CODE:

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.sql.*;
public class Main {
// GUI components
    private JFrame mainWindow;
    private JPanel inputPanel;
    private JTextField idField, nameField, sodiumField, potassiumField, lactoseField;
    private JButton checkButton;
    private JTextArea chatArea;
    private JTextField userInput;
// Database connection details
    private static final String URL =
"jdbc:mysql://localhost:3306/HEALTHs?serverTimezone=UTC";
    private static final String USER = "root";
    private static final String PASSWORD = "1410";
 // Application constructor
    public Main() {
        initializeUI();
        setupEventListeners();
    }
 // Initialize the user interface
    private void initializeUI() {
        // Configure main window
        mainWindow = new JFrame("Health Monitoring System");
        mainWindow.setSize(1200, 700);
```

```
        mainWindow.setLocationRelativeTo(null);
        mainWindow.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        mainWindow.setResizable(false);
       // Initialize input panel
        inputPanel = new JPanel();
        inputPanel.setLayout(new GridLayout(7, 2, 10, 10));
        inputPanel.setBackground(new Color(240, 248, 255));
        // Add input fields with labels
        addLabelAndField("Enter ID:", idField = createInputField());
        addLabelAndField("Enter Name:", nameField = createInputField());
        addLabelAndField("Enter Sodium Level (mg/L):", sodiumField = createInputField());
        addLabelAndField("Enter Potassium Level (mg/L):", potassiumField =
createInputField());
        addLabelAndField("Enter Lactose Level (mg/L):", lactoseField = createInputField());
        // Add check button
        checkButton = new JButton("Check");
        formatButton(checkButton, new Color(255, 182, 193));
        inputPanel.add(new JLabel(""));
        inputPanel.add(checkButton);
        // Add panel to frame and display
        mainWindow.add(inputPanel);
        mainWindow.setVisible(true);
    }
    // Helper method to add label and field to panel
    private void addLabelAndField(String labelText, JTextField field) {
        inputPanel.add(createStyledLabel(labelText));
        inputPanel.add(field);
    }
```

```
// Configure event handlers
    private void setupEventListeners() {
        checkButton.addActionListener(e -> processInputAndSaveData());
    }
    // Create styled text labels
    private JLabel createStyledLabel(String text) {
        JLabel label = new JLabel(text);
        label.setFont(new Font("Arial", Font.BOLD, 22));
        label.setForeground(Color.BLACK);
        return label;
    }
     // Create styled text input fields
    private JTextField createInputField() {
        JTextField field = new JTextField();
        field.setFont(new Font("Arial", Font.BOLD, 20));
        return field;
    }
// Apply styling to buttons
    private void formatButton(JButton button, Color backgroundColor) {
        button.setBackground(backgroundColor);
        button.setForeground(Color.WHITE);
        button.setFont(new Font("Arial", Font.BOLD, 20));
    }

    // Process user input, perform calculations and store to database
    private void processInputAndSaveData() {
        try {
            // Retrieve and parse user input
```

```
            int id = Integer.parseInt(idField.getText());
            String name = nameField.getText();
            float sodium = Float.parseFloat(sodiumField.getText());
            float potassium = Float.parseFloat(potassiumField.getText());
            float lactose = Float.parseFloat(lactoseField.getText());

            // Calculate health metrics using formulas
            float glucose = calculateGlucose(sodium, potassium, lactose);
            float hemoglobin = calculateHemoglobin(sodium, potassium);
            float cholesterol = calculateCholesterol(sodium, lactose);

            // Store data in database
            storeHealthData(id, name, sodium, potassium, lactose, glucose, hemoglobin,
cholesterol);

            // Display results to user
            displayHealthResults(glucose, hemoglobin, cholesterol);
        } catch (Exception ex) {
            showErrorMessage("Invalid Input! Please enter correct values.");
        }
    }
    // Formula for glucose calculation
    private float calculateGlucose(float sodium, float potassium, float lactose) {
        return (sodium * 1.5f) + (potassium * 2) + (lactose * 0.5f);
    }

    // Formula for hemoglobin calculation
    private float calculateHemoglobin(float sodium, float potassium) {
```

```
        return (sodium * 0.8f) + (potassium * 1.2f);
    }

    // Formula for cholesterol calculation
    private float calculateCholesterol(float sodium, float lactose) {
        return (sodium * 2) + (lactose * 1.5f);
    }

    // Show error message dialog
    private void showErrorMessage(String message) {
        JOptionPane.showMessageDialog(mainWindow, "  " + message, "Error",
JOptionPane.ERROR_MESSAGE);
    }

    // Display calculated health metrics
    private void displayHealthResults(float glucose, float hemoglobin, float cholesterol) {
        JFrame resultsWindow = new JFrame("Health Results");
        resultsWindow.setSize(600, 400);
        resultsWindow.setLocationRelativeTo(null);

        JPanel resultsPanel = new JPanel(new GridLayout(4, 3, 10, 10));
        resultsPanel.setBackground(new Color(230, 255, 230));

        // Add glucose result row
        resultsPanel.add(createStyledLabel("Glucose Level (mg/L):"));
        resultsPanel.add(createResultDisplay(glucose, 70, 140));
        resultsPanel.add(createStyledLabel("Normal: 70-140 mg/L"));
```

```
    // Add hemoglobin result row
    resultsPanel.add(createStyledLabel("Hemoglobin Level (g/L):"));
    resultsPanel.add(createResultDisplay(hemoglobin, 12, 17));
    resultsPanel.add(createStyledLabel("Normal: 12-17 g/L"));

    // Add cholesterol result row
    resultsPanel.add(createStyledLabel("Cholesterol Level (mg/L):"));
    resultsPanel.add(createResultDisplay(cholesterol, 125, 200));
    resultsPanel.add(createStyledLabel("Normal: 125-200 mg/L"));

    resultsWindow.add(resultsPanel);
    resultsWindow.setVisible(true);
}

// Create a label for displaying health metric results with color coding
private JTextField createResultDisplay(float value, float minNormal, float maxNormal) {
    JTextField field = new JTextField(String.format("%.2f", value));
    field.setEditable(false);
    field.setFont(new Font("Arial", Font.BOLD, 20));
    field.setHorizontalAlignment(JTextField.CENTER);

    // Color code based on normal range
    if (value < minNormal || value > maxNormal) {
        field.setBackground(new Color(255, 200, 200)); // Light red for abnormal
    } else {
        field.setBackground(new Color(200, 255, 200)); // Light green for normal
    }
```

```
        return field;
    }
    // Save health data to database
    private void storeHealthData(int id, String name, float sodium, float potassium,
                                 float lactose, float glucose, float hemoglobin, float cholesterol) {
        try {
            // Establish database connection
            Connection connection = DriverManager.getConnection(URL, USER, PASSWORD);

            // Prepare SQL statement
            String insertQuery = "INSERT INTO health_data (id, name, sodium, potassium, lactose,
" +
                                 "glucose, hemoglobin, cholesterol) VALUES (?, ?, ?, ?, ?, ?, ?, ?)";
            PreparedStatement statement = connection.prepareStatement(insertQuery);

            // Set parameters and execute
            statement.setInt(1, id);
            statement.setString(2, name);
            statement.setFloat(3, sodium);
            statement.setFloat(4, potassium);
            statement.setFloat(5, lactose);
            statement.setFloat(6, glucose);
            statement.setFloat(7, hemoglobin);
            statement.setFloat(8, cholesterol);
            statement.executeUpdate();

            // Close resources
            statement.close();
```

```
            connection.close();

            // Inform user of success
            JOptionPane.showMessageDialog(mainWindow, "  Data Saved Successfully!",
                                          "Success", JOptionPane.INFORMATION_MESSAGE);
        } catch (SQLException e) {
            showErrorMessage("Database Error: " + e.getMessage());
        }
    }

    // Process user chat input (placeholder for chatbot functionality)
    private void handleChatInput() {
        // Get user input and process
        String message = userInput.getText().toLowerCase();
        userInput.setText("");
        chatArea.append("You: " + message + "\n");

        // Generate appropriate response based on keywords
        String botResponse;
        if (message.contains("diet")) {
            botResponse = "Maintain a balanced diet with whole grains, proteins, and healthy
fats.";
        } else if (message.contains("exercise")) {
            botResponse = "Regular exercise helps improve overall health. Try a 30-minute walk
daily.";
        } else if (message.contains("glucose")) {
            botResponse = "Normal glucose levels range from 70-140 mg/L. Consume fiber-rich
foods to regulate levels.";
        } else if (message.contains("cholesterol")) {
```

```
            botResponse = "Normal cholesterol is 125-200 mg/L. Reduce saturated fats for better health.";
        } else if (message.contains("hemoglobin")) {
            botResponse = "Normal hemoglobin is 12-17 g/L. Consume iron-rich foods like spinach and lentils.";
        } else {
            botResponse = "I'm here to help! Ask about diet, glucose, cholesterol, or general health tips.";
        }

        // Display bot response
        chatArea.append("Bot: " + botResponse + "\n");
    }

    // Application entry point
    public static void main(String[] args) {
        SwingUtilities.invokeLater(() -> new Main());
    }
}
```