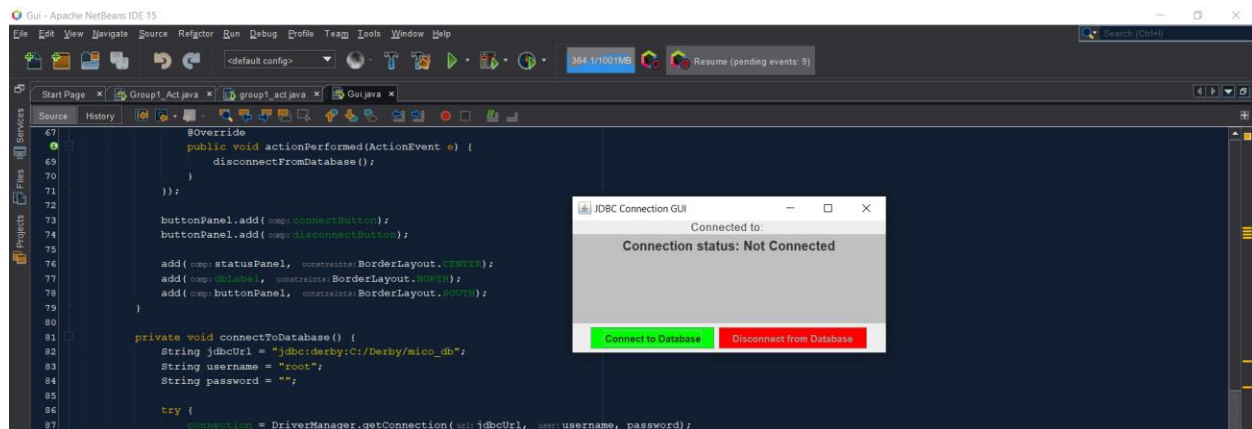


RABINO, MICO T.

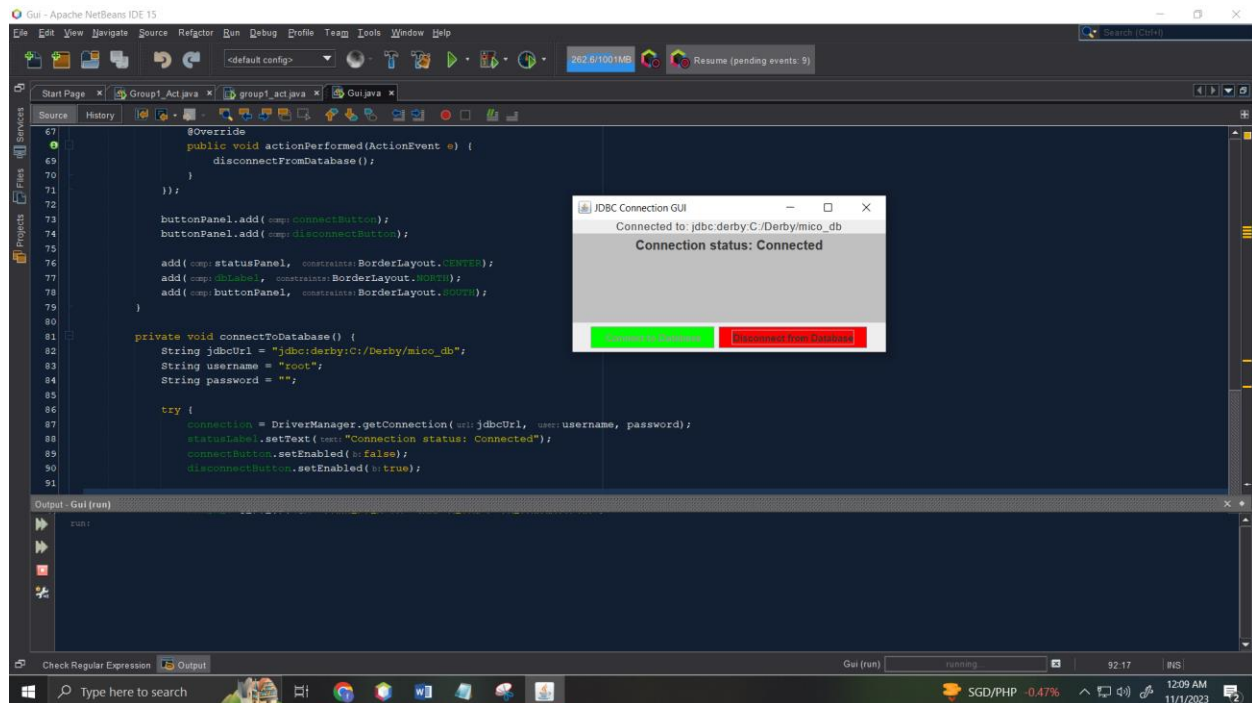
BSIT-S-2A

MACHINE PROBLEM 2 (GROUP 2) SCREENSHOTS AND PDF

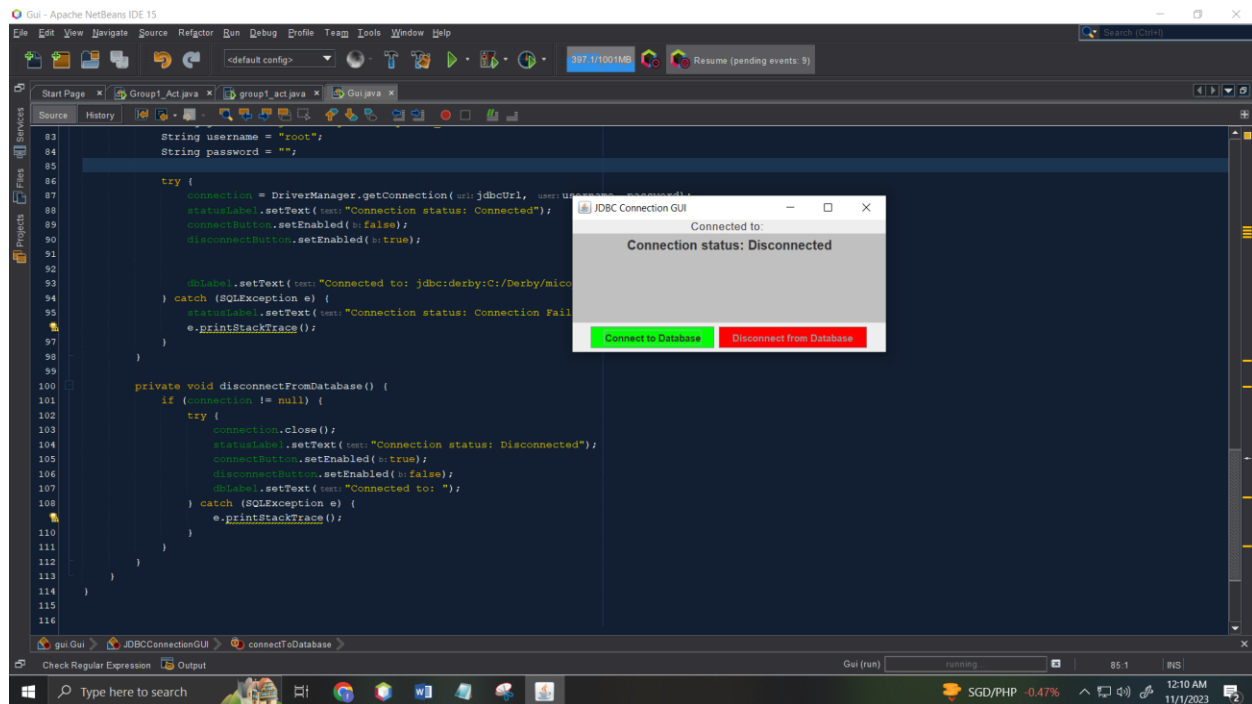
GUI:



CONNECT BTN:



DISCONNECT BTN:



SOURCE CODE:

/*

* Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license

* Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Main.java to edit this template

*/

package gui;

import javax.swing.*;

import java.awt.*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.SQLException;

```
public class Gui {  
    public static void main(String[] args) {  
        SwingUtilities.invokeLater(() -> {  
            JDBCConnectionGUI gui = new JDBCConnectionGUI();  
            gui.setVisible(true);  
        });  
    }  
}
```

```
public static class JDBCConnectionGUI extends JFrame {  
    private JLabel statusLabel;  
    private JLabel dbLabel;  
    private JButton connectButton;  
    private JButton disconnectButton;  
    private Connection connection;  
  
    public JDBCConnectionGUI() {  
        setTitle("JDBC Connection GUI");  
        setSize(400, 200);  
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        setLayout(new BorderLayout());  
        setBackground(Color.WHITE);  
  
        JPanel statusPanel = new JPanel();  
        statusPanel.setLayout(new FlowLayout(FlowLayout.CENTER));  
        statusLabel = new JLabel("Connection status: Not Connected");  
        statusLabel.setHorizontalAlignment(SwingConstants.CENTER);  
        statusLabel.setFont(new Font("Arial", Font.BOLD, 16));  
        statusPanel.add(statusLabel);  
        statusPanel.setBackground(Color.LIGHT_GRAY);  
    }  
}
```

```
dbLabel = new JLabel("Connected to ");
dbLabel.setHorizontalAlignment(SwingConstants.CENTER);
dbLabel.setFont(new Font("Arial", Font.PLAIN, 14));

JPanel buttonPanel = new JPanel();
buttonPanel.setLayout(new FlowLayout(FlowLayout.CENTER));
connectButton = new JButton("Connect to Database");
disconnectButton = new JButton("Disconnect from Database");
disconnectButton.setEnabled(false);

connectButton.setBackground(Color.GREEN);
disconnectButton.setBackground(Color.RED);

connectButton.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        connectToDatabase();
    }
});

disconnectButton.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        disconnectFromDatabase();
    }
});
```

```

        buttonPanel.add(connectButton);
        buttonPanel.add(disconnectButton);

        add(statusPanel, BorderLayout.CENTER);
        add(dbLabel, BorderLayout.NORTH);
        add(buttonPanel, BorderLayout.SOUTH);
    }

    private void connectToDatabase() {
        String jdbcUrl = "jdbc:derby:C:/Derby/mico_db";
        String username = "root";
        String password = "";

        try {
            connection = DriverManager.getConnection(jdbcUrl, username, password);
            statusLabel.setText("Connection status: Connected");
            connectButton.setEnabled(false);
            disconnectButton.setEnabled(true);

            java.sql.DatabaseMetaData metaData = connection.getMetaData();
            dbLabel.setText("Connected to: " + metaData.getURL());
        } catch (SQLException e) {
            statusLabel.setText("Connection status: Connection Failed");
            e.printStackTrace();
        }
    }

    private void disconnectFromDatabase() {
        if (connection != null) {
            try {

```

```
connection.close();
statusLabel.setText("Connection status: Disconnected");
connectButton.setEnabled(true);
disconnectButton.setEnabled(false);
dbLabel.setText("Connected to: ");
} catch (SQLException e) {
    e.printStackTrace();
}
}
}
}
```