CyberSecurityRiskApp.java

import java.sql.\*;  
import java.util.Scanner;  
  
public class CyberSecurityRiskApp {  
 private static final String DB\_URL = "jdbc:sqlite:risk\_data.db";  
  
 public static void main(String[] args) {  
 createTableIfNotExists();  
 Scanner scanner = new Scanner(System.in);  
 int choice;  
  
 do {  
 System.out.println("\n--- Cybersecurity Risk Assessment ---");  
 System.out.println("1. Add Risk");  
 System.out.println("2. View Risks");  
 System.out.println("3. Exit");  
 System.out.print("Choose an option: ");  
 choice = scanner.nextInt(); scanner.nextLine();  
  
 switch (choice) {  
 case 1 -> addRisk(scanner);  
 case 2 -> viewRisks();  
 case 3 -> System.out.println("Exiting...");  
 default -> System.out.println("Invalid choice.");  
 }  
 } while (choice != 3);  
 }  
  
 private static void createTableIfNotExists() {  
 try (Connection conn = DriverManager.getConnection(DB\_URL);  
 Statement stmt = conn.createStatement()) {  
 String sql = "CREATE TABLE IF NOT EXISTS risks (" +  
 "id INTEGER PRIMARY KEY AUTOINCREMENT," +  
 "description TEXT NOT NULL," +  
 "severity INTEGER," +  
 "likelihood INTEGER," +  
 "mitigation TEXT);";  
 stmt.execute(sql);  
 } catch (SQLException e) {  
 e.printStackTrace();  
 }  
 }  
  
 private static void addRisk(Scanner scanner) {  
 System.out.print("Enter risk description: ");  
 String description = scanner.nextLine();  
  
 System.out.print("Enter severity (1-10): ");  
 int severity = scanner.nextInt();  
  
 System.out.print("Enter likelihood (1-10): ");  
 int likelihood = scanner.nextInt(); scanner.nextLine();  
  
 System.out.print("Enter mitigation plan: ");  
 String mitigation = scanner.nextLine();  
  
 try (Connection conn = DriverManager.getConnection(DB\_URL);  
 PreparedStatement pstmt = conn.prepareStatement(  
 "INSERT INTO risks(description, severity, likelihood, mitigation) VALUES (?, ?, ?, ?)")) {  
  
 pstmt.setString(1, description);  
 pstmt.setInt(2, severity);  
 pstmt.setInt(3, likelihood);  
 pstmt.setString(4, mitigation);  
 pstmt.executeUpdate();  
  
 System.out.println("Risk added successfully.");  
 } catch (SQLException e) {  
 e.printStackTrace();  
 }  
 }  
  
 private static void viewRisks() {  
 try (Connection conn = DriverManager.getConnection(DB\_URL);  
 Statement stmt = conn.createStatement();  
 ResultSet rs = stmt.executeQuery("SELECT \* FROM risks")) {  
  
 System.out.println("\n--- Risk Records ---");  
 while (rs.next()) {  
 int id = rs.getInt("id");  
 String desc = rs.getString("description");  
 int sev = rs.getInt("severity");  
 int like = rs.getInt("likelihood");  
 String mitig = rs.getString("mitigation");  
 String level = getRiskLevel(sev, like);  
  
 System.out.printf("ID: %d\nDescription: %s\nSeverity: %d\nLikelihood: %d\nRisk Level: %s\nMitigation: %s\n\n",  
 id, desc, sev, like, level, mitig);  
 }  
  
 } catch (SQLException e) {  
 e.printStackTrace();  
 }  
 }  
  
 private static String getRiskLevel(int severity, int likelihood) {  
 int riskScore = severity \* likelihood;  
 if (riskScore >= 70) return "HIGH";  
 else if (riskScore >= 40) return "MEDIUM";  
 else return "LOW";  
 }  
}