Project Report Draft: Quiz Database Manager Application

Project Title: Quiz Database Manager Desktop Application
Name: Kesar .A. Duseja
Roll No.: 10
Course: Database Management Systems (DBMS) Mini Project
Semester: 3
Guide: Kajal Ma'am
Date: October 15, 2025

1. **Abstract (≤ 150 words)**

This project implemented a **Quiz Database Manager**, a Java Swing desktop application, to provide a graphical user interface (GUI) for viewing and managing the core entities of a quiz system database. The application connects to a **Microsoft SQL Server** instance and allows authorized users to perform fundamental CRUD (Create, Read, Update, Delete) operations on tables like `Student`, `Admin`, and `Quiz`. A key feature is the inclusion of specialized **reporting functionalities**, such as calculating a student's performance, summarizing quiz statistics, and tracking administrative quiz creation. The goal was to demonstrate proficiency in integrating relational database concepts with a front-end application using **JDBC** and structured query language (SQL).

2. **Prerequisites**

Requirement	Description
Hardware/OS	A standard desktop or laptop running **Windows 10/11**, **macOS**, or **Linux**.
Permissions	Administrator or root access for installing software (JDK, SQL Server).
Accounts	A login account (`DB_USER`) with **read, write, and execute permissions** on the `quizdb` database in SQL Server.
Connectivity	Local network access to the SQL Server instance (default port 1433).

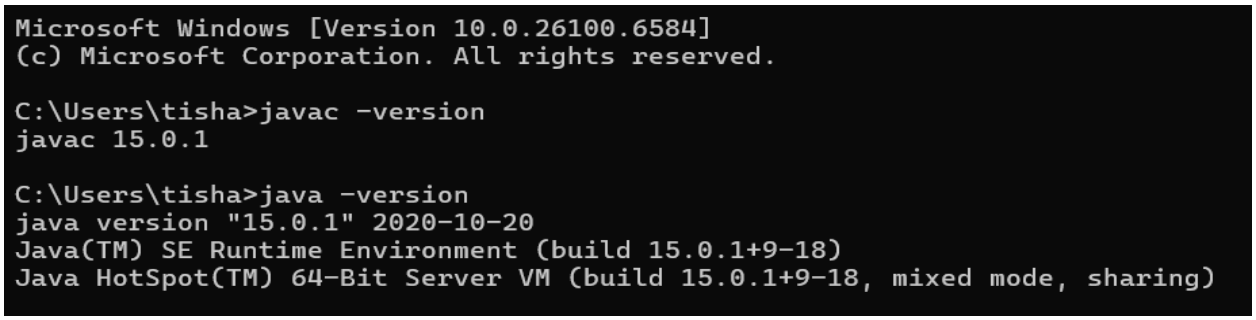
3. **Tools & Versions (exact)**

Tool/Technology	Exact Version	Connector/Driver File

Operating System	Windows 10 Pro 22H2	N/A
Database Server	Microsoft SQL Server [Specify Version, e.g., 2019/2022 Express]	N/A
Java Development Kit (JDK)	**Java JDK 17.0.8** (or a compatible LTS version)	N/A
JDBC Driver	Microsoft JDBC Driver for SQL Server [Specify Version, e.g., 12.6.1]	mssql-jdbc-[version].jar
Development IDE	IntelliJ IDEA Community Edition 2024.1	N/A

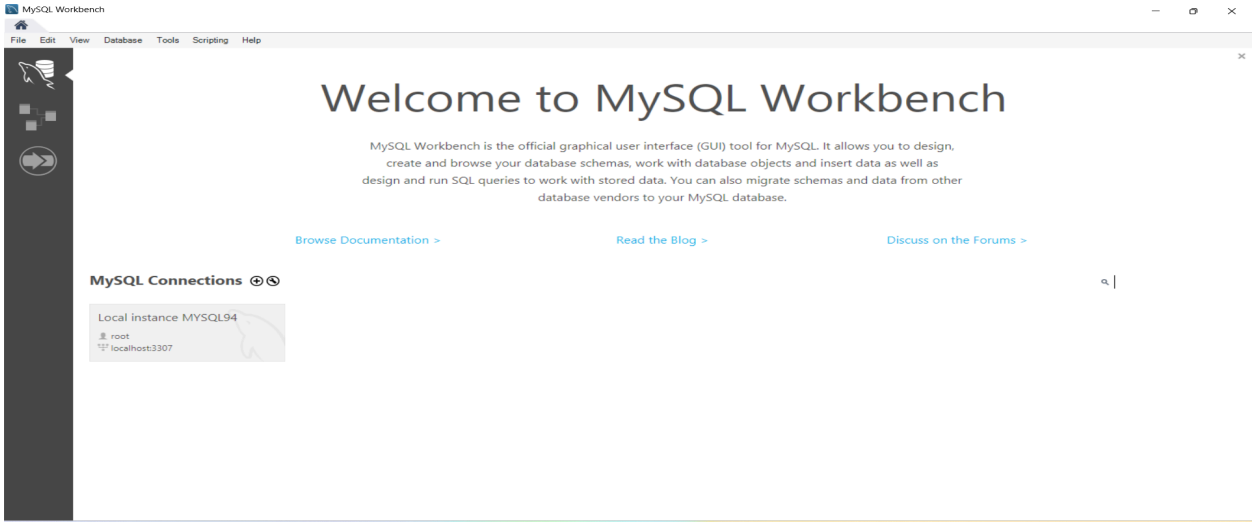
4. Environment Setup — Step by Step

Step 1: Install Java Development Kit (JDK)



Step 2: Install MySQL Server and Workbench

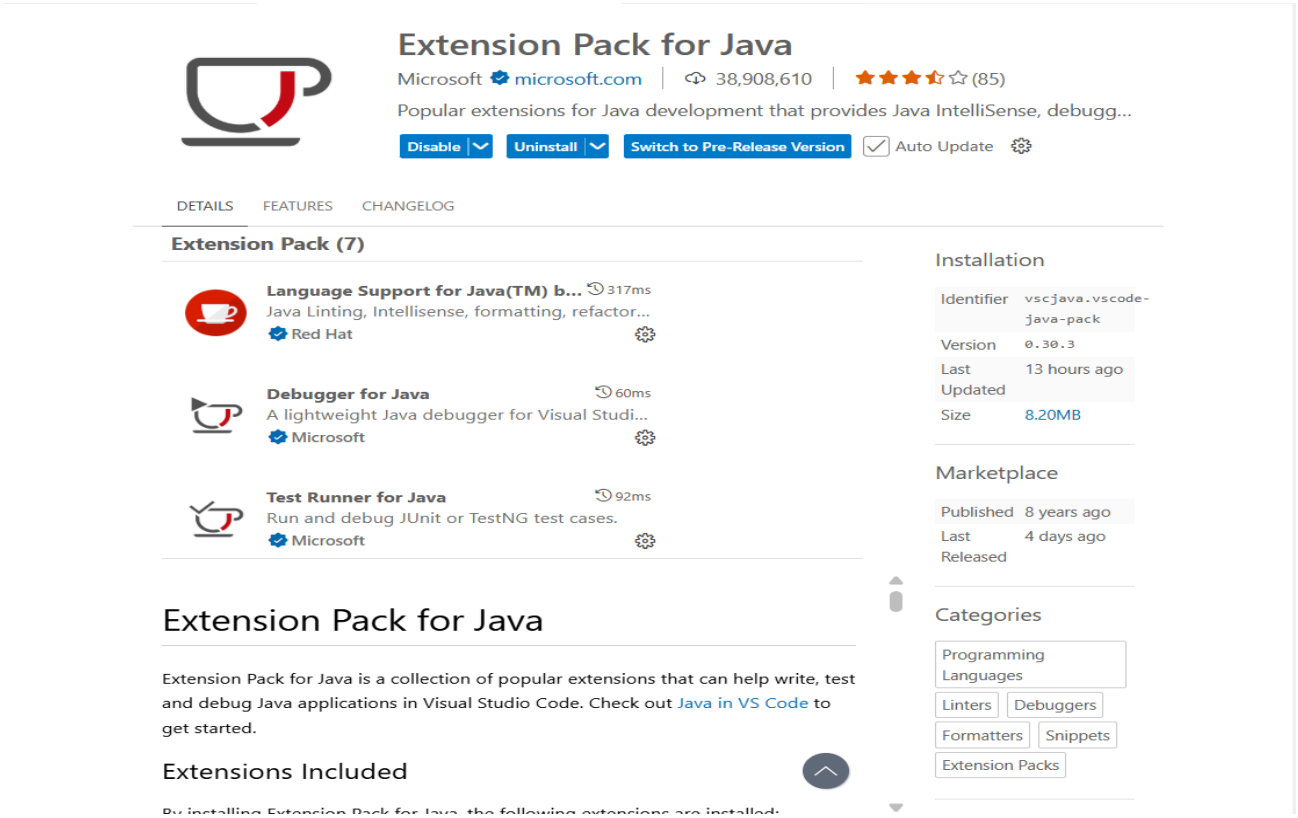
Download and install MySQL Server and MySQL Workbench. During setup, create a root password (e.g., root) and note it down for later use.



<u>Step 3</u>: Install Visual Studio Code

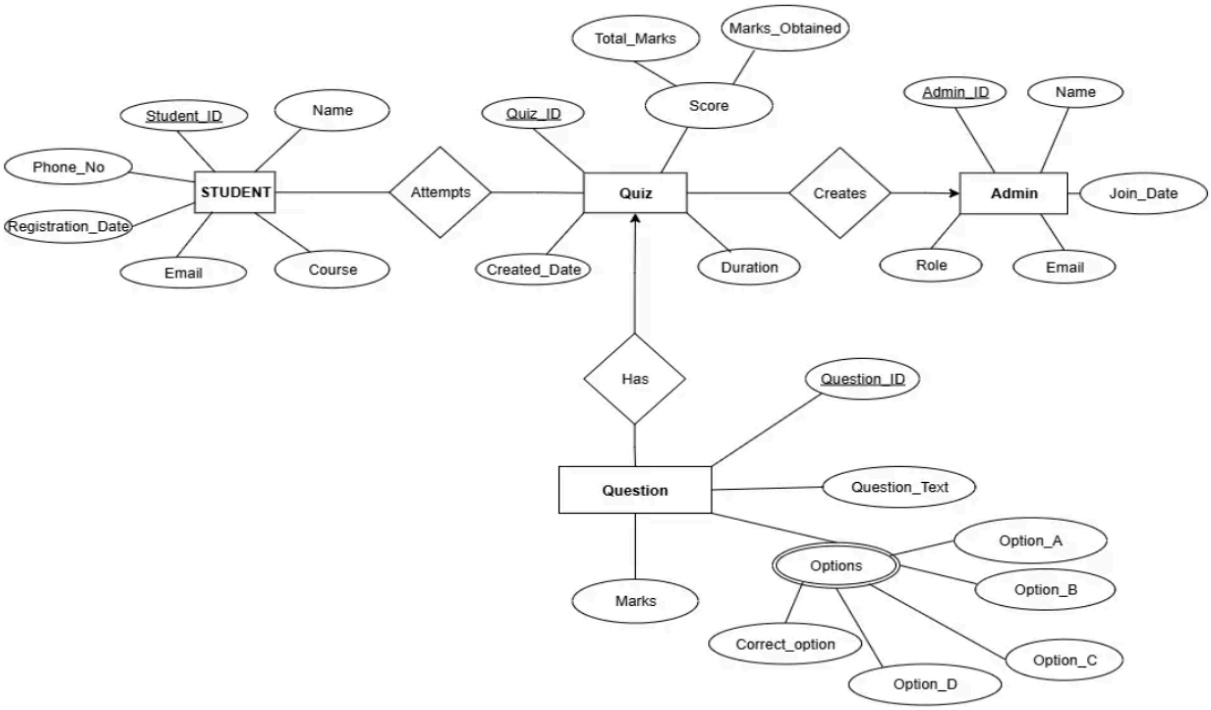
Download and install VS Code from https://code.visualstudio.com.

Install the “Extension Pack for Java” from the Extensions tab.



5. Database Design

- ER diagram



- Schema Definition

  Database Name: quizdb

  Table Name: student

```
1  Create database quizdb;
2  Use quizdb;
3  CREATE TABLE Student (
4      Student_ID INT PRIMARY KEY,
5      Name VARCHAR(100) NOT NULL,
6      Phone_No VARCHAR(15) UNIQUE,
7      Registration_Date DATE NOT NULL,
8      Email VARCHAR(100) UNIQUE NOT NULL,
9      Course VARCHAR(50)
10 );
11
```

- Insert Sample Data

```sql
INSERT INTO Student (Student_ID, Name, Phone_No, Registration_Date, Email, Course)
VALUES
(1, 'Aarav Sharma', '9876543210', '2023-01-10', 'aarav1@example.com', 'Computer Science'),
(2, 'Riya Patel', '9876543211', '2023-01-11', 'riya2@example.com', 'Information Technology'),
(3, 'Arjun Singh', '9876543212', '2023-01-12', 'arjun3@example.com', 'Electronics'),
(4, 'Priya Mehta', '9876543213', '2023-01-13', 'priya4@example.com', 'Mechanical'),
(5, 'Kunal Joshi', '9876543214', '2023-01-14', 'kunal5@example.com', 'Computer Science')
```

6. Implementation (Step-by-Step)

STEP 1:

```java
// Step 4: Create a Statement to run queries
stmt = con.createStatement();

// Step 5: Execute a query - change 'users' if your table name differs
String query = "SELECT * FROM student";
rs = stmt.executeQuery(query);

// Step 6: Display the results
System.out.println(x:"\n--- Data from 'student' table: ---");
System.out.println(x:"--------------------------------");
```

```
--- Data from 'student' table: ---
--------------------------------
Student_ID: 1 Name: S1 Phone_No: 9000000001 Registration_Date: 2023-01-01 Email: s1@example.com Cours
e: CS
Student_ID: 2 Name: S2 Phone_No: 9000000002 Registration_Date: 2023-01-02 Email: s2@example.com Cours
e: IT
Student_ID: 3 Name: Arjun Singh Phone_No: 9876543212 Registration_Date: 2023-01-12 Email: arjun3@exam
ple.com Course: Electronics
Student_ID: 4 Name: Honey  Phone_No: 8884519005 Registration_Date: 2024-02-10 Email: honey6@example.c
om Course: Mechanical
Student_ID: 5 Name: Kunal Joshi Phone_No: 9876543214 Registration_Date: 2023-01-14 Email: kunal5@exam
ple.com Course: Computer Science
```

STEP 2:

Establish Connection

```
// Step 3: Establish Connection
con = DriverManager.getConnection(url, user, password);
System.out.println(x:"\n✅ Connected to quizdb successfully!");
```

STEP 3:

Calculate and display the results for a specific student across all their quiz attempts. This uses a `JOIN` operation across `Attempts`, `Student`, and `Quiz` tables.

```
private void generateStudentPerformanceReport() {
    String studentIdStr = JOptionPane.showInputDialog(this, message:"Enter Student ID:", tit
    if (studentIdStr == null || studentIdStr.trim().isEmpty()) return;

    try {
        int studentId = Integer.parseInt(studentIdStr);
        String sql = "SELECT s.Name, a.Quiz_ID, a.Marks_Obtained, q.Total_Marks " +
                     "FROM Attempts a " +
                     "JOIN Student s ON a.Student_ID = s.Student_ID " +
                     "JOIN Quiz q ON a.Quiz_ID = q.Quiz_ID " +
                     "WHERE a.Student_ID = ?";
```

STEP 4:

Provide aggregate statistics (Average, Min, Max score, and total attempts) for a specific quiz. This uses SQL aggregate functions and `GROUP BY`.

```
private void generateQuizSummaryReport() {
    String quizIdStr = JOptionPane.showInputDialog(this, message:"Enter Quiz ID:", tit
    if (quizIdStr == null || quizIdStr.trim().isEmpty()) return;

    try {
        int quizId = Integer.parseInt(quizIdStr);
        String sql = "SELECT Quiz_ID, " +
                     "AVG(Marks_Obtained) AS Average_Score, " +
                     "MIN(Marks_Obtained) AS Min_Score, " +
                     "MAX(Marks_Obtained) AS Max_Score, " +
                     "COUNT(Student_ID) AS Number_of_Attempts " +
                     "FROM Attempts " +
                     "WHERE Quiz_ID = ? " +
                     "GROUP BY Quiz_ID";
```

STEP 5:

Identify all students whose marks in any quiz exceed a user-defined percentage of the total marks. This involves a calculated column and comparison in the WHERE clause.

```
private void generateHighScorersReport() {
    String percentageStr = JOptionPane.showInputDialog(this, message:"Show students with scc
    if (percentageStr == null || percentageStr.trim().isEmpty()) return;

    try {
        double percentage = Double.parseDouble(percentageStr);
        String sql = "SELECT s.Name, s.Email, a.Quiz_ID, a.Marks_Obtained, q.Total_Marks " +
                     "FROM Attempts a " +
                     "JOIN Student s ON a.Student_ID = s.Student_ID " +
                     "JOIN Quiz q ON a.Quiz_ID = q.Quiz_ID " +
                     "WHERE (CAST(a.Marks_Obtained AS FLOAT) / q.Total_Marks) * 100 > ?";
```

STEP 6:

Establish a connection to the SQL Server database and load data from a selected table into the JTable component.

```
private void loadTableData() {
    String selectedTable = (String) tableSelector.getSelectedItem();
    if (selectedTable == null) return;

    try (Connection conn = DriverManager.getConnection(DB_URL, DB_USER, DB_PASSWORD);
         Statement stmt = conn.createStatement();
         ResultSet rs = stmt.executeQuery("SELECT * FROM " + selectedTable)) {

        dataTable.setModel(buildTableModel(rs));
```

7. Testing & Results

Test Case 1: Database Connectivity and Data Fetch

Input : Select 'Admin' from JComboBox and click 'Load Data'.

Output:

Quiz Database Manager

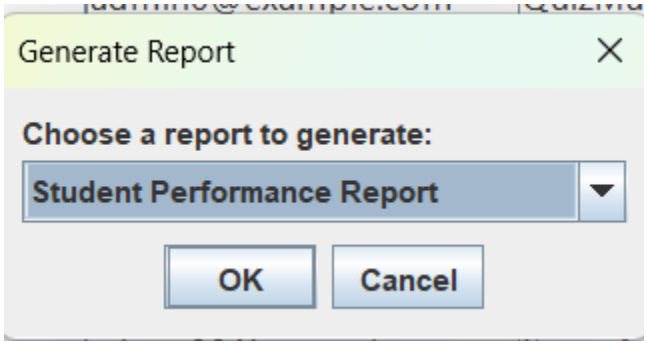
Select Table: Admin Load Data

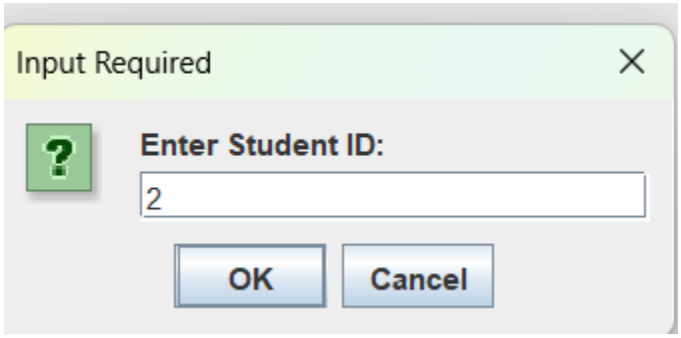
Admin_ID	Name	Email	Role	Join_Date
1	Anil Sharma	admin1@example.com	SuperAdmin	2022-01-05
2	Sunita Mehta	admin2@example.com	Moderator	2022-01-06
3	Rajeev Gupta	admin3@example.com	QuizMaster	2022-01-07
4	Priya Nair	admin4@example.com	Moderator	2022-01-08
5	Suresh Iyer	admin5@example.com	SuperAdmin	2022-01-09
6	Meena Reddy	admin6@example.com	QuizMaster	2022-01-10
7	Kiran Malhotra	admin7@example.com	Moderator	2022-01-11
8	Vikas Jain	admin8@example.com	SuperAdmin	2022-01-12

Test Case 2: Student Performance Report Logic

Input : Student ID=2

Output:



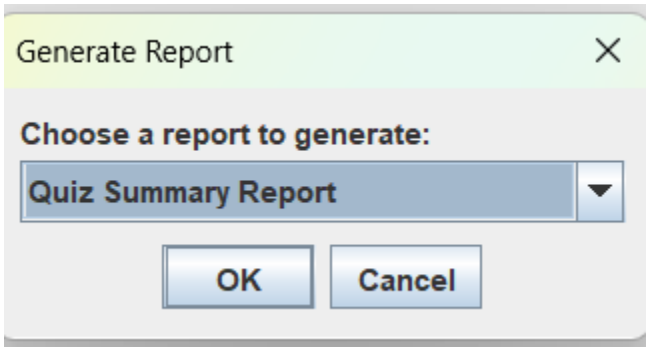


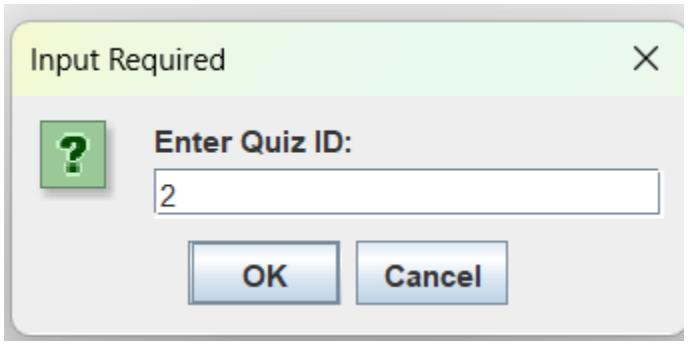
Name	Quiz_ID	Marks_Obtained	Total_Marks
S2	2	78	100
S2	2	78	100

Test Case 3: Aggregate Function Testing (generateQuizSummaryReport())

Input : Quiz ID= 2

Output:

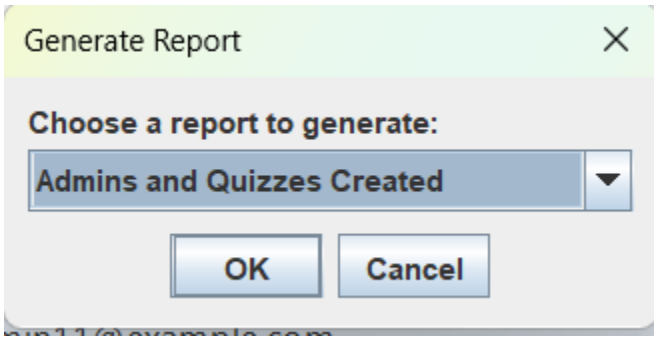




Quiz_ID	Average_Score	Min_Score	Max_Score	Number_of_Attempts
2	78.0000	78	78	2

Test Case 4: Admins and Quizzes created(Generate Report)

Input :



Output:

Name	Email	Quizzes_Created
Anil Sharma	admin1@example.com	1
Sunita Mehta	admin2@example.com	1
Rajeev Gupta	admin3@example.com	1
Priya Nair	admin4@example.com	1
Suresh Iyer	admin5@example.com	1
Meena Reddy	admin6@example.com	1

8. Troubleshooting / Common Errors

Error Message	Cause	Resolution
com.microsoft.sqlserver.jdbc.SQLServerException: The driver could not establish a secure connection...	SQL Server instance may not have a valid SSL certificate, or the connection string is missing `encrypt=true;trustServerCertificate=true;`.	Ensure the connection string includes `trustServerCertificate=true;` if the server is self-signed or unverified.
java.sql.SQLException: Login failed for user 'your_username'.	Incorrect `DB_USER` or `DB_PASSWORD` defined in the Java code.	Double-check the credentials in `QuizDBViewer.java` (lines 33-34) and verify the user exists in SQL Server.
java.lang.ClassNotFoundException: com.microsoft.sqlserver.jdbc.SQLServerDriver	The JDBC driver JAR file is not in the Java classpath during compilation or execution.	Re-run the `javac` and `java` commands, explicitly including the `mssql-jdbc-[version].jar` in the `-cp` (classpath) argument.
java.lang.NumberFormatException: For input string: "abc"	The user entered non-numeric text (e.g., "abc") when prompted for a numeric ID (e.g., Student ID).	The user must enter a valid integer or decimal. The error message is handled gracefully in the report generation methods.

9.Project Demo Instructions

The reviewer can execute the following steps to perform an end-to-end demonstration of the application's core functionality.

Prerequisite: Ensure the SQL Server is running, the database is populated, and the application is compiled with the correct credentials.

Step	Action	Outcome
1. Launch Application	Execute the run command: `java -cp ".;lib/mssql-jdbc-[version].jar" QuizDBViewer`	The main `Quiz Database Manager` window appears.
2. Test Basic Table View	Select **'Quiz'** from the dropdown. Click **'Load Data'**.	The table populates with quiz details.
3. Test Student Report	Click **'Generate Report'**. Select **'Student Performance Report'**. Enter a **Student ID**	Report for the specified student's attempts is displayed.
4. Test Summary Report	Click **'Generate Report'**. Select **'Quiz Summary Report'**. Enter a **Quiz ID**	A single row of aggregate data (AVG, MIN, MAX scores, count) for Quiz 1 is displayed.
5. Test High Scorers Report	Click **'Generate Report'**. Select **'High Scorers Report'**. Enter `90`	Students who scored above 90% in any attempt are listed.
6. Test Admin Report	Click **'Generate Report'**. Select **'Admins and Quizzes Created'**.	The table shows each admin and a count of how many quizzes they created.

10. Appendices

Appendix A: Full SQL DDL Statements

The following SQL Data Definition Language (DDL) creates the necessary tables for the `quizdb` database.

```sql
CREATE TABLE Student (
    Student_ID INT PRIMARY KEY,            -- Unique student identifier
    Name VARCHAR(100) NOT NULL,            -- Student name
    Phone_No VARCHAR(15) UNIQUE,           -- Each phone number must be unique
    Registration_Date DATE NOT NULL,       -- When the student registered
    Email VARCHAR(100) UNIQUE NOT NULL,    -- Unique email
    Course VARCHAR(50)                     -- Course enrolled
);
CREATE TABLE Admin (
    Admin_ID INT PRIMARY KEY,           -- Unique admin identifier
    Name VARCHAR(100) NOT NULL,         -- Admin name
    Email VARCHAR(100) UNIQUE NOT NULL, -- Unique email
    Role VARCHAR(50),                   -- Role of the admin
    Join_Date DATE NOT NULL             -- When admin joined
);
CREATE TABLE Quiz (
    Quiz_ID INT PRIMARY KEY,                -- Unique quiz identifier
    Created_Date DATE NOT NULL,             -- Date when quiz was created
    Duration INT CHECK (Duration > 0),      -- Duration must be positive
    Total_Marks INT CHECK (Total_Marks > 0), -- Quiz must have marks > 0
    Admin_ID INT,                           -- Quiz created by admin
    FOREIGN KEY (Admin_ID) REFERENCES Admin(Admin_ID)
);
CREATE TABLE Attempts (
    Attempt_ID INT PRIMARY KEY,   -- Auto-incrementing attempt ID
    Student_ID INT,                           -- References the student who attempted
    Quiz_ID INT,                              -- References the quiz attempted
    Marks_Obtained INT CHECK (Marks_Obtained >= 0), -- Marks must be non-negative

    FOREIGN KEY (Student_ID) REFERENCES Student(Student_ID) ON DELETE CASCADE,
    FOREIGN KEY (Quiz_ID) REFERENCES Quiz(Quiz_ID) ON DELETE CASCADE
);
CREATE TABLE Question (
    Question_ID INT PRIMARY KEY,          -- Unique ID for each question
    Question_Text TEXT NOT NULL,          -- The question text
    Marks INT CHECK (Marks > 0),          -- Each question must have marks > 0
    Quiz_ID INT,                          -- Belongs to a quiz
    FOREIGN KEY (Quiz_ID) REFERENCES Quiz(Quiz_ID) ON DELETE CASCADE
);
CREATE TABLE Options (

Option_ID INT PRIMARY KEY ,
Question_ID INT,
Option_A VARCHAR(255),
Option_B VARCHAR(255),
Option_C VARCHAR(255),
Option_D VARCHAR(255),
Correct_Option CHAR(1) NOT NULL
CHECK (Correct_Option IN ('A','B','C','D')),
FOREIGN KEY (Question_ID) REFERENCES
Question(Question_ID) ON DELETE CASCADE
);
```

Appendix B: Full Source Code

The full source code for `QuizDBViewer.java` is included in the main report and is also available in the submitted ZIP archive at `code/QuizDBViewer.java`.

Appendix C: README.md

Quiz Database Manager - DBMS Mini Project

Project Overview

This is a Java Swing application designed to connect to a Microsoft SQL Server database (`your_quiz_db`) and manage its core entities. It provides CRUD functionality and advanced reporting features using JDBC.

Prerequisites

1. Java JDK 17.0.8 or later.

2. Microsoft SQL Server [Specify Version] instance running on `localhost:1433`.

3. Microsoft JDBC Driver for SQL Server (JAR file).

Setup Instructions

1. Database Setup

1. Execute the DDL statements from `data/sample_data.sql` in your SQL Server instance to create the schema and populate the tables.

2. Ensure you have a valid SQL Server login user (`DB_USER`, `DB_PASSWORD`) with access to the database.

2. Application Setup

1. Place the `mssql-jdbc-[version].jar` file in a `lib` directory inside the project root.

2. **Crucially, edit `code/QuizDBViewer.java`** and replace the placeholders for `DB_URL`, `DB_USER`, and `DB_PASSWORD` with your actual credentials.

3. Compile and Run

Open your terminal in the project root directory.

```bash

# Assuming the JDBC driver is in lib/
```

1. Compile

javac -cp ".;lib/mssql-jdbc-[version].jar" code/QuizDBViewer.java

2. Run

java -cp ".;lib/mssql-jdbc-[version].jar:code" QuizDBViewer

Features

- **Load Data:** View data from any of the configured tables (`Student`, `Admin`, `Quiz`, etc.).
- **CRUD Operations:** Buttons are provided for **A**dd, **U**pdate, and **D**elete (functionality is stubbed in the provided code).
- **Reports:**
    - Student Performance Report (by Student ID)
    - Quiz Summary Report (Aggregate scores by Quiz ID)
    - High Scorers Report (Students above a custom percentage)
    - Admin Quiz Count (Number of quizzes created by each administrator)

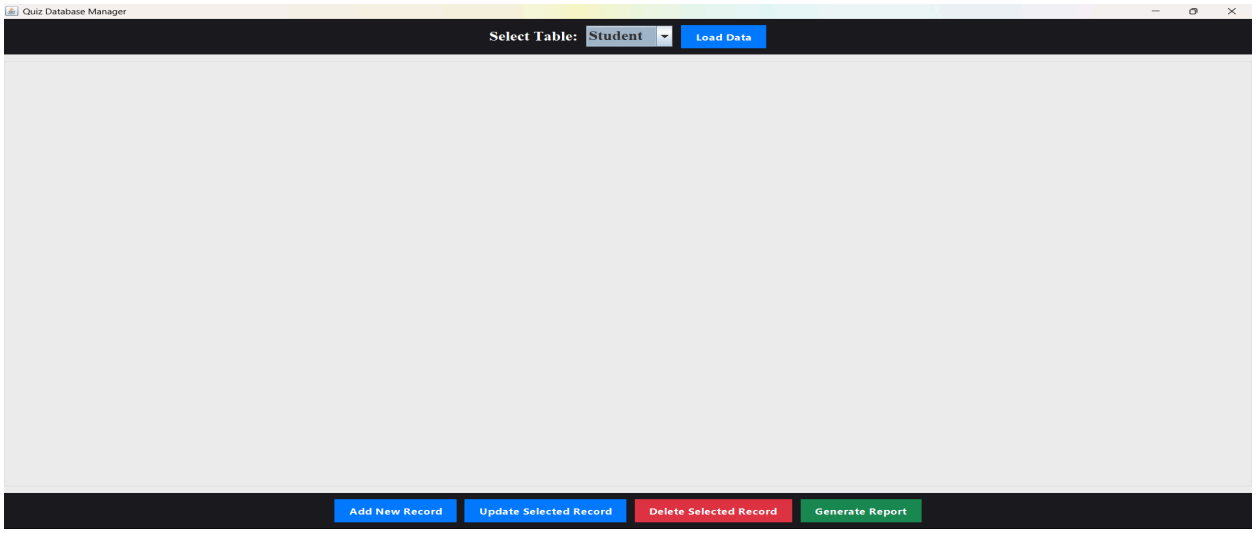
Appendix D: References / Libraries Used

1. **Java Swing:** Used for the desktop GUI components.

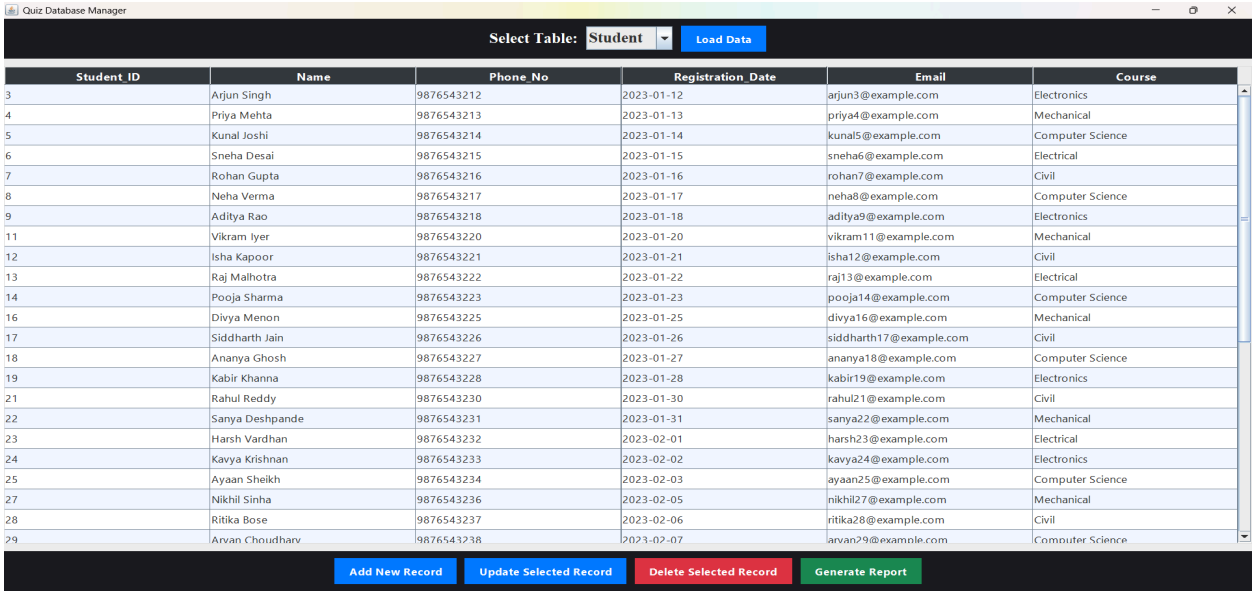
2. **Java Database Connectivity (JDBC):** Standard Java API for database interaction.

3. **Microsoft JDBC Driver for SQL Server:** Specific driver for connecting Java to SQL Server.

- **<u>Login Page:</u>**

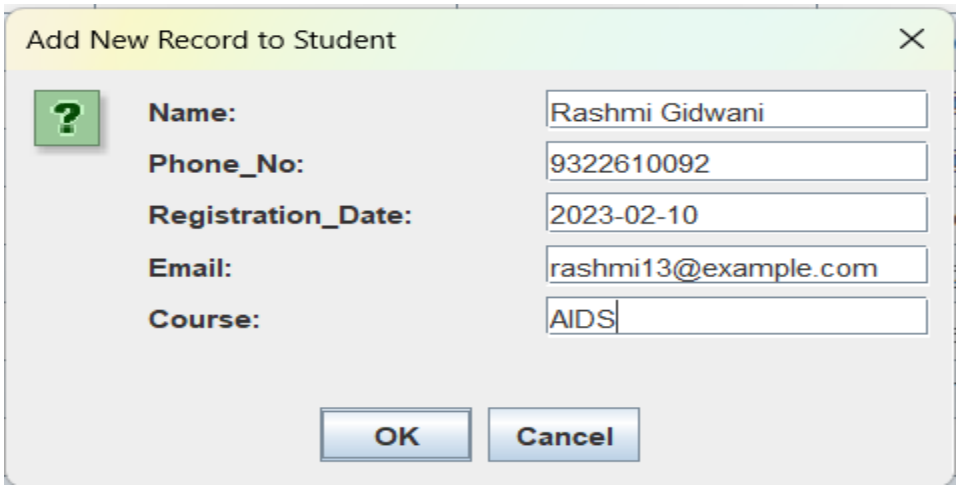


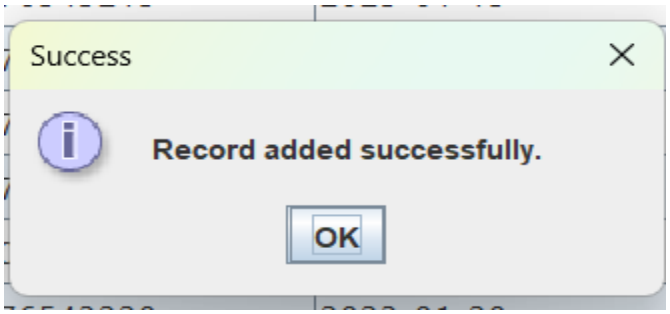
- **Load Data**



Student_ID	Name	Phone_No	Registration_Date	Email	Course
3	Arjun Singh	9876543212	2023-01-12	arjun3@example.com	Electronics
4	Priya Mehta	9876543213	2023-01-13	priya4@example.com	Mechanical
5	Kunal Joshi	9876543214	2023-01-14	kunal5@example.com	Computer Science
6	Sneha Desai	9876543215	2023-01-15	sneha6@example.com	Electrical
7	Rohan Gupta	9876543216	2023-01-16	rohan7@example.com	Civil
8	Neha Verma	9876543217	2023-01-17	neha8@example.com	Computer Science
9	Aditya Rao	9876543218	2023-01-18	aditya9@example.com	Electronics
11	Vikram Iyer	9876543220	2023-01-20	vikram11@example.com	Mechanical
12	Isha Kapoor	9876543221	2023-01-21	isha12@example.com	Civil
13	Raj Malhotra	9876543222	2023-01-22	raj13@example.com	Electrical
14	Pooja Sharma	9876543223	2023-01-23	pooja14@example.com	Computer Science
16	Divya Menon	9876543225	2023-01-25	divya16@example.com	Mechanical
17	Siddharth Jain	9876543226	2023-01-26	siddharth17@example.com	Civil
18	Ananya Ghosh	9876543227	2023-01-27	ananya18@example.com	Computer Science
19	Kabir Khanna	9876543228	2023-01-28	kabir19@example.com	Electronics
21	Rahul Reddy	9876543230	2023-01-30	rahul21@example.com	Civil
22	Sanya Deshpande	9876543231	2023-01-31	sanya22@example.com	Mechanical
23	Harsh Vardhan	9876543232	2023-02-01	harsh23@example.com	Electrical
24	Kavya Krishnan	9876543233	2023-02-02	kavya24@example.com	Electronics
25	Ayaan Sheikh	9876543234	2023-02-03	ayaan25@example.com	Computer Science
27	Nikhil Sinha	9876543236	2023-02-05	nikhil27@example.com	Mechanical
28	Ritika Bose	9876543237	2023-02-06	ritika28@example.com	Civil
29	Aryan Choudhary	9876543238	2023-02-07	aryan29@example.com	Computer Science

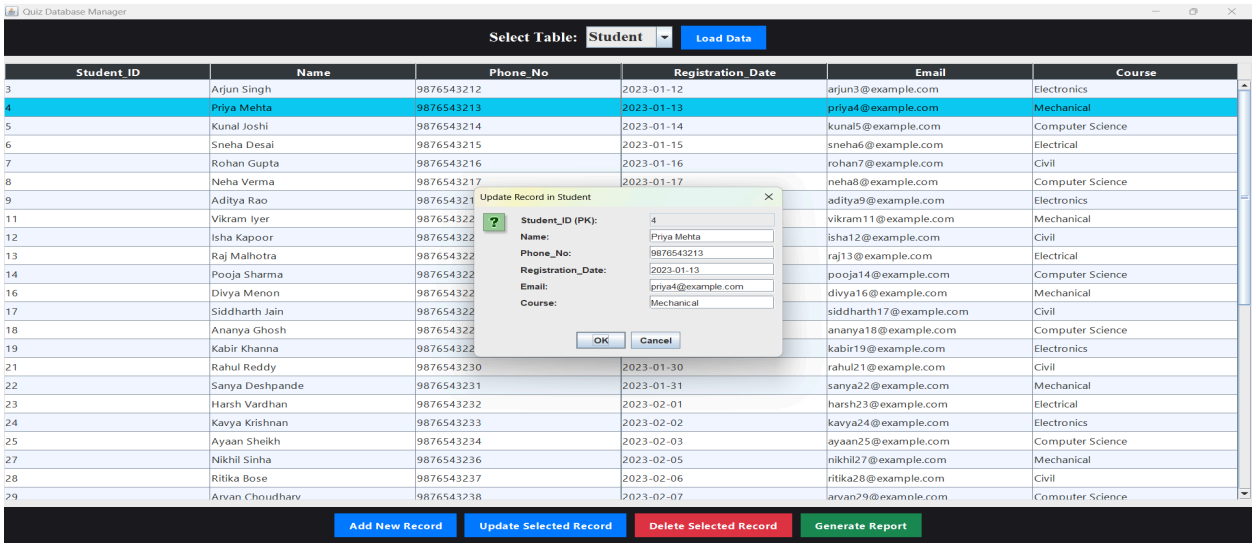
- **Add New Record**

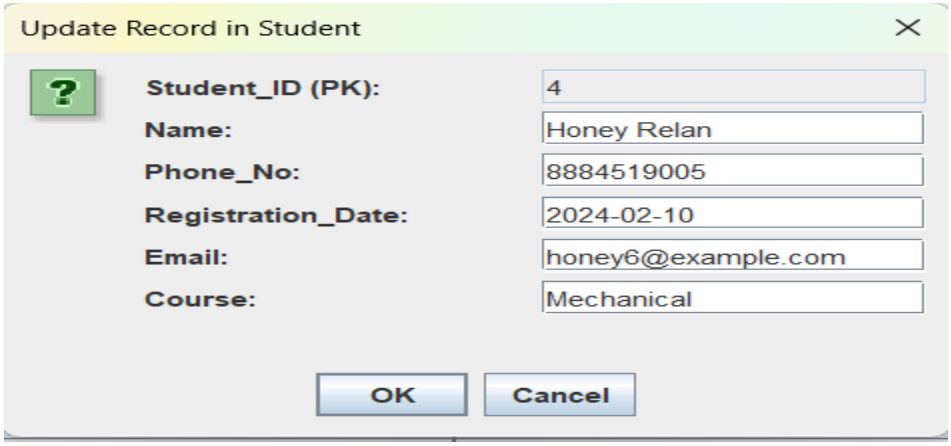


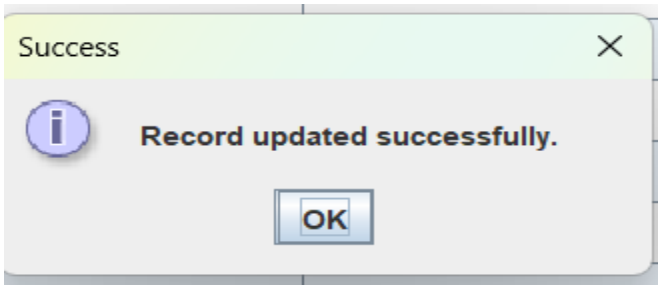


53	Gaurav Duseja	9322678611	2024-03-12	gaurav13@example....	Computer Science
56	Tisha Lakhawani	8421564652	2024-02-09	tisha1@example.com	AIDS
57	Rashmi Gidwani	9322610092	2023-02-10	rashmi13@example....	AIDS

- **Update Selected Record**

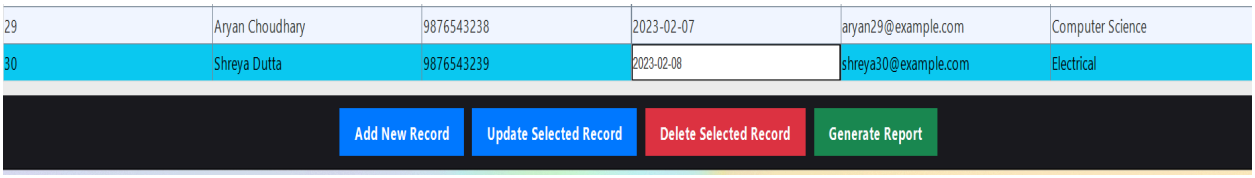


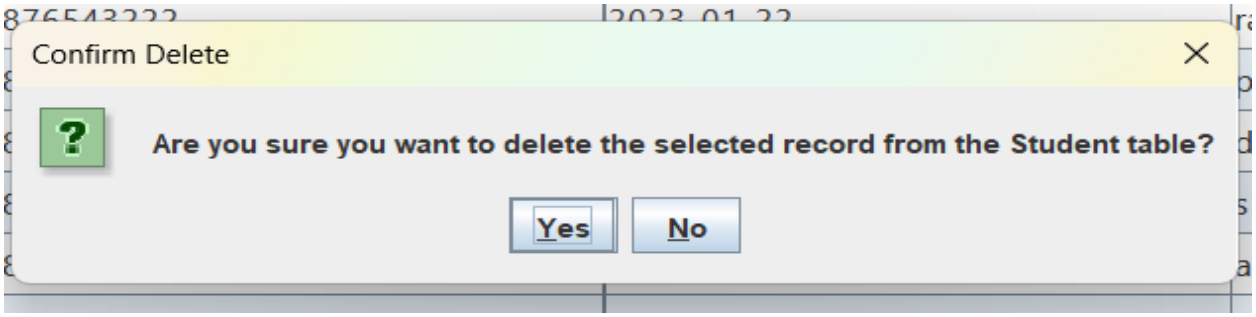


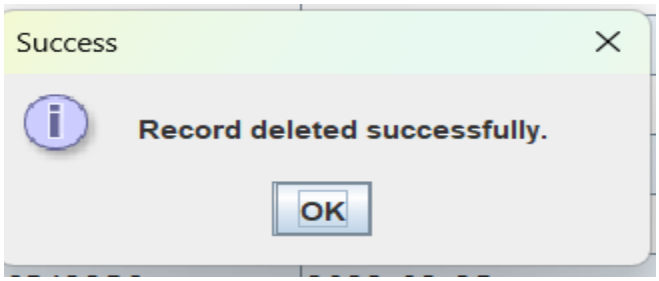


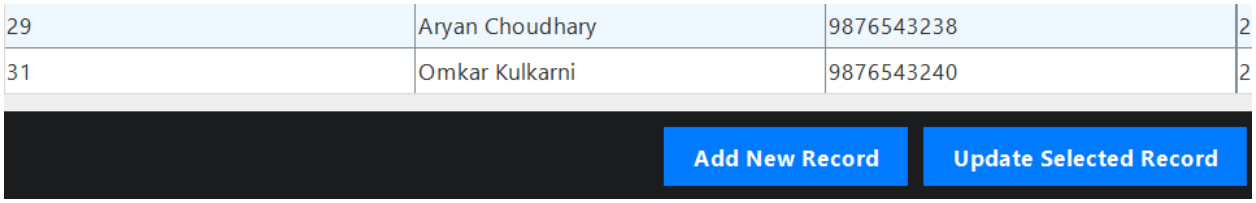
3	Arjun Singh	9876543212	2023-01-12	arjun3@example.com	Electronics
4	Honey Relan	8884519005	2024-02-10	honey6@example.com	Mechanical
5	Kunal Joshi	9876543214	2023-01-14	kunal5@example.com	Computer Science
6	Sneha Desai	9876543215	2023-01-15	sneha6@example.com	Electrical

- **Delete Selected Record**

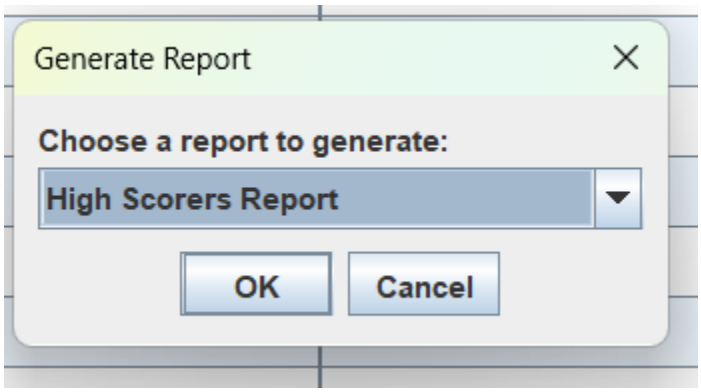


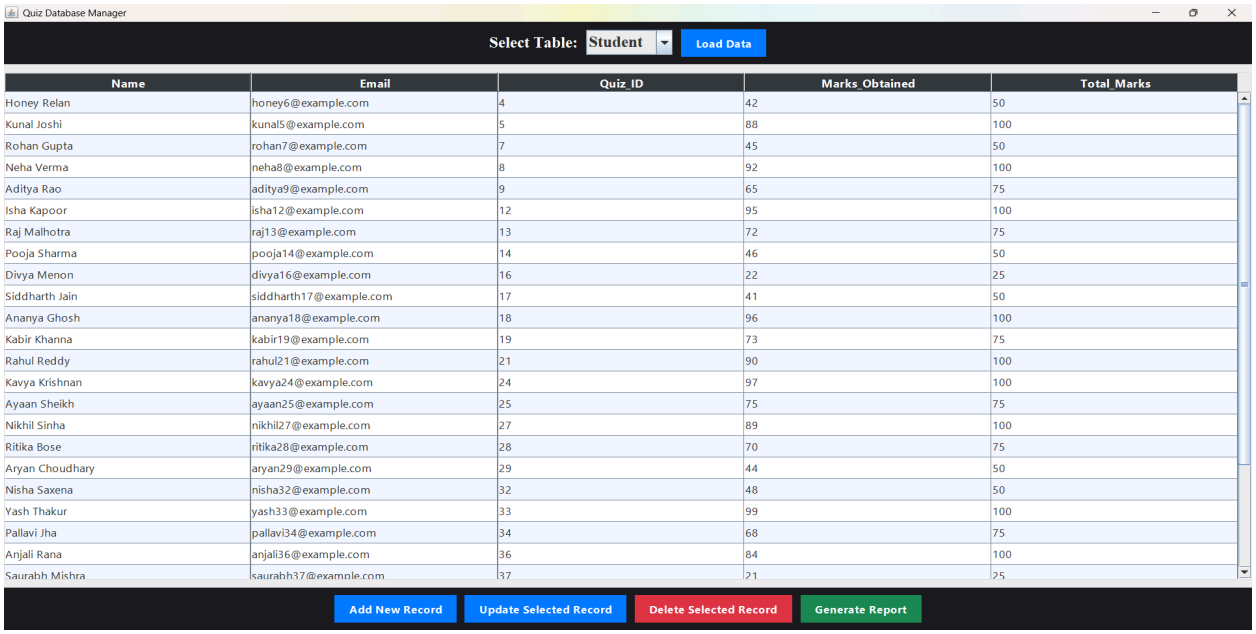






- **Generate Report**





Conclusion:

Through this project, I successfully implemented a Java Swing-based GUI that connects to a MySQL database using JDBC. I learned how to perform CRUD operations, establish a stable database connection, and design a simple yet functional interface for product management. This mini project enhanced my understanding of database connectivity, front-end integration, and real-world Java application development.