import javax.swing.\*;

import java.awt.\*;

import java.sql.\*;

public class ese2 {

private JFrame frame;

private JTextField nameField, emailField, phoneField, contactIdField;

private JButton addButton, updateButton, deleteButton, searchButton, refreshButton;

private JTextArea displayArea;

private JScrollPane scrollPane;

static final String DB\_URL = "jdbc:mysql://localhost/";

static final String USER = "root";

static final String PASSWORD = "root";

static final String DATABASE\_NAME = "ese2DB";

public ese2() {

frame = new JFrame("Contact Manager");

frame.setSize(600, 400);

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

JPanel panel = new JPanel();

panel.setLayout(new BoxLayout(panel, BoxLayout.PAGE\_AXIS));

JLabel nameLabel = new JLabel("Name:");

nameField = new JTextField();

JLabel emailLabel = new JLabel("Email:");

emailField = new JTextField();

JLabel phoneLabel = new JLabel("Phone:");

phoneField = new JTextField();

JLabel contactIdLabel=new JLabel("ID");

contactIdField=new JTextField();

panel.add(contactIdLabel);

panel.add(contactIdField);

panel.add(nameLabel);

panel.add(nameField);

panel.add(emailLabel);

panel.add(emailField);

panel.add(phoneLabel);

panel.add(phoneField);

addButton = new JButton("Add Contact");

updateButton = new JButton("Update Contact");

deleteButton = new JButton("Delete Contact");

searchButton = new JButton("Search Contact");

refreshButton=new JButton("Refresh");

addButton.addActionListener(e -> addContact());

updateButton.addActionListener(e -> updateContact());

deleteButton.addActionListener(e -> deleteContact());

searchButton.addActionListener(e -> searchContact());

refreshButton.addActionListener(e-> retrieveContacts());

JPanel buttonPanel = new JPanel(new GridLayout(1, 5));

buttonPanel.add(addButton);

buttonPanel.add(updateButton);

buttonPanel.add(deleteButton);

buttonPanel.add(searchButton);

buttonPanel.add(refreshButton);

displayArea = new JTextArea(10, 30);

scrollPane = new JScrollPane(displayArea);

panel.add(buttonPanel);

panel.add(scrollPane);

frame.getContentPane().add(BorderLayout.CENTER, panel);

frame.setVisible(true);

initializeDatabase();

retrieveContacts();

clearFields();

}

private void initializeDatabase() {

try {

Class.forName("com.mysql.cj.jdbc.Driver");

Connection connection = DriverManager.getConnection(DB\_URL, USER, PASSWORD);

Statement statement = connection.createStatement();

statement.executeUpdate("CREATE DATABASE IF NOT EXISTS " + DATABASE\_NAME);

statement.executeUpdate("USE " + DATABASE\_NAME);

statement.executeUpdate("CREATE TABLE IF NOT EXISTS contacts (" +

"id INT AUTO\_INCREMENT PRIMARY KEY," +

"name VARCHAR(100) ," +

"email VARCHAR(100) ," +

"phone VARCHAR(20) )");

statement.close();

connection.close();

} catch (Exception e) {

e.printStackTrace();

}

}

private void addContact() {

String name = nameField.getText();

String email = emailField.getText();

String phone = phoneField.getText();

if (name.isEmpty() || email.isEmpty() || phone.isEmpty()) {

JOptionPane.showMessageDialog(frame, "Please enter all fields.", "Error", JOptionPane.ERROR\_MESSAGE);

return;

}else if(email.indexOf("@")==-1 || email.indexOf(".")==-1){

JOptionPane.showMessageDialog(frame, "Invalid Email", "Number Error", JOptionPane.ERROR\_MESSAGE);

return;

}else if(phone.length()!=10){

JOptionPane.showMessageDialog(frame, "Phone Number should have 10 digits", "Number Error", JOptionPane.ERROR\_MESSAGE);

return;

}

try {

Connection connection = DriverManager.getConnection(DB\_URL + DATABASE\_NAME, USER, PASSWORD);

PreparedStatement preparedStatement = connection.prepareStatement(

"SELECT \* FROM contacts WHERE phone=?");

preparedStatement.setString(1, phone);

ResultSet resultSet = preparedStatement.executeQuery();

StringBuilder contactsText = new StringBuilder();

while (resultSet.next()) {

int tid = resultSet.getInt("id");

String tname = resultSet.getString("name");

String temail = resultSet.getString("email");

String tphone = resultSet.getString("phone");

contactsText.append("ID: " + tid + "\t Name: " + tname + "\t\t Email: " + temail + "\t Phone: " + tphone + "\n");

}

if (contactsText.length() > 0) {

JOptionPane.showMessageDialog(frame, "Phone Number Already Exists", "Error", JOptionPane.ERROR\_MESSAGE);

} else {

preparedStatement = connection.prepareStatement("INSERT INTO contacts (name, email, phone) VALUES (?, ?, ?)");

preparedStatement.setString(1, name);

preparedStatement.setString(2, email);

preparedStatement.setString(3, phone);

preparedStatement.executeUpdate();

preparedStatement.close();

connection.close();

clearFields();

retrieveContacts();

}

} catch (SQLException e) {

e.printStackTrace();

}

}

private void updateContact() {

int contactID = getContactID();

String name = nameField.getText();

String email = emailField.getText();

String phone = phoneField.getText();

if(contactID<=0){

JOptionPane.showMessageDialog(frame, "Invalid Contact ID", "Error", JOptionPane.ERROR\_MESSAGE);

return;

}else if (name.isEmpty() || email.isEmpty() || phone.isEmpty()) {

JOptionPane.showMessageDialog(frame, "Please enter all fields.", "Error", JOptionPane.ERROR\_MESSAGE);

return;

}else if(email.indexOf("@")==-1 || email.indexOf(".")==-1){

JOptionPane.showMessageDialog(frame, "Invalid Email", "Number Error", JOptionPane.ERROR\_MESSAGE);

return;

}else if(phone.length()!=10){

JOptionPane.showMessageDialog(frame, "Phone Number should have 10 digits", "Number Error", JOptionPane.ERROR\_MESSAGE);

return;

}

try {

Connection connection = DriverManager.getConnection(DB\_URL + DATABASE\_NAME, USER, PASSWORD);

PreparedStatement preparedStatement = connection.prepareStatement(

"UPDATE contacts SET name = ?, email = ?, phone = ? WHERE id = ?");

preparedStatement.setString(1, name);

preparedStatement.setString(2, email);

preparedStatement.setString(3, phone);

preparedStatement.setInt(4, contactID);

int rowsAffected = preparedStatement.executeUpdate();

if (rowsAffected > 0) {

JOptionPane.showMessageDialog(frame, "Contact updated successfully.", "Success", JOptionPane.INFORMATION\_MESSAGE);

clearFields();

retrieveContacts();

} else {

JOptionPane.showMessageDialog(frame, "Failed to update contact. Contact ID not found.", "Error", JOptionPane.ERROR\_MESSAGE);

}

preparedStatement.close();

connection.close();

} catch (SQLException e) {

e.printStackTrace();

}

}

private int getContactID(){

try {

return Integer.parseInt(contactIdField.getText().trim());

} catch (NumberFormatException e) {

return 0;

}

}

private void deleteContact() {

int contactID = getContactID();

if (contactID <= 0) {

JOptionPane.showMessageDialog(frame, "Invalid Contact ID", "Error", JOptionPane.ERROR\_MESSAGE);

return;

}

try {

Connection connection = DriverManager.getConnection(DB\_URL + DATABASE\_NAME, USER, PASSWORD);

PreparedStatement preparedStatement = connection.prepareStatement("DELETE FROM contacts WHERE id = ?");

preparedStatement.setInt(1, contactID);

int rowsAffected = preparedStatement.executeUpdate();

if (rowsAffected > 0) {

JOptionPane.showMessageDialog(frame, "Contact deleted successfully.", "Success", JOptionPane.INFORMATION\_MESSAGE);

clearFields();

retrieveContacts();

} else {

JOptionPane.showMessageDialog(frame, "Failed to delete contact. Contact ID not found.", "Error", JOptionPane.ERROR\_MESSAGE);

}

preparedStatement.close();

connection.close();

} catch (SQLException e) {

e.printStackTrace();

}

}

private void searchContact() {

String keyword = JOptionPane.showInputDialog(frame, "Enter name or email to search:");

if (keyword == null) {

return;

}

try {

Connection connection = DriverManager.getConnection(DB\_URL + DATABASE\_NAME, USER, PASSWORD);

PreparedStatement preparedStatement = connection.prepareStatement(

"SELECT \* FROM contacts WHERE name LIKE ? OR email LIKE ?");

preparedStatement.setString(1, "%" + keyword + "%");

preparedStatement.setString(2, "%" + keyword + "%");

ResultSet resultSet = preparedStatement.executeQuery();

StringBuilder contactsText = new StringBuilder();

while (resultSet.next()) {

int id = resultSet.getInt("id");

String name = resultSet.getString("name");

String email = resultSet.getString("email");

String phone = resultSet.getString("phone");

contactsText.append("ID: " + id + "\t Name: " + name + "\t\t Email: " + email + "\t Phone: " + phone + "\n");

}

if (contactsText.length() > 0) {

displayArea.setText(contactsText.toString());

} else {

displayArea.setText("No matching contacts found.");

}

resultSet.close();

preparedStatement.close();

connection.close();

} catch (SQLException e) {

e.printStackTrace();

}

}

private void retrieveContacts() {

try {

Connection connection = DriverManager.getConnection(DB\_URL + DATABASE\_NAME, USER, PASSWORD);

Statement statement = connection.createStatement();

ResultSet resultSet = statement.executeQuery("SELECT \* FROM contacts");

StringBuilder contactsText = new StringBuilder();

while (resultSet.next()) {

int id = resultSet.getInt("id");

String name = resultSet.getString("name");

String email = resultSet.getString("email");

String phone = resultSet.getString("phone");

contactsText.append("ID: "+id+"\t Name: "+name +"\t\t Email: "+email+"\t Phone: "+phone+"\n"); }

displayArea.setText(contactsText.toString());

resultSet.close();

statement.close();

connection.close();

} catch (SQLException e) {

e.printStackTrace();

}

}

private void clearFields() {

nameField.setText("");

emailField.setText("");

phoneField.setText("");

contactIdField.setText("Not required for Creating a new contact");

}

public static void main(String[] args) {

SwingUtilities.invokeLater(() -> new ese2());

}

}































