RAJALAKSHMI ENGINEERING COLLEGE

RAJALAKSHMI NAGAR, THANDALAM-602 105

CS23333 OOPS Using Java

Laboratory Record Note Book

Name :	MANISHA M
Year / Branch / Section :	BE CSE II YEAR
University Register No. :	.. 2116240701304
College Roll No. :	. 240701304
Semester :	III .
Academic Year :	2025-2026 .

RAJALAKSHMI ENGINEERING COLLEGE

An Autonomous Institution

BONAFIDE CERTIFICATE

Name: MANISHA M

Academic Year: 2025-2026 **Semester:** III **Branch:** CSE

Register No. 240701304

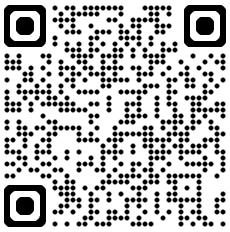
Certified that this is the bonafide record of work done by the above student in the CS23333-OOPS USING JAVA ***Laboratory during the academic year 2025- 2026***

Signature of Faculty in-charge

Submitted for the Practical Examination held on 19-11-2025

Internal Examiner **External Examiner**

INDEX

EX.NO	DATE	NAME OF THE EXPERIMENT	GITHUB QR
1	29-07-2025	I/O, Data Types, Operators	
2	5-08-2025	Control Structures	
3	12--08-2025	Arrays	
4	19-08-2025	Strings	
5	26-08-2025	Classes & Objects	
6	2-09-2025	Inheritance	
7	9-09-2025	Interface	
8	16-09-2025	Exceptions	
9	23-09-2025	Collections	
10	14-10-2025	Collections	
11	21-10-2025	Project	
12	4-11-2025	Lambda	

STUDENT RESULT MANAGEMENT SYSTEM

A MINI-PROJECT REPORT

Submitted by

240701304 MANISHA M

in partial fulfillment of the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

RAJALAKSHMI ENGINEERING COLLEGE, CHENNAI

An Autonomous Institute

CHENNAI

NOVEMBER 2025

BONAFIDE CERTIFICATE

Certified that this project **“STUDENT RESULT MANAGEMENT SYSTEM”** is the bonafide work**"MANISHA M”** who carried out the project work under my
supervision.

SIGNATURE

B.DEEPA

ASSISTANT PROFESSOR SG

Dept. of Computer Science and Engg,
Rajalakshmi Engineering College
Chennai

This mini project report is submitted for the viva voce examination to be held on

_____

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ABSTRACT

The **Student Result Management System** is designed to digitalize and simplify the process of managing student results in educational institutions. The system eliminates manual errors, ensures data accuracy, and improves transparency in academic performance management.

This project enables teachers to input marks for students, automatically calculate grades and CGPA, and allows students to securely view their performance through login credentials. Administrators can manage student data, subjects, and result reports efficiently from a centralized database.

The system provides an efficient way to store, retrieve, and analyze academic data, thereby reducing paperwork and saving time. It enhances productivity for teachers, ensures quick result generation, and maintains transparency between students and institutions.

ACKNOWLEDGEMENT

We express our sincere thanks to our beloved and honorable chairman **MR. S. MEGANATHAN** and the chairperson **DR. M.THANGAM MEGANATHAN** for their timely support and encouragement.

We are greatly indebted to our respected and honorable principal **Dr. S.N. MURUGESAN** for his able support and guidance.

No words of gratitude will suffice for the unquestioning support extended to us by our Head Of The Department **Dr. E.M. MALATHY** and our Deputy Head Of The Department **Dr. J. MANORANJINI** for being ever supporting force during our project work

We also extend our sincere and hearty thanks to our internal guide

B.Deepa , for His valuable guidance and motivation during the completion of this project.

Our sincere thanks to our family members, friends and other staff members of computer science engineering.

1. MANISHA M

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

The Student Result Management System is a digital platform developed to automate the evaluation and management of student results. It provides an easy interface for administrators, teachers, and students to interact with the system and access results efficiently.

1.2 SCOPE OF THE WORK

**The system simplifies the process of maintaining student records and results.
It allows teachers to enter marks for each subject, calculates CGPA automatically, and generates student-wise reports accessible through login credentials.**

1.3 PROBLEM STATEMENT

**Manual result processing is time-consuming and error-prone.
Institutions face difficulties in record management, data retrieval, and report generation.
This project aims to overcome these limitations by implementing a reliable and user-friendly result management system.**

1.4 AIM AND OBJECTIVES OF THE PROJECT

**aim of the project is to design an efficient and secure system to manage academic results.
Objectives include:**

1.Automate result entry and calculation.

2.Provide secure login for admin, teachers, and students.

3.Enable instant report generation and Maintain transparency and accessibility

CHAPTER 2

SYSTEM SPECIFICATIONS

2.1 HARDWARE SPECIFICATIONS

Processor	:	ultra core 5
Memory Size	:	16GB (Minimum)
HDD	:	1 TB (Minimum)

2.2 SOFTWARE SPECIFICATIONS

Operating System	:	WINDOWS 11
Front – End	:	java(java swing)
Back - End	:	MySql
Language	:	java,SQL

CHAPTER 3

MODULE DESCRIPTION

The Student Result Management System consists of multiple interconnected modules designed to streamline the process of storing, updating, and retrieving student result information. Each module performs specific functions to ensure smooth data flow and efficient operation of the entire system. The modules are explained below:

Admin login

In the **Admin Panel**, the administrator can perform several key operations, such as:

Add Student: Enter new student information, including name, roll number, department, and semester details.

Edit / Update Student: Modify student data whenever necessary, such as updating marks or personal details.

Manage Subjects: Add or remove subjects associated with different semesters and departments.

Add Department: Create and manage departments to categorize students and teachers systematically.

Add Teacher: Register new teachers in the system and assign them to their respective departments and subjects.

Generate Reports: Create and export overall academic reports that summarize performance across students, departments, and semesters.

.

2.Teacher Module

The **Teacher Module** is responsible for academic data entry and evaluation tasks.
Teachers can:

- Log in securely using their credentials.
- Enter marks or grades for each subject and student.
- Modify or update results if corrections are required.
- View performance summaries for students within their subject.

3. Student Module

The **Student Module** provides a user-friendly interface for students to access their personal academic information.
Each student can:

- Log in with their unique username and password.
- View individual subject marks, total marks, grades, and overall CGPA.
- Track their academic performance across semesters.

This module enhances transparency, allowing students to instantly view their results once they are approved by the admin. It promotes digital accessibility and reduces dependency on printed mark sheets.

4. Database Module

The **Database Module** serves as the foundation of the entire system. It stores and manages all records, including student details, teacher credentials, subjects, marks, and result statistics.
Developed using **MySQL**, it provides:

- Secure data storage and retrieval.
- Fast query execution for report generation.
- Data integrity and backup features to prevent data loss.
- Smooth communication between the front-end (Java Swing interface) and back-end database.

This module ensures the system operates with reliability, scalability, and consistency, supporting real-time data access for all users.

CHAPTER 4

SAMPLE CODING

Sample 1

Shows how your Java program connects to MySQL using JDBC.

```
package com.recresultsystem;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

public class DBConnection {
   private static final String URL = "jdbc:mysql://localhost:3306/rec_result_system";
   private static final String USER = "root";
   private static final String PASSWORD = "Kaushika@2510";

   public static Connection getConnection() throws SQLException {
      try {
         Class.forName("com.mysql.cj.jdbc.Driver");
      } catch (ClassNotFoundException e) {
         throw new SQLException("MySQL JDBC Driver not found.", e);
      }
      return DriverManager.getConnection(URL, USER, PASSWORD);
   }
}
```

Sample 2

Sample 2 depicts how your program calculates CGPA based on subject marks.

```
package com.recresultsystem;
import java.sql.*;
import javax.swing.*;
public class CGPACalculator {
   public static void calculateAndUpdateCGPA(int studentId) {
      try (Connection conn = DBConnection.getConnection()) {
         String query = "SELECT marks FROM results WHERE student_id = ?";
         PreparedStatement ps = conn.prepareStatement(query);
         ps.setInt(1, studentId);
         ResultSet rs = ps.executeQuery();

         int subjectCount = 0;
         double totalPoints = 0.0;

         while (rs.next()) {
            double marks = rs.getDouble("marks");
            totalPoints += getGradePoint(marks);
            subjectCount++;
         }

         double cgpa = (subjectCount == 0) ? 0 : totalPoints / subjectCount;

         PreparedStatement update = conn.prepareStatement(
               "UPDATE students SET cgpa=? WHERE student_id=?");
         update.setDouble(1, cgpa);
         update.setInt(2, studentId);
         update.executeUpdate();
```

```
            System.out.println("CGPA updated for student ID: " + studentId + " = " + cgpa);
        } catch (SQLException e) {
            JOptionPane.showMessageDialog(null, "Error calculating CGPA: " + e.getMessage());
        }
    }
    private static int getGradePoint(double marks) {
        if (marks >= 90) return 10;
        else if (marks >= 80) return 9;
        else if (marks >= 70) return 8;
        else if (marks >= 60) return 7;
        else if (marks >= 50) return 6;
        else if (marks >= 40) return 5;
        else return 0;
    }
    public static void showCGPA(int studentId) {
        try (Connection conn = DBConnection.getConnection()) {
            PreparedStatement ps = conn.prepareStatement(
                    "SELECT name, cgpa FROM students WHERE student_id=?");
            ps.setInt(1, studentId);
            ResultSet rs = ps.executeQuery();
            if (rs.next()) {
                String name = rs.getString("name");
                double cgpa = rs.getDouble("cgpa");
                JOptionPane.showMessageDialog(null,
                        "Student: " + name + "\nCGPA: " + String.format("%.2f", cgpa),
                        "CGPA Report", JOptionPane.INFORMATION_MESSAGE);
            } else {
                JOptionPane.showMessageDialog(null, "Student not found!");
            }
        } catch (SQLException e) {
            JOptionPane.showMessageDialog(null, "Error displaying CGPA: " + e.getMessage());
        }
    }
}
```

CHAPTER 5

SCREEN SHOTS

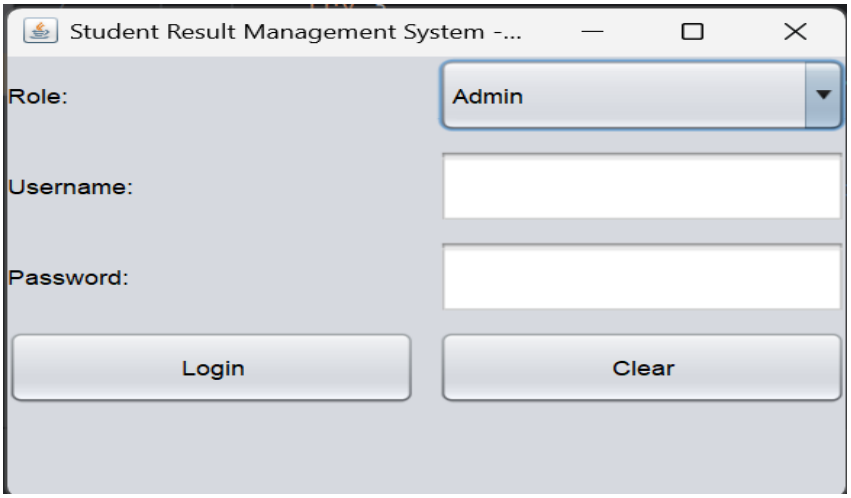


Fig 5.1 LOGIN PAGE

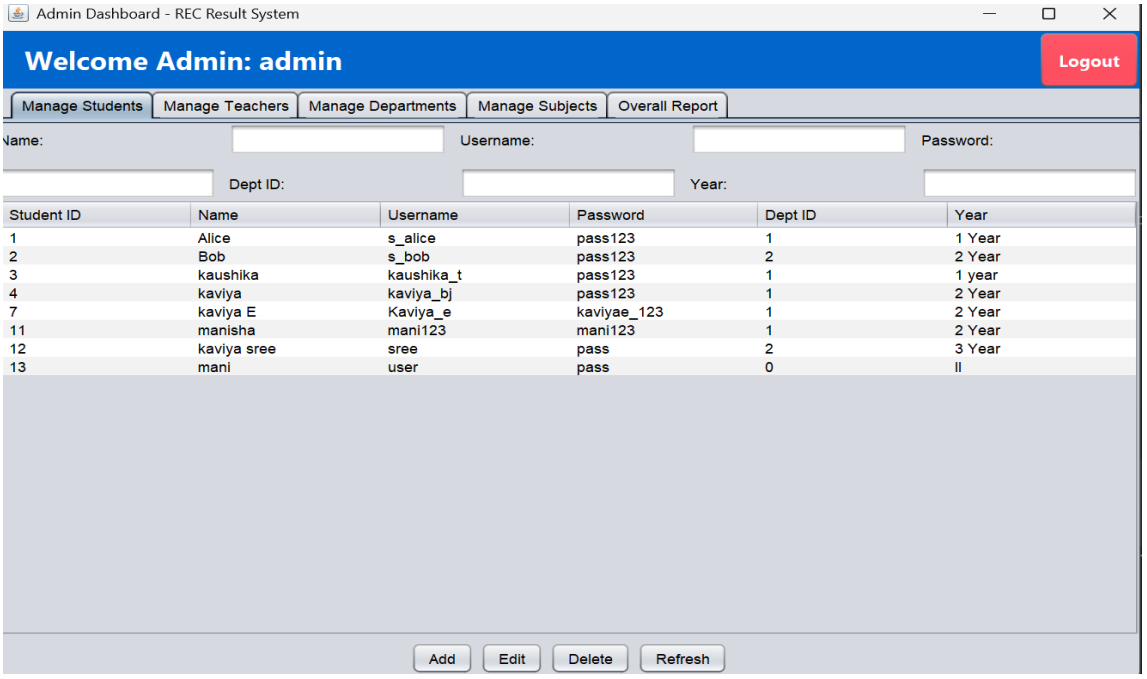


Fig 5.2 ADMIN PANEL(manage student)

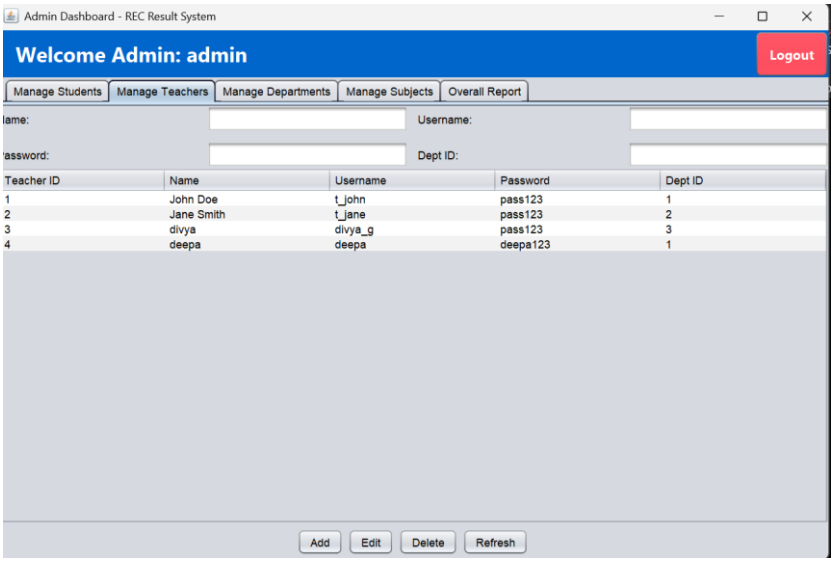


Fig 5.2.1 manage Teachers

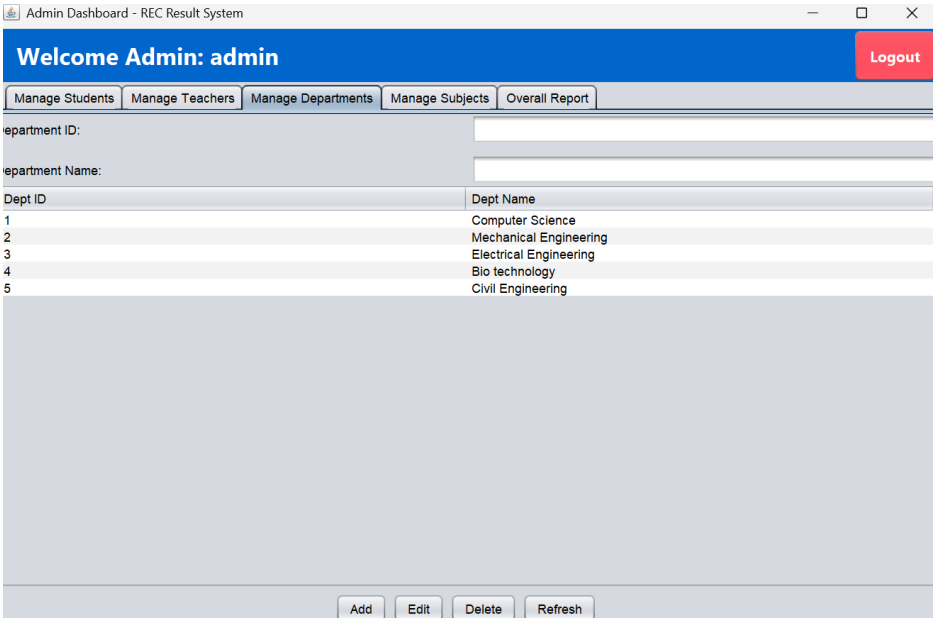


Fig 5.2.2 manage Department

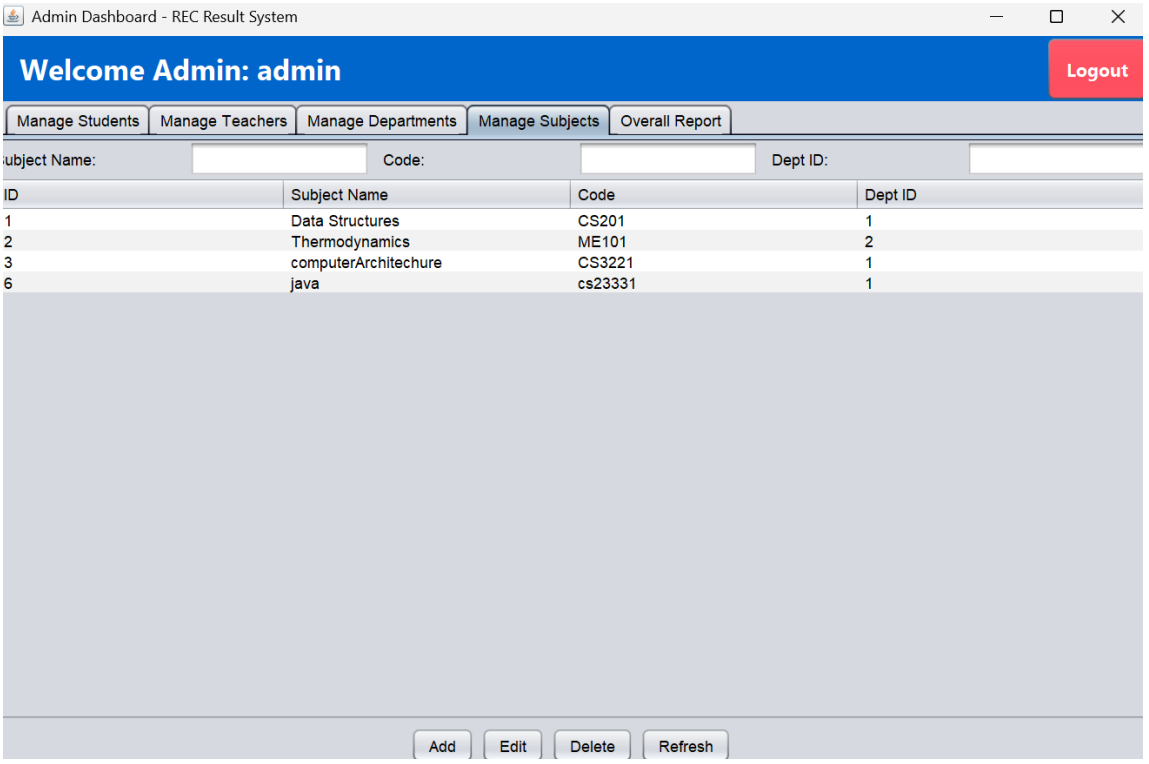


Fig 5.2.3.manage subject

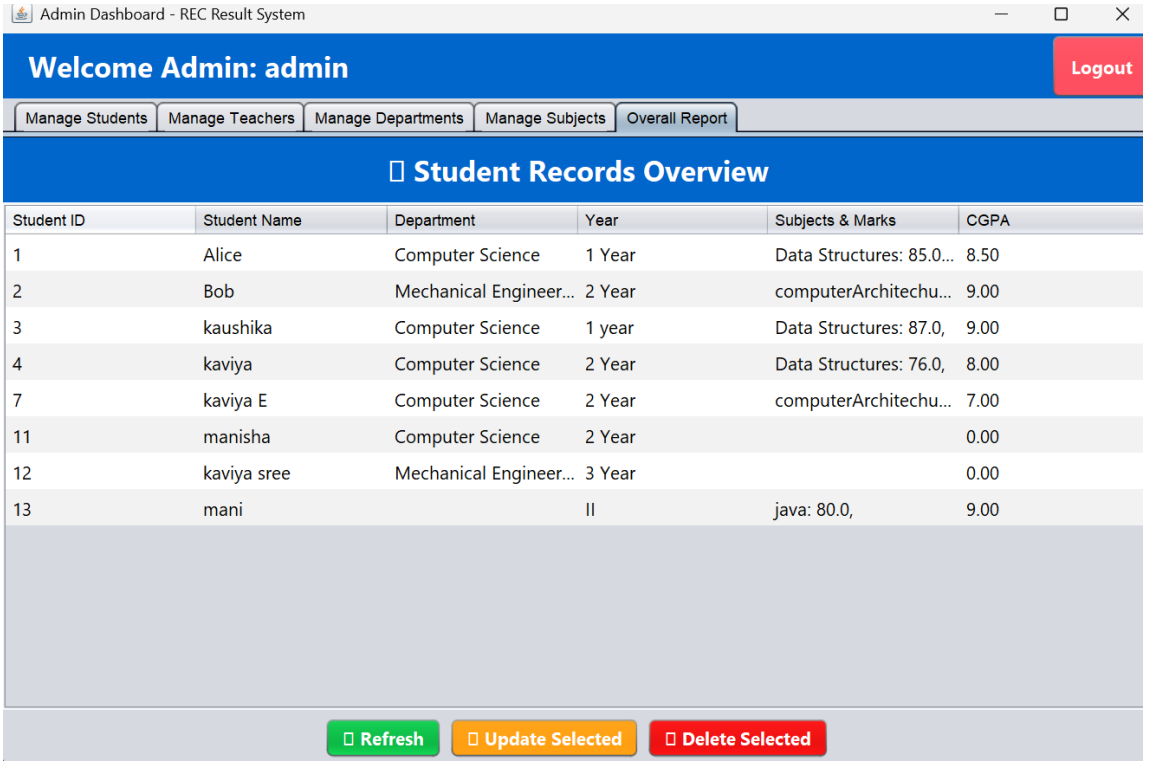


FIG 5.2.4 OVERALL REPORT

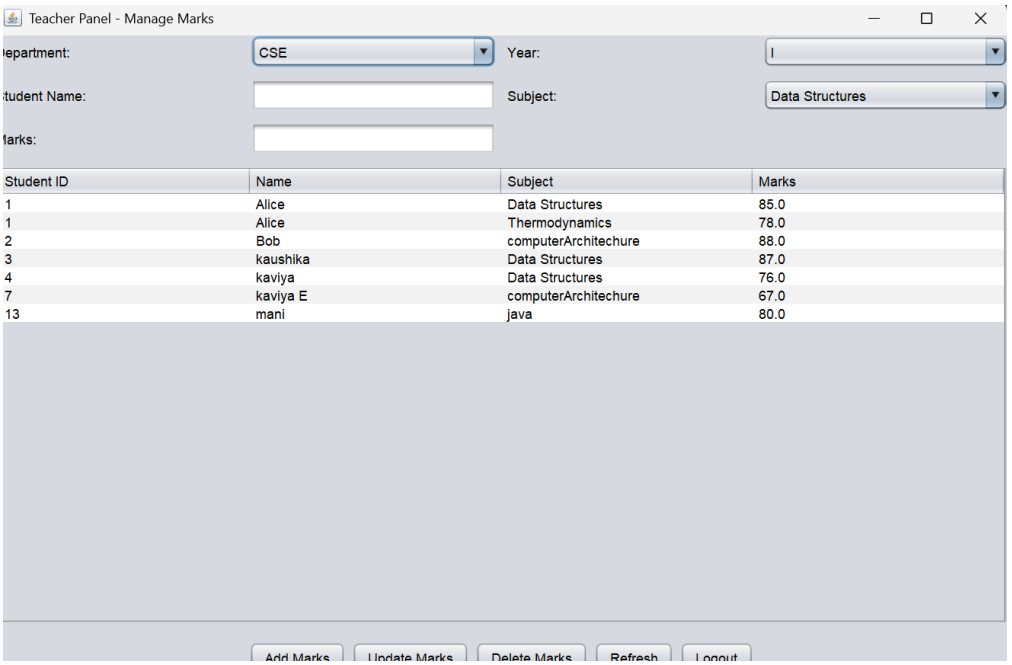


Fig 5.3 Teacher panel

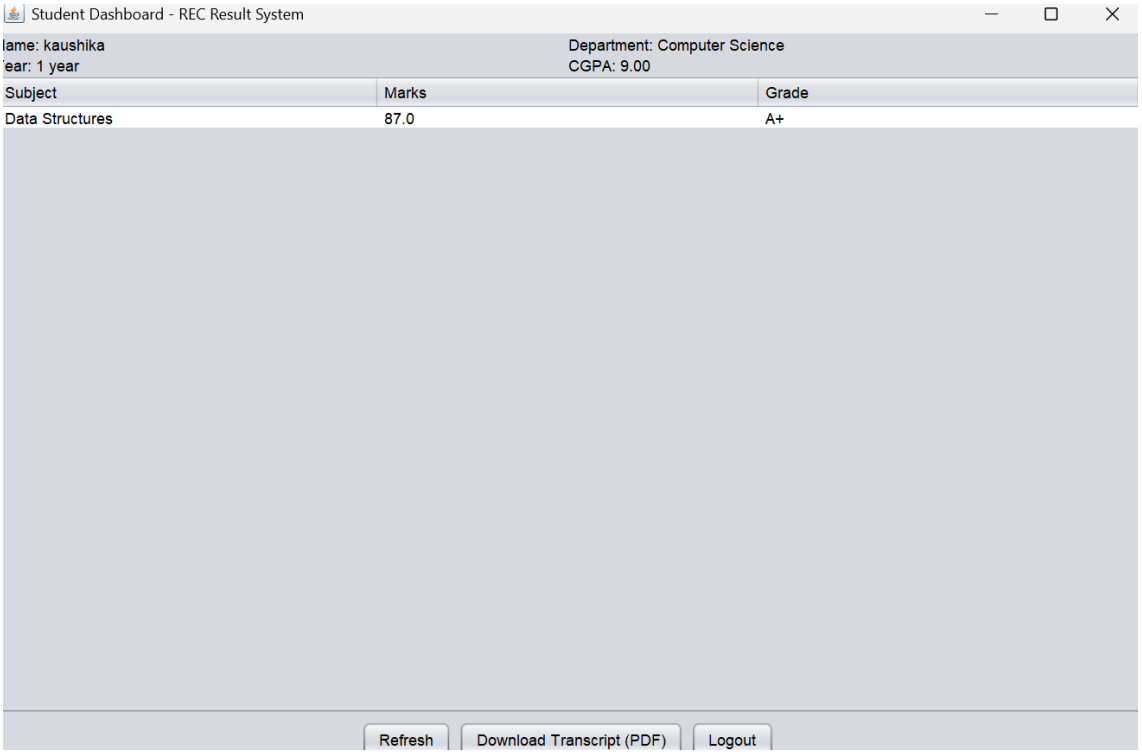
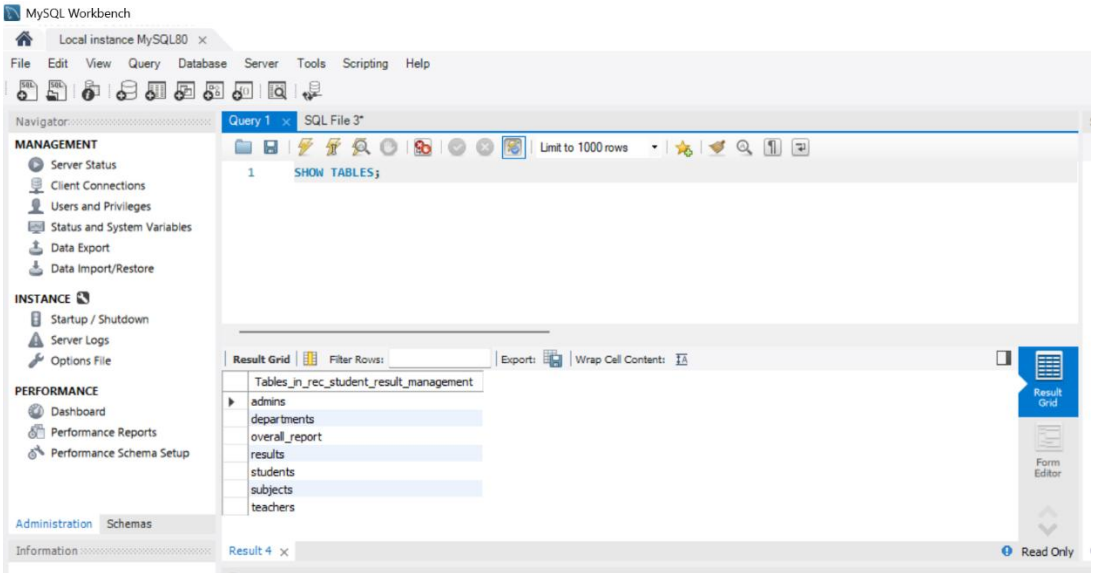
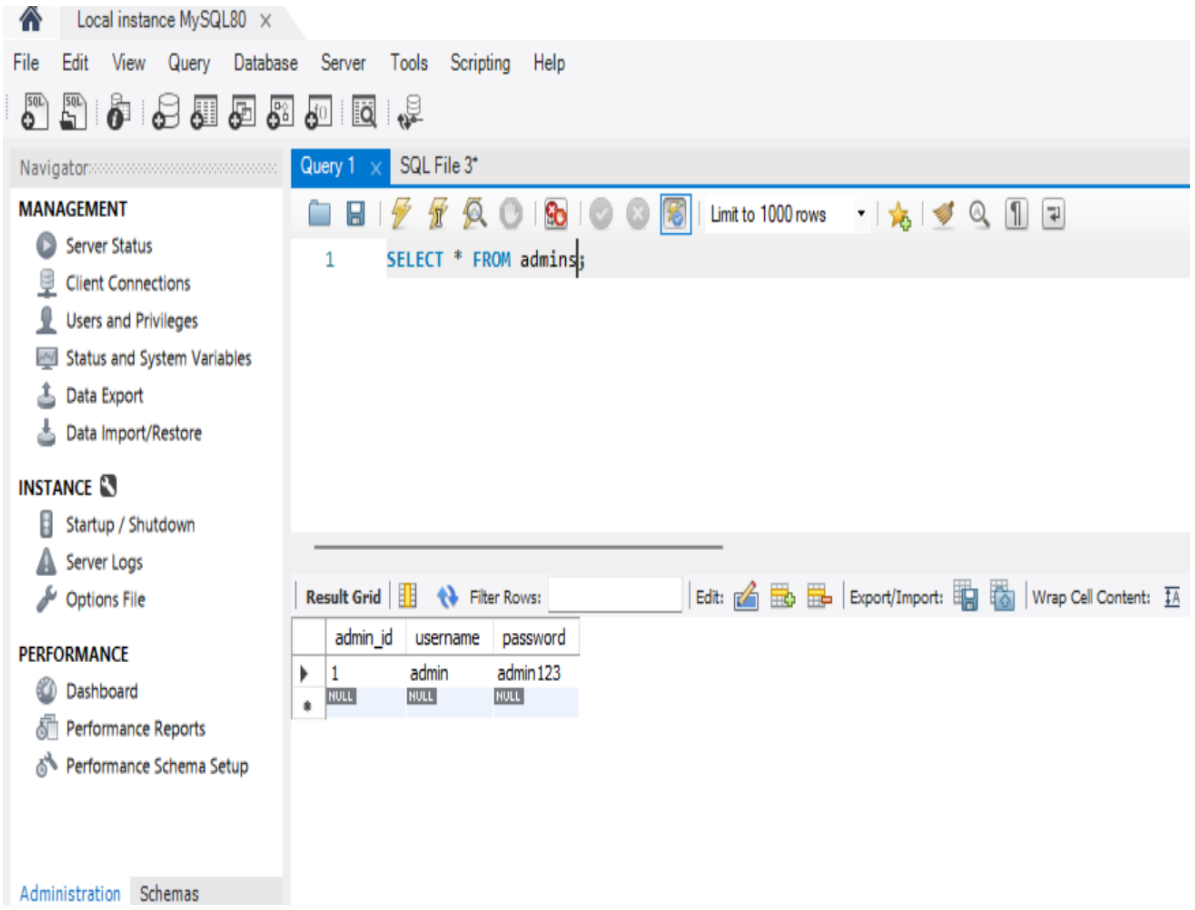
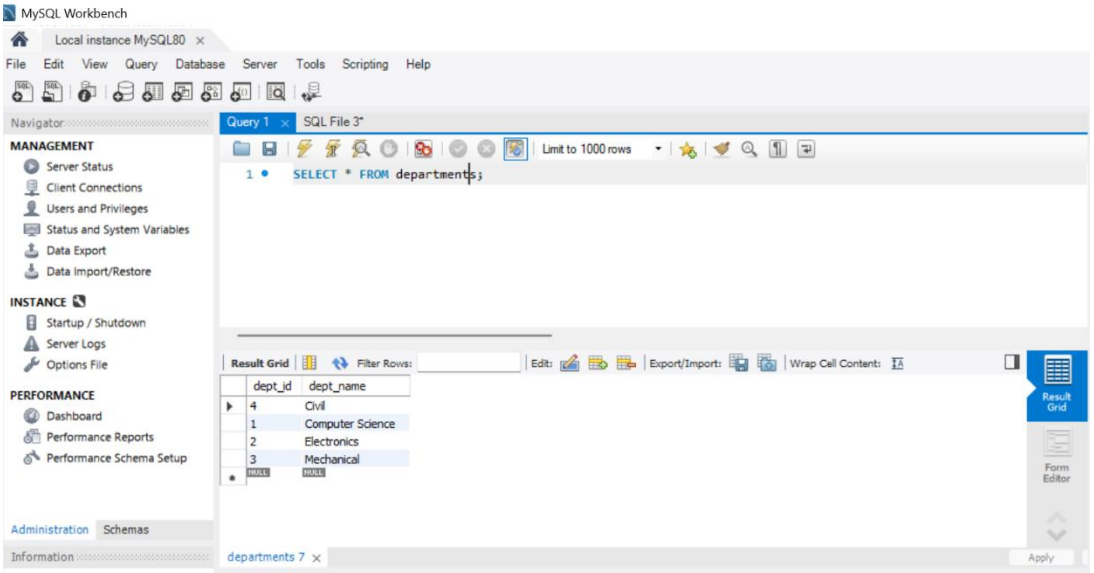


Fig 5.4.Student panel




Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
Navigator:
MANAGEMENT
Server Status
Client Connections
Users and Privileges
Status and System Variables
Data Export
Data Import/Restore
INSTANCE
Startup / Shutdown
Server Logs
Options File
PERFORMANCE
Dashboard
Performance Reports
Performance Schema Setup
Administration Schemas
Query 1
SQL File 3*
Limit to 1000 rows
1 SELECT * FROM admins;
Result Grid
Filter Rows:
Edit:
Export/Import:
Wrap Cell Content:
admin_id username password
1 admin admin123
NULL NULL NULL


MySQL Workbench
Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
Navigator
MANAGEMENT
Server Status
Client Connections
Users and Privileges
Status and System Variables
Data Export
Data Import/Restore
INSTANCE
Startup / Shutdown
Server Logs
Options File
PERFORMANCE
Dashboard
Performance Reports
Performance Schema Setup
Administration Schemas
Information
Query 1
SQL File 3*
Limit to 1000 rows
1 SELECT * FROM departments;
Result Grid
Filter Rows:
Edit:
Export/Import:
Wrap Cell Content:
dept_id dept_name
4 Civil
1 Computer Science
2 Electronics
3 Mechanical
NULL NULL
Result Grid
Form Editor
departments 7
Apply

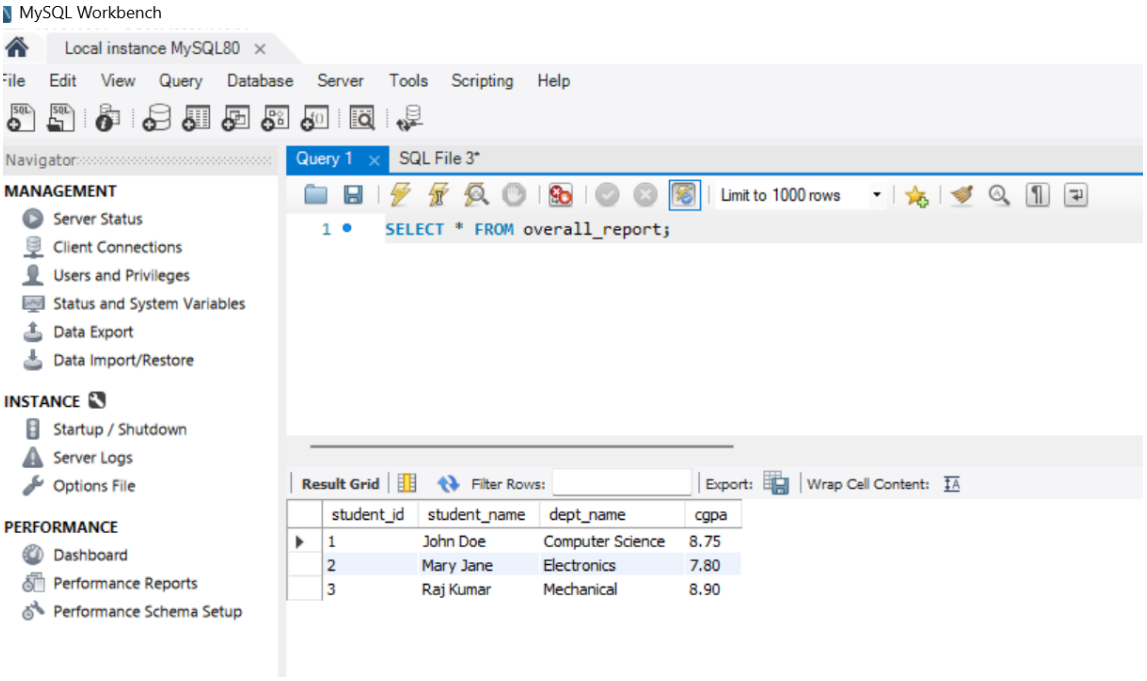


Fig 5.5 Database creation

CHAPTER 6

CONCLUSION AND FUTURE ENHANCEMENT

digitalizes the manual process of managing student results in educational institutions. It The **Student Result Management System** is a highly efficient application that successfully eliminates redundant paperwork, reduces human errors, and saves time by automating the process of result entry, calculation, and report generation.

Through its role-based modules — Admin, Teacher, and Student — the system ensures smooth data flow and secure access.
Administrators can maintain records, teachers can efficiently enter and update marks, and students can instantly view their performance through an easy-to-use interface.

The project enhances transparency between faculty and students while maintaining the accuracy and integrity of academic data. It demonstrates how database management systems and Java-based front-end technologies can be effectively combined to develop a practical, real-world academic solution.

REFERENCES

1. https://www.w3schools.com/sql/
2. https://www.tutorialspoint.com/sqlite/index.htm
3. https://www.wikipedia.org/
4. https://www.learnpython.org/
5. https://www.codecademy.com/learn/learn-python