// Name: Leen Sobhia  
// Student ID:  
  
//References: Please note that I used the following for reference code: https://www.postgresqltutorial.com/postgresql-jdbc/  
  
import java.sql.\*;  
import java.util.Scanner;  
  
public class Assignment3 {  
 //Function name: getConnection  
 //Input: none  
 //Output: a return of type Connection  
 //Purpose: to establish and check a connection to the POSTGRESQL database. Return a connection type data if successful, null otherwise.  
 public static Connection getConnection(){  
 String url = "jdbc:postgresql://localhost:5432/Assignment3";  
 String user = "postgres";  
 String password = "Sobh0011!";  
  
 try{  
 Class.*forName*("org.postgresql.Driver");  
 Connection connection = DriverManager.*getConnection*(url, user, password);  
 return connection;  
 }  
 catch(Exception e) {  
 System.*out*.println("something");  
 //e.printStackTrace();  
 }  
  
 return null;  
 }  
 //Function name: getAllStudents()  
 //Input: none  
 //Output: void, display data from students table.  
 //Purpose: the purpose of this function is to retrieve and display all records from the students table in the POSTGRESQL database.  
 public static void getAllStudents() {  
 System.*out*.println("Returning all student records: ");  
 Connection newConnect = *getConnection*();  
  
 if (newConnect != null) {  
 try (Statement statement = newConnect.createStatement()) {  
 ResultSet resultSet = statement.executeQuery("SELECT \* FROM students");  
 while (resultSet.next()) {  
 System.*out*.print(resultSet.getInt("student\_id") + " \t");  
 System.*out*.print(resultSet.getString("first\_name") + " \t");  
 System.*out*.print(resultSet.getString("last\_name") + " \t");  
 System.*out*.print(resultSet.getString("email") + " \t");  
 System.*out*.println(resultSet.getDate("enrollment\_date") + " \t");  
 }  
 } catch (Exception e) {  
 System.*out*.println("Error: " + e.getMessage());  
 e.printStackTrace();  
 }  
 }  
 }  
  
 //Function name: addStudent()  
 //Input: first name, last name, e-mail, and date of student enrollment  
 //Output: void  
 //Purpose: the purpose of this function is to insert a new student record into the students table in the POSTGRESQL database, using the input values.  
 public static void addStudent(String fname, String lname, String e\_mail, Date date\_enrolled) {  
 Connection newConnect = *getConnection*();  
  
 if (newConnect != null) {  
 String resultSet = "INSERT INTO students (first\_name, last\_name, email, enrollment\_date) VALUES (?, ?, ?, ?)";  
  
 try (PreparedStatement statement = newConnect.prepareStatement(resultSet)) {  
 statement.setString(1, fname);  
 statement.setString(2, lname);  
 statement.setString(3, e\_mail);  
 statement.setDate(4, date\_enrolled);  
  
 int rowsAffected = statement.executeUpdate();  
  
 if (rowsAffected > 0) {  
 System.*out*.println("Student " + fname + " " + lname + " added successfully");  
 } else {  
 System.*out*.println("Failed to add student");  
 }  
  
 } catch (SQLException e) {  
 System.*out*.println("Error: " + e.getMessage());  
 e.printStackTrace();  
 }  
 }  
  
 }  
  
 //Function name: updateStudentEmail()  
 //Input: student id and new email  
 //Output: void  
 //Purpose: the purpose of this function is to update the email address for a student with the specified student id.  
 public static void updateStudentEmail(int stu\_id, String new\_email){  
 Connection newConnect = *getConnection*();  
  
 if (newConnect != null) {  
 String resultSet = "UPDATE students SET email = ? WHERE student\_id = ?";  
  
 try (PreparedStatement statement = newConnect.prepareStatement(resultSet)) {  
 statement.setString(1, new\_email);  
 statement.setInt(2, stu\_id);  
  
 int rowsAffected = statement.executeUpdate();  
  
 if (rowsAffected > 0) {  
 System.*out*.println("Email updated for student with ID: " + stu\_id);  
 } else {  
 System.*out*.println("No student found with ID: " + stu\_id);  
 }  
  
 } catch (SQLException e) {  
 System.*out*.println("Error: " + e.getMessage());  
 e.printStackTrace();  
 }  
 }  
 }  
  
 //Function name: deleteStudent()  
 //Input: student id  
 //Output: void  
 //Purpose: the purpose of this function is to delete a student with a specific id from the students table in POSTGRESQL database.  
 public static void deleteStudent(int stu\_id){  
  
 Connection newConnect = *getConnection*();  
  
 if (newConnect != null) {  
 String resultSet = "DELETE FROM students WHERE student\_id = ?";  
  
 try (PreparedStatement statement = newConnect.prepareStatement(resultSet)) {  
 statement.setInt(1, stu\_id);  
 int rowsAffected = statement.executeUpdate();  
  
 if (rowsAffected > 0) {  
 System.*out*.println("Successfully deleted student with ID: " + stu\_id);  
 } else {  
 System.*out*.println("No student found with ID: " + stu\_id);  
 }  
  
 } catch (SQLException e) {  
 System.*out*.println("Error: " + e.getMessage());  
 e.printStackTrace();  
 }  
 }  
 }  
  
 //Function name: main()  
 //Purpose: test the functionality of the class functions based on user test inputs.  
 public static void main (String[] args){  
 Scanner scanner = new Scanner(System.*in*);  
  
 while (true) {  
 System.*out*.println(" ");  
 System.*out*.println("Choose an option:");  
 System.*out*.println("1. Get all students");  
 System.*out*.println("2. Add a student");  
 System.*out*.println("3. Update student email");  
 System.*out*.println("4. Delete a student");  
 System.*out*.println("5. Exit");  
 System.*out*.println(" ");  
 System.*out*.print("Enter selection: ");  
  
 if (scanner.hasNextInt()) {  
 int choice = scanner.nextInt();  
  
 switch (choice) {  
 case 1:  
 *getAllStudents*();  
 break;  
 case 2:  
 scanner.nextLine(); // Consume newline  
 System.*out*.println("Enter first name:");  
 String firstName = scanner.nextLine();  
 System.*out*.println("Enter last name:");  
 String lastName = scanner.nextLine();  
 System.*out*.println("Enter email:");  
 String email = scanner.nextLine();  
 System.*out*.println("Enter enrollment date (yyyy-mm-dd):");  
 String enrollmentDateStr = scanner.nextLine();  
 Date enrollmentDate = Date.*valueOf*(enrollmentDateStr);  
 *addStudent*(firstName, lastName, email, enrollmentDate);  
 break;  
 case 3:  
 System.*out*.println("Enter student id:");  
 if (scanner.hasNextInt()) {  
 int studentId = scanner.nextInt();  
 scanner.nextLine(); // Consume newline  
 System.*out*.println("Enter new email:");  
 String newEmail = scanner.nextLine();  
 *updateStudentEmail*(studentId, newEmail);  
 } else {  
 System.*out*.println("Invalid input. Please enter a valid student id.");  
 scanner.next(); // Clear invalid input  
 }  
 break;  
 case 4:  
 System.*out*.println("Enter student id to delete:");  
 if (scanner.hasNextInt()) {  
 int studentIdToDelete = scanner.nextInt();  
 scanner.nextLine(); // Consume newline  
 *deleteStudent*(studentIdToDelete);  
 } else {  
 System.*out*.println("Invalid input. Please enter a valid student id.");  
 scanner.next(); // Clear invalid input  
 }  
 break;  
 case 5:  
 System.*out*.println("Exiting...");  
 System.*exit*(0);  
 break;  
 default:  
 System.*out*.println("Invalid choice. Please try again.");  
 }  
 }else {  
 System.*out*.println("Invalid input. Please enter a valid option.");  
 scanner.next(); // Clear invalid input  
 }  
 }  
 }  
}