        });


        table.addMouseListener(new MouseAdapter() {
                @Override
                public void mouseClicked(MouseEvent e) {
                        h=0;
                        delete.setEnabled(true);
                        viewNext.setEnabled(true);
                        viewPrev.setEnabled(true);
                        int row = table.getSelectedRow();
                        cd = String.valueOf(model.getValueAt(row, 0));
                        try {
                                ResultSet rst1 = manage.getQuestions(cd);
                                for(int x=0; rst1.next(); x++) {
                                        questionsArray[x] = rst1.getString("question");
                                        option1Array[x] = rst1.getString("option1");
                                        option2Array[x] = rst1.getString("option2");
                                        option3Array[x] = rst1.getString("option3");
                                        option4Array[x] = rst1.getString("option4");
                                }
                                op1Select.setText(String.valueOf(manage.getCount(cd, h,
1)));
                                op2Select.setText(String.valueOf(manage.getCount(cd, h,
2)));
                                op3Select.setText(String.valueOf(manage.getCount(cd, h,
3)));
```

```
                                        op4Select.setText(String.valueOf(manage.getCount(cd, h,
4)));
                                }
                                catch (SQLException e1) {
                                        e1.printStackTrace();
                                }
                                quesView.setText(questionsArray[h]);
                                op1View.setText(option1Array[h]);
                                op2View.setText(option2Array[h]);
                                op3View.setText(option3Array[h]);
                                op4View.setText(option4Array[h]);
                        }
                });


                frame.add(viewPanel);


//-------------------------------------------------------------------------------------------------------------

                //-----------------------------------------------SIDE
PANEL----------------------------------------------------
                JPanel optionPanel = new JPanel();
                optionPanel.setBounds(0, 0, 250, 600);
                optionPanel.setBackground(Color.gray);
                optionPanel.setLayout(null);
                frame.add(optionPanel);

                JButton addSurvey = new JButton("ADD
                SURVEY"); addSurvey.setBounds(50, 113, 150,
                50); optionPanel.add(addSurvey);
                addSurvey.addActionListener(new ActionListener() {
                        @Override
                        public void actionPerformed(ActionEvent e) {
                                viewPanel.setVisible(false);
                                addPanel.setVisible(true);
                        }
                });

                JButton viewSurvey = new JButton("VIEW
                SURVEY"); viewSurvey.setBounds(50, 276, 150, 50);
                optionPanel.add(viewSurvey);
                viewSurvey.addActionListener(new ActionListener() {
                        @Override
```

```
                public void actionPerformed(ActionEvent e) {
                        tblupdt(search.getText());
                        viewPanel.setVisible(true);
                        addPanel.setVisible(false);
                }
        });

        JButton logout = new JButton("LOGOUT");
        logout.setBounds(50, 440, 150, 50);
        optionPanel.add(logout);
        logout.addActionListener(new ActionListener() {
                @Override
                public void actionPerformed(ActionEvent e) {
                        Login login = new Login();
                        try {
                                login.loginView();
                          } catch (SQLException e1) {
                                e1.printStackTrace();
                        }
                        frame.dispose();
                }
        });

        viewPanel.setVisible(false);

        frame.setVisible(true);
}

public String stringGenerator() {
          String AlphaNumericString = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
                String sb = "";
                for (int i = 0; i < 5; i++) {
                 int index = (int)(AlphaNumericString.length() *
                Math.random()); sb += (AlphaNumericString.charAt(index)); }

        return sb;
}

public void tblupdt(String str) {
        try {
                SQLManage man = new SQLManage();
                ResultSet res = man.surveys(id, str);
                int rowCount = model.getRowCount();
```

```
                        int i;
                        for (i = rowCount - 1; i >= 0; i--)
                           model.removeRow(i);
                        for(i=0; res.next(); i++) {
                                model.addRow(new Object[0]);
                      model.setValueAt(res.getString("surveycode"), i, 0);
                        }
                } catch (SQLException e1) {
                        e1.printStackTrace();
                }
        }

}
```

The main purpose of this Java code is to manage the interaction with the MySQL database for the online survey system application. It includes functionalities such as user authentication, adding new users, adding survey questions, updating survey responses, retrieving survey questions, managing surveys, and retrieving response counts.

```
package survey;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

public class SQLManage {

        Connection con;

        public SQLManage() throws SQLException {
                String url = "jdbc:mysql://localhost:3306/survey";
                String usr = "root";
                String pass = "root";
                con = DriverManager.getConnection(url, usr, pass);
        }

        public void newUser(String name, String uname, String pass) throws SQLException
                { String str = "INSERT INTO actors(fname, uname, pass) values ('"+name+"',
'"+uname+"', '"+pass+"')";
                Statement stm = con.createStatement();
                stm.executeUpdate(str);
        }
```

```
	public int authUser(String uname, String pass) throws SQLException { String
		str = "SELECT * FROM actors WHERE uname = '"+uname+"'";
		Statement stm = con.createStatement();
		ResultSet rst = stm.executeQuery(str);
		if (!rst.next())
			return -1;
		else {
			if(rst.getString("pass").equals(pass))
				return rst.getInt("id");
			else
				return 0;
		}
	}

	public void newQuestion(String code, String question, String op1, String op2, String op3,
String op4) throws SQLException {
		String str = "INSERT INTO questions values ('"+code+"', '"+question+"',
'"+op1+"', '"+op2+"', '"+op3+"', '"+op4+"')";
		Statement stm = con.createStatement();
		stm.executeUpdate(str);
	}

	public void userQuestionAdd(int id, String surveycode) throws SQLException {
		String str = "INSERT INTO userQuestions values ("+id+", '"+surveycode+"', 0)";
		Statement stm = con.createStatement();
		stm.executeUpdate(str);
	}

	public void answerUpdt(String surveycode, int qno, int option) throws SQLException { String
		str = "INSERT INTO surveyquestions values ('"+surveycode+"', " + qno + ",
" + option + ")";
		Statement stm = con.createStatement();
		stm.executeUpdate(str);
	}

	public ResultSet getQuestions(String surveycode) throws SQLException
		{ String str = "SELECT * FROM questions WHERE surveycode =
'"+surveycode+"'";
		Statement stm = con.createStatement();
		ResultSet rst = stm.executeQuery(str);
		return rst;
	}
```

```
	public ResultSet surveys(int id, String search) throws SQLException {
		String str = "SELECT * FROM userQuestions WHERE id = "+id+" and
surveycode like '%"+search+"%'";
		Statement stm = con.createStatement();
		ResultSet rst = stm.executeQuery(str);
		return rst;
	}

	public void addTotal() throws SQLException {
		String str = "UPDATE userQuestions SET total =
		total+1"; Statement stm = con.createStatement();
		stm.executeUpdate(str);
	}

	public boolean check(String search) throws SQLException {
		String str = "SELECT * FROM userQuestions WHERE surveycode =
'"+search+"'";
		Statement stm = con.createStatement();
		ResultSet rst = stm.executeQuery(str);
		if(rst.next())
			return true;
		else
			return false;
	}

	public void removeSurvey(String surveycode) throws SQLException {
		String str = "DELETE FROM questions WHERE surveycode = '"+surveycode+"'";
		Statement stm = con.createStatement();
		stm.executeUpdate(str);
		str = "DELETE FROM surveyquestions WHERE surveycode =
		'"+surveycode+"'"; stm.executeUpdate(str);
		str = "DELETE FROM userQuestions WHERE surveycode = '"+surveycode+"'";
		stm.executeUpdate(str);
	}

	public int getCount(String surveycode, int qno, int op) throws SQLException {
		String str = "SELECT count(opno) FROM surveyquestions WHERE surveycode =
'"+surveycode+"' AND qno = "+(qno+1)+" AND opno = "+op;
		Statement stm = con.createStatement();
		ResultSet rst = stm.executeQuery(str);
		if(rst.next())
			return rst.getInt("count(opno)");
		else
			return 0;
```

```
        }

}
```

This code is to provide a graphical user interface (GUI) for user registration (sign-up) in the survey system application. It allows users to enter their name, username, and password, and upon submission, it registers the user in the system after performing validation checks such as ensuring all fields are filled and matching the passwords.

```
 package survey;

import java.awt.Font;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.SQLException;
import java.util.logging.Level;
import java.util.logging.Logger;

import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPasswordField;
import javax.swing.JTextField;

public class SignUp {
        public void signUpView() {
                JFrame frame = new JFrame();
                frame.setSize(450, 450);
                frame.setLayout(null);
                frame.setLocationRelativeTo(null);

                JLabel heading = new JLabel("SURVEY SYSTEM");
                heading.setBounds(0, 50, 450, 50);
                heading.setHorizontalAlignment(JLabel.CENTER);
                heading.setFont(new Font("Times New Roman", Font.BOLD,
                40)); frame.add(heading);

                JLabel fName = new JLabel("Name : ");
                fName.setBounds(50, 120, 150, 50);
                frame.add(fName);

                JTextField fNameField = new JTextField();
```

```
fNameField.setBounds(50, 160, 350, 30);
frame.add(fNameField);

JLabel uName = new JLabel("Username : ");
uName.setBounds(50, 190, 150, 50);
frame.add(uName);

JTextField uNameField = new JTextField();
uNameField.setBounds(50, 230, 350, 30);
frame.add(uNameField);

JLabel uPass = new JLabel("Password : ");
uPass.setBounds(50, 260, 150, 50);
frame.add(uPass);

JPasswordField uPassField = new JPasswordField();
uPassField.setBounds(50, 300, 150, 30);
frame.add(uPassField);

JLabel uPass2 = new JLabel("Confirm Password :
"); uPass2.setBounds(250, 260, 150, 50);
frame.add(uPass2);

JPasswordField uPassField2 = new
JPasswordField(); uPassField2.setBounds(250, 300,
150, 30); frame.add(uPassField2);

JButton submit = new JButton("SUBMIT");
submit.setBounds(175, 350, 100, 40);
frame.add(submit);
submit.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
                String fname = fNameField.getText();
                String uname = uNameField.getText();
                String pass1 = uPassField.getText();
                String pass2 = uPassField2.getText();
                if(fname.isEmpty() || uname.isEmpty() || pass1.isEmpty() ||
pass2.isEmpty()) {
                        JOptionPane.showMessageDialog(frame, "Please Enter
All Details!!!", "Warning Message", JOptionPane.WARNING_MESSAGE);
                }
                else {
                        if(pass1.equals(pass2)) {
```

```
                                SQLManage manage = null;
                                try {
                                    manage = new SQLManage();
                                } catch (SQLException ex) {
                                    Logger.getLogger(SignUp.class.getName()).log(Level.SEVERE,
null, ex);
                                }
                                try {
                                    manage.newUser(fname, uname, pass1);
                                } catch (SQLException ex) {
                                    Logger.getLogger(SignUp.class.getName()).log(Level.SEVERE,
null, ex);
                                }
                                fNameField.setText("");
                                uNameField.setText("");
                                uPassField.setText("");
                                uPassField2.setText("");
                                JOptionPane.showMessageDialog(frame, "User Created
Succesfully!!!", "User Created", JOptionPane.PLAIN_MESSAGE);
                                frame.dispose();

                                        }
                                        else {
                                                JOptionPane.showMessageDialog(frame,
"Password Mismatch", "Warning Message", JOptionPane.WARNING_MESSAGE);
                                        }
                                    }
                                }
                        });

                        frame.setVisible(true);
                }
}
```

Future Scope of the ATM Interface Project:

The future scope of an online survey system project involves implementing advanced features to enhance its functionality, usability, and effectiveness. This includes features such as advanced survey creation options with support for various question types and multimedia integration, robust user management capabilities including role-based access control and user registration, comprehensive survey analytics and reporting tools for real-time data analysis, integration with CRM and other systems to streamline data exchange, mobile compatibility for convenient participation, enhanced security measures to protect sensitive data, multi-language support for a diverse audience, and scalability and performance optimizations to hand increased

usage. Additionally, leveraging AI and machine learning techniques for advanced data analysis and predictive analytics can further enrich the capabilities of the system. These enhancements will ensure that the online survey system remains adaptable and responsive to the evolving needs of organizations and users.

## Conclusion

The conclusion of an online survey system project highlights the successful development and deployment of a versatile platform for conducting surveys and gathering valuable insights. Through careful planning, implementation, and testing, the project has delivered a user-friendly interface for creating, distributing, and analyzing surveys efficiently. The system offers flexibility in survey design, robust user management features, comprehensive analytics tools, and integration capabilities with other systems. With a focus on security, scalability, and performance, the online survey system ensures data integrity and user satisfaction. As a result, the project concludes with the delivery of a reliable and adaptable solution that meets the needs of organizations and users alike, facilitating informed decision-making and driving continuous improvement initiatives.