import java.sql.\*;

import java.util.Scanner;

public class DatabaseExample {

   private static final String URL = "jdbc:mysql://10.10.13.97/te31340\_db";

   private static final String USER = "te31340";

   private static final String PASSWORD = "te31340";

   public static void main(String[] args) {

       try {

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

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

           System.out.println("Classpath: " + System.getProperty("jmysql -h 10.10.14.184 -u te31340 -p java.class.path"));

           try (Connection connection = DriverManager.getConnection(URL, USER, PASSWORD);

                Scanner scanner = new Scanner(System.in)) {

               System.out.println("Connected to the database.");

               boolean exit = false;

               while (!exit) {

                   System.out.println("\nMenu:");

                   System.out.println("1. Add User");

                   System.out.println("2. Read Users");

                   System.out.println("3. Update User");

                   System.out.println("4. Delete User");

                   System.out.println("5. Exit");

                   System.out.print("Choose an option: ");

                   int choice = scanner.nextInt();

                   scanner.nextLine(); // Consume newline

                   switch (choice) {

                       case 1:

                           System.out.print("Enter name: ");

                           String name = scanner.nextLine();

                           System.out.print("Enter email: ");

                           String email = scanner.nextLine();

                           if (isValidEmail(email)) {

                               addUser(connection, name, email);

                           } else {

                               System.out.println("Invalid email format.");

                           }

                           break;

                       case 2:

                           readUsers(connection);

                           break;

                       case 3:

                           System.out.print("Enter user ID to update: ");

                           int updateId = scanner.nextInt();

                           scanner.nextLine(); // Consume newline

                           System.out.print("Enter new name: ");

                           String newName = scanner.nextLine();

                           System.out.print("Enter new email: ");

                           String newEmail = scanner.nextLine();

                           if (isValidEmail(newEmail)) {

                               updateUser(connection, updateId, newName, newEmail);

                           } else {

                               System.out.println("Invalid email format.");

                           }

                           break;

                       case 4:

                           System.out.print("Enter user ID to delete: ");

                           int deleteId = scanner.nextInt();

                           deleteUser(connection, deleteId);

                           break;

                       case 5:

                           exit = true;

                           System.out.println("Exiting...");

                           break;

                       default:

                           System.out.println("Invalid option. Please try again.");

                   }

               }

           }

       } catch (ClassNotFoundException e) {

           System.err.println("MySQL JDBC Driver not found. Make sure it is in your classpath.");

       } catch (SQLException e) {

           System.err.println("Database error: " + e.getMessage());

       }

   }

   public static void addUser(Connection connection, String name, String email) {

       String sql = "INSERT INTO users (name, email) VALUES (?, ?)";

       try (PreparedStatement statement = connection.prepareStatement(sql)) {

           statement.setString(1, name);

           statement.setString(2, email);

           statement.executeUpdate();

           System.out.println("User added: " + name);

       } catch (SQLException e) {

           System.err.println("Error adding user: " + e.getMessage());

       }

   }

   public static void readUsers(Connection connection) {

       String sql = "SELECT \* FROM users";

       try (PreparedStatement statement = connection.prepareStatement(sql);

            ResultSet resultSet = statement.executeQuery()) {

           while (resultSet.next()) {

               System.out.println("ID: " + resultSet.getInt("id") +

                                  ", Name: " + resultSet.getString("name") +

                                  ", Email: " + resultSet.getString("email"));

           }

       } catch (SQLException e) {

           System.err.println("Error reading users: " + e.getMessage());

       }

   }

   public static void updateUser(Connection connection, int id, String name, String email) {

       String sql = "UPDATE users SET name = ?, email = ? WHERE id = ?";

       try (PreparedStatement statement = connection.prepareStatement(sql)) {

           statement.setString(1, name);

           statement.setString(2, email);

           statement.setInt(3, id);

           int rowsUpdated = statement.executeUpdate();

   if (rowsUpdated > 0) {

               System.out.println("User updated: " + name);

           } else {

               System.out.println("User not found.");

           }

       } catch (SQLException e) {

           System.err.println("Error updating user: " + e.getMessage());

       }

   }

   public static void deleteUser(Connection connection, int id) {

       String sql = "DELETE FROM users WHERE id = ?";

       try (PreparedStatement statement = connection.prepareStatement(sql)) {

           statement.setInt(1, id);

           int rowsDeleted = statement.executeUpdate();

           if (rowsDeleted > 0) {

               System.out.println("User deleted with ID: " + id);

           } else {

               System.out.println("User not found.");

           }

       } catch (SQLException e) {

           System.err.println("Error deleting user: " + e.getMessage());

       }

   }

   private static boolean isValidEmail(String email) {

       String emailRegex = "^[a-zA-Z0-9.\_%+-]+@[a-zA-Z0-9.-]+\\.[a-zA-Z]{2,}$";

       return email.matches(emailRegex);

   }

}