تقرير: شرح كود تطبيق Tracking management system في Java

مقدمة

يُعد هذا الكود تطبيقًا رسوميًا (GUI) باستخدام Java لإدارة بيانات الطلاب الخريجين عبر قاعدة بيانات MySQL. يوفر التطبيق إمكانيات مثل إضافة، تحديث، حذف، والبحث عن بيانات الطلاب. كما يُمكّن من عرض البيانات داخل جدول رسومي. يعتمد الكود على مكتبات Swing لإنشاء الواجهة الرسومية ومكتبات JDBC للاتصال بقاعدة البيانات.

مكونات الكود

1.استيراد المكتبات

```
import javax.swing.*;
import javax.swing.table.DefaultTableModel;
import java.awt.event.*;
import java.sql.*;
import java.awt.Color;
import java.awt.Font;
import java.awt.Toolkit;
```

- **javax.swing:** لإنشاء المكونات الرسومية مثل الأزرار والنوافذ.
- **DefaultTableModel:** لإدارة بيانات الجدول.
- **java.awt:** لتحديد الألوان والخطوط.
- **java.awt.event:** لإضافة الأحداث مثل النقر على الأزرار.
- **java.sql:** لإجراء عمليات الاتصال بقاعدة البيانات.

2.الاتصال بقاعدة البيانات

```
try {
    Class.forName("com.mysql.cj.jdbc.Driver");
    con =
DriverManager.getConnection("jdbc:mysql://localhost/tracking_system", "root",
"");
    JOptionPane.showMessageDialog(null, "تم الاتصال بنجاح");
} catch (Exception ex) {
    ex.printStackTrace();
}
```

- يتم تحميل درايفر MySQL.
- يتم الاتصال بقاعدة البيانات باستخدام اسم المستخدم "root" وكلمة مرور فارغة.
- يتم عرض رسالة تأكيد عند نجاح الاتصال.

3.إعداد واجهة المستخدم(GUI)

إنشاء نافذة التطبيق

```
JFrame CRUD = new JFrame("Graduate Student Form");
CRUD.setIconImage(Toolkit.getDefaultToolkit().getImage("C:\\Users\\DUBAI_KEY\
\Desktop\\1674050617804.jpeg"));
CRUD.getContentPane().setBackground(new Color(132, 207, 236));
CRUD.setBounds(500, 30, 700, 800);
CRUD.getContentPane().setLayout(null);
```

- يتم إنشاء نافذة باسم."Graduate Student Form"
- يتم تحديد أيقونة خاصة، لون الخلفية، وحجم النافذة.

إضافة الحقول النصية

```
JLabel nameLabel = new JLabel("Student Name");
JTextField nameField = new JTextField();
CRUD.getContentPane().add(nameLabel);
CRUD.getContentPane().add(nameField);
```

- يتم إنشاء مكونات رسومية مثل **Labels** لعرض النصوص و **TextFields** لإدخال البيانات.

إضافة الجدول

```
String[] columns = {"StudentID", "Name", "Phone", "Address", "Department",
"AvailabilityStatus", "LastCommunicationDate"};
table = new JTable(new DefaultTableModel(columns, 0));
JScrollPane scrollPane = new JScrollPane(table);
CRUD.getContentPane().add(scrollPane);
```

- يتم إنشاء جدول يحتوي على أعمدة مخصصة لعرض البيانات.
- يُضاف الجدول داخل ScrollPane لتمكين التمرير العمودي.

إضافة الأزرار

- **زر البحث :(Search)**

```
JButton searchButton = new JButton("Search");
searchButton.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        // منطق البحث
    }
});
CRUD.getContentPane().add(searchButton);
```

- **زر الإضافة:(Add Graduate)**

```
JButton addButton = new JButton("Add Graduate");
addButton.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        // منطق الإضافة
    }
});
CRUD.getContentPane().add(addButton);
```

- **زر التحديث :(Update Graduate)**

```
JButton updateButton = new JButton("Update Graduate");
updateButton.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        // منطق التحديث
    }
});
CRUD.getContentPane().add(updateButton);
```

- **زر الحذف :(Delete Graduate)**

```
JButton deleteButton = new JButton("Delete Graduate");
deleteButton.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        // منطق الحذف
    }
});
CRUD.getContentPane().add(deleteButton);
```

4.أحداث الأزرار

زر البحث

```
pst = con.prepareStatement("SELECT Name, Phone, Address, Department,
AvailabilityStatus, LastCommunicationDate FROM graduatestudent WHERE
StudentID = ?");
pst.setString(1, id);
ResultSet rs = pst.executeQuery();
if (rs.next()) {
    nameField.setText(rs.getString("Name"));
                                                    // باقي الحقول
                                                    }
```

- يتم تنفيذ استعلام **SELECT** لجلب بيانات الطالب وعرضها في الحقول النصية.

زر الإضافة

```
        pst = con.prepareStatement("INSERT INTO graduatestudent (Name, Phone,
 Address, Department, AvailabilityStatus, LastCommunicationDate) VALUES (?, ?,
                                                              ?, ?, ? , ?)");
                                                      pst.setString(1, name);
                                                                //باقي القيم
                                                     pst.executeUpdate();
```

- يتم تنفيذ استعلام **INSERT** لإضافة بيانات جديدة إلى قاعدة البيانات.

زر التحديث

```
   pst = con.prepareStatement("UPDATE graduatestudent SET Name = ?, Phone = ?,
  Address = ?, Department = ?, AvailabilityStatus = ?, LastCommunicationDate =
                                                    ? WHERE StudentID = ?");
                                                     pst.setString(1, name);
                                                               //باقي القيم
                                                    pst.executeUpdate();
```

- يتم تنفيذ استعلام **UPDATE** لتحديث بيانات الطالب.

زر الحذف

```
   pst = con.prepareStatement("DELETE FROM graduatestudent WHERE StudentID =
                                                                        ?");
                                                          pst.setInt(1, id);
                                                       pst.executeUpdate();
```

- يتم تنفيذ استعلام **DELETE** لحذف بيانات الطالب.

---

5. تحميل البيانات في الجدول

```
private static void loadTableData() {
    DefaultTableModel model = (DefaultTableModel) table.getModel();
    model.setRowCount(0);
    Statement stmt = con.createStatement();
    ResultSet rs = stmt.executeQuery("SELECT * FROM graduatestudent");
    while (rs.next()) {
        Object[] row = new Object[7];
        row[0] = rs.getInt("StudentID");
        row[1] = rs.getString("Name");
        // باقي الأعمدة
        model.addRow(row);
    }
}
```

- يتم تنفيذ استعلام **SELECT** لجلب جميع السجلات وعرضها في الجدول.