EduBridge

A Project Report

On

"SCHOOL MANAGEMENT SYSTEM"

By

LOGESH M

Batch: 2022 – 7671

Center: C h e n n a i c h r o m p e t

Under the Guidance of,

Chittaranjan Ghosh.

Technical Trainer

EduBridge

ABSTRACT

School Management System is a large database system which can be used for managing your school's day to day business. School Management System allows users to store almost all of their school's information electronically, including information on students, properties, teaching meteorites etc. Most importantly, this information can be easily shared with authorized users, records can be easily searched, and reports can be easily generated School Management System is software which is helpful for students as well as the school authorities. In the current system all the activities are done manually. It is very time consuming and costly. Our School Management System deals with the various Activities. An efficient Student Management Solution that, enables in maintaining and give the promotions and remarks for the student in the School management system.

INTRODUCTION

System Overview

The title of the project is “School Management system”. This project will handle the whole activities of the school. SMS has most of the facilities that a modern school requires to computerize its day-to-day jobs. It provides facilities to keep the records of student, fees, teaching and non-teaching staff with all their required details along with all required transaction handling. It has facilities to generate various types of reports, which are required by the management during normal business operations to operate the business effectively.

Existing system

In the earlier system, it was difficult to store all the student, teacher and staff records which required a lot of paperwork and time consumption. It also leads to errors and mistakes. Everything like attendance, fees and results have to be marked manually. These systems also caused redundancy of data. If any records need to be updated then it has to be changed in every file so making it difficult to manage all the data. There was no security as anyone could view those records. So there was the need for developing an automated system which could handle all these problems making the system easier to use.

Proposed system

Now the proposed system is developed to solve all the problems that occurred in the earlier systems. It provides a user-friendly interface where there is an administrator which only has the privilege to access the data. This makes the system Easy and reliable. This system allows to keep track of the student details and maintain the staff information. So everything becomes easy thus reducing the risks that come with an existing system.

School Management System Modules

- **Add module:** This allows adding new users to the system database.
- **Show module:** This allows Show the records in a database.
- **Delete module:** It deleted the student records from a database.

- **Update module:** This module displays the Updated Information of the database.

- **Exit module:** This module is use to exit from the Information of the database.

Software Requirements

- JVM(Eclipse IDE)
- MYSQL Workbench 8.0

Hardware Requirements

- Hard Disk – 2 GB.
- RAM required – 2 GB (minimum)
- Processor – Dual Core or Above.
- OS supported: Windows, Linux

Technology Used

- Java
- MySql

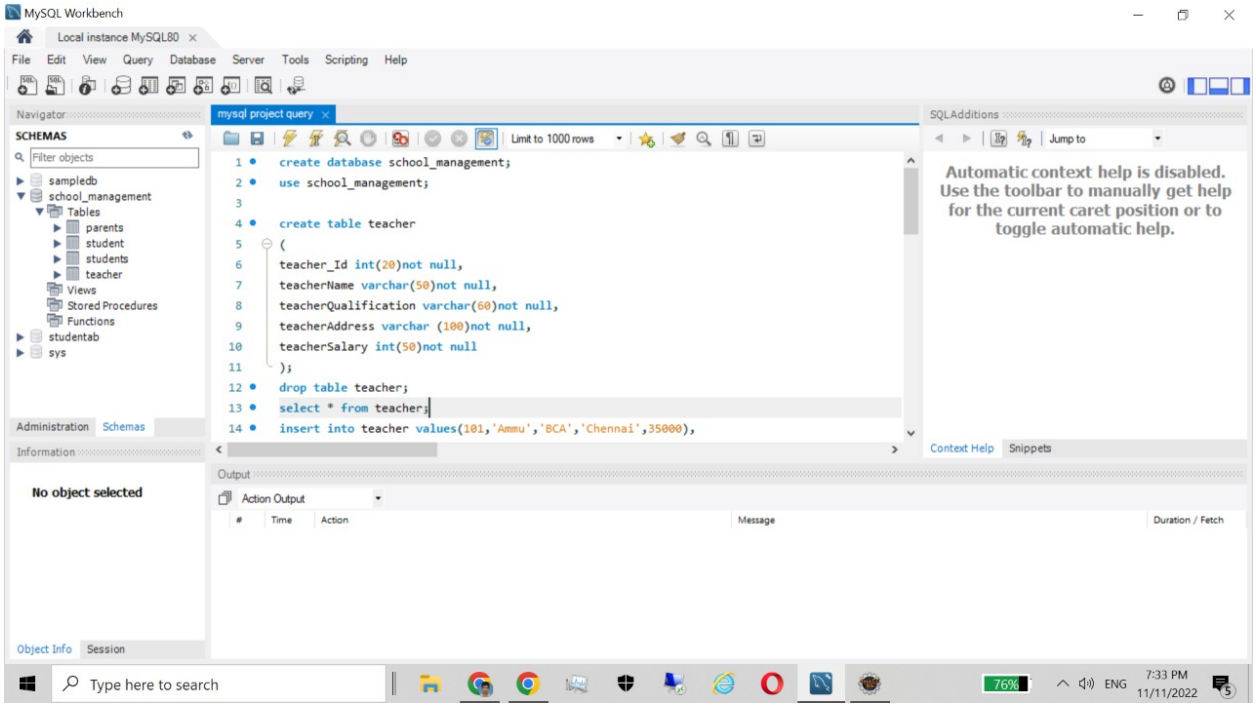
System Design

System Design is the solution to the creation of a new system. This is the important aspect made up of several steps. The complete, efficient and successful system should provide the following in succession .If the project is to be successful, we will need to answer these questions. The answer of these questions is schema manner and is known as system design. A systematic manner will be followed so

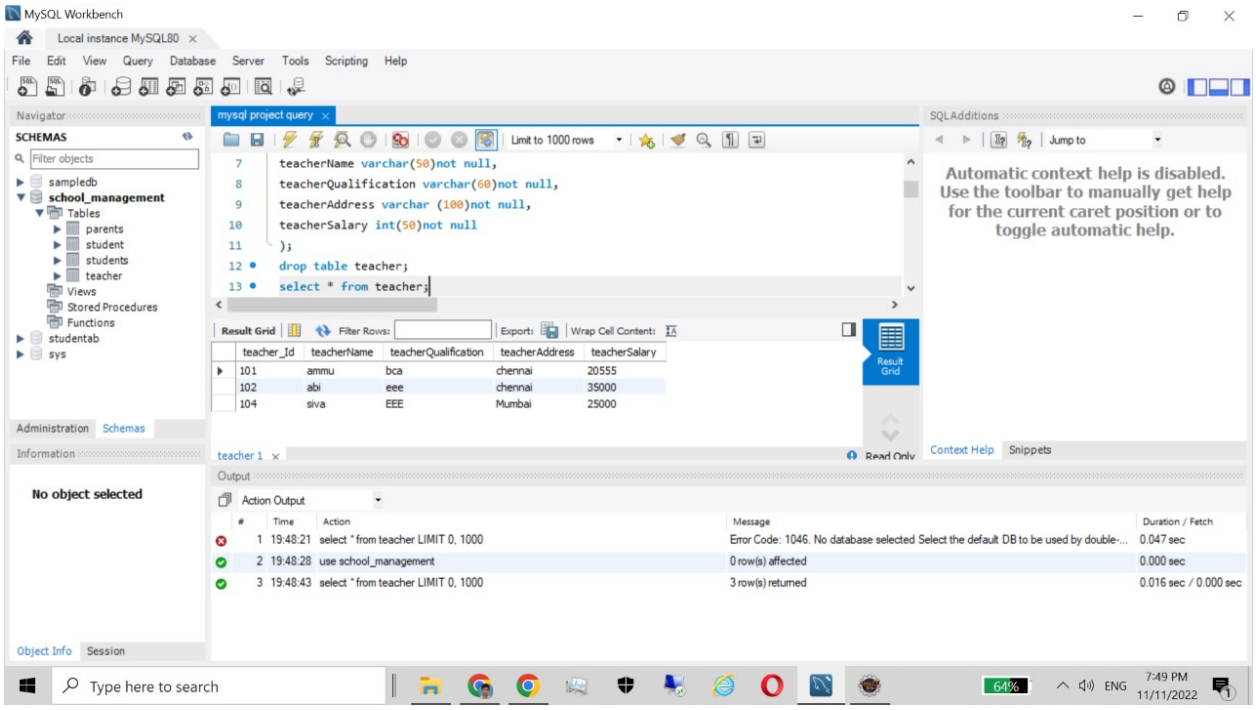
as to achieve beneficial results at the end. Software report can be broken into a series of steps starting with the basic ideas and ending with the finished project.

Data Dictionary

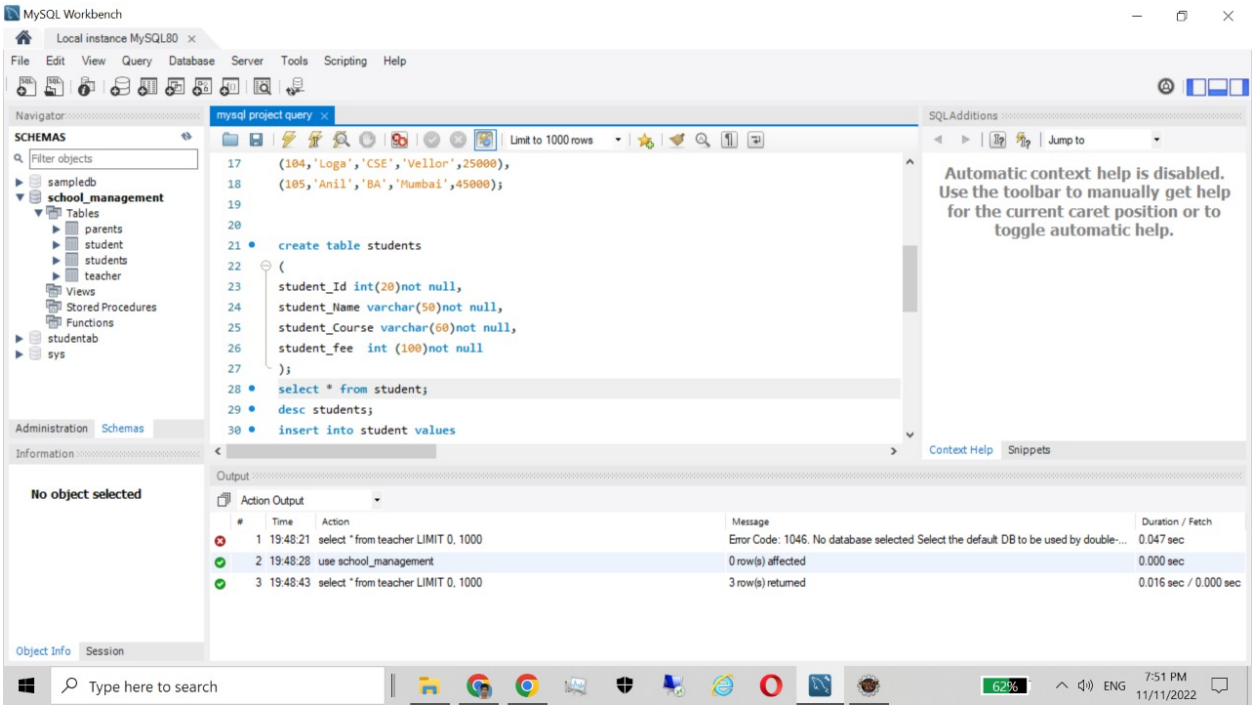
Creating Database



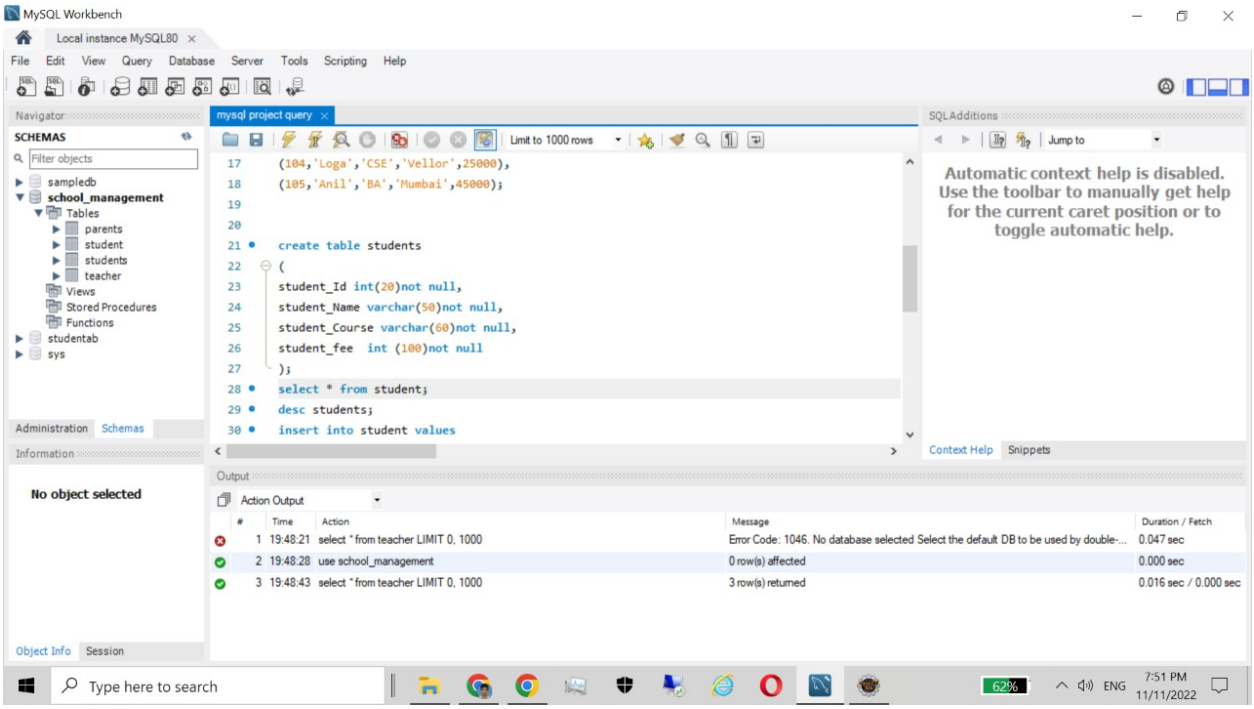
Creating Teacher Table



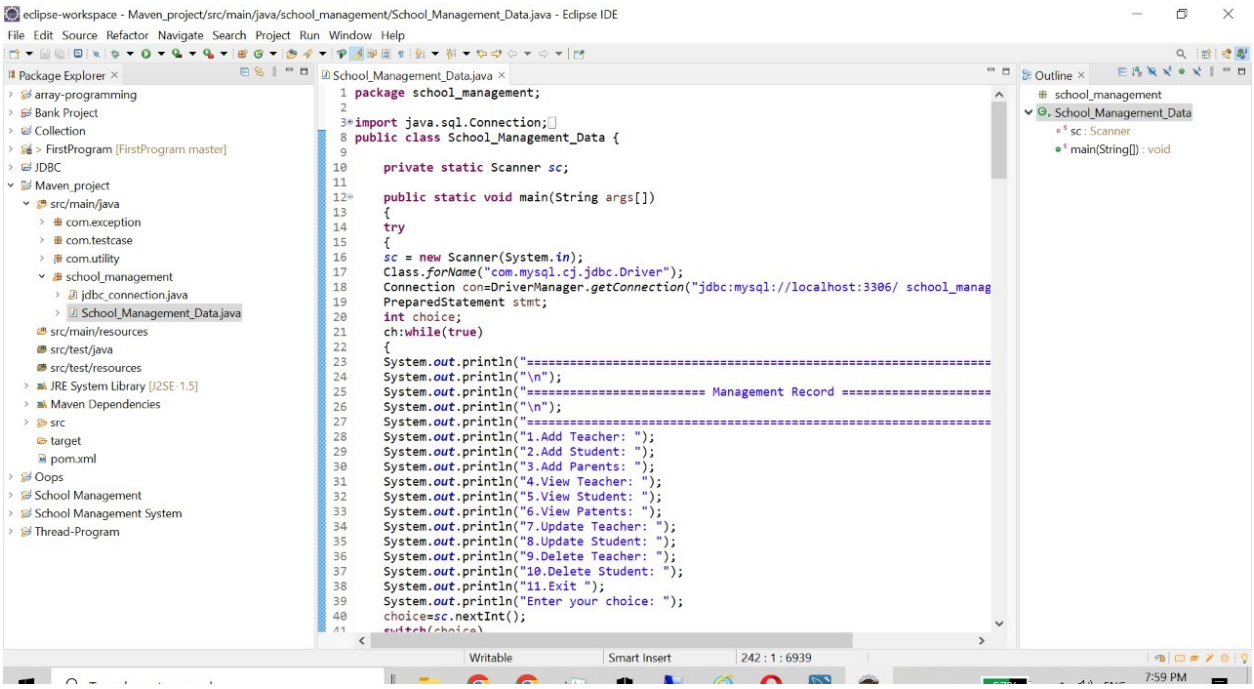
Creating Student Table

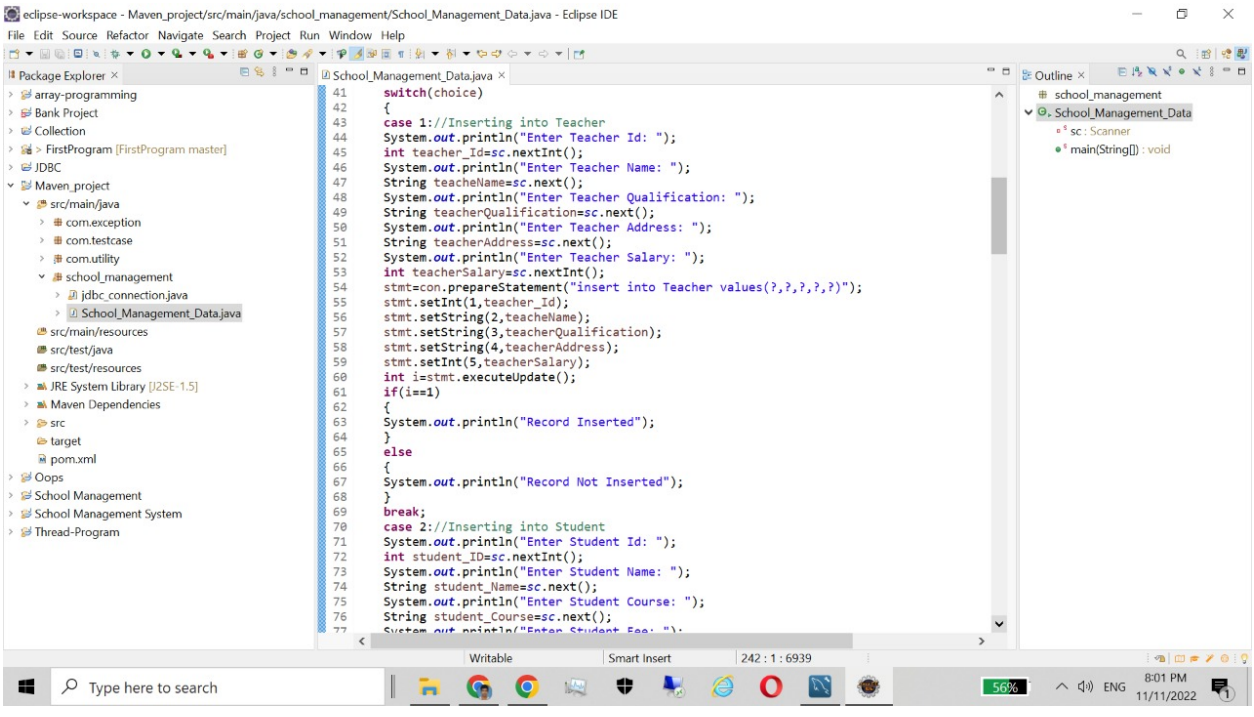


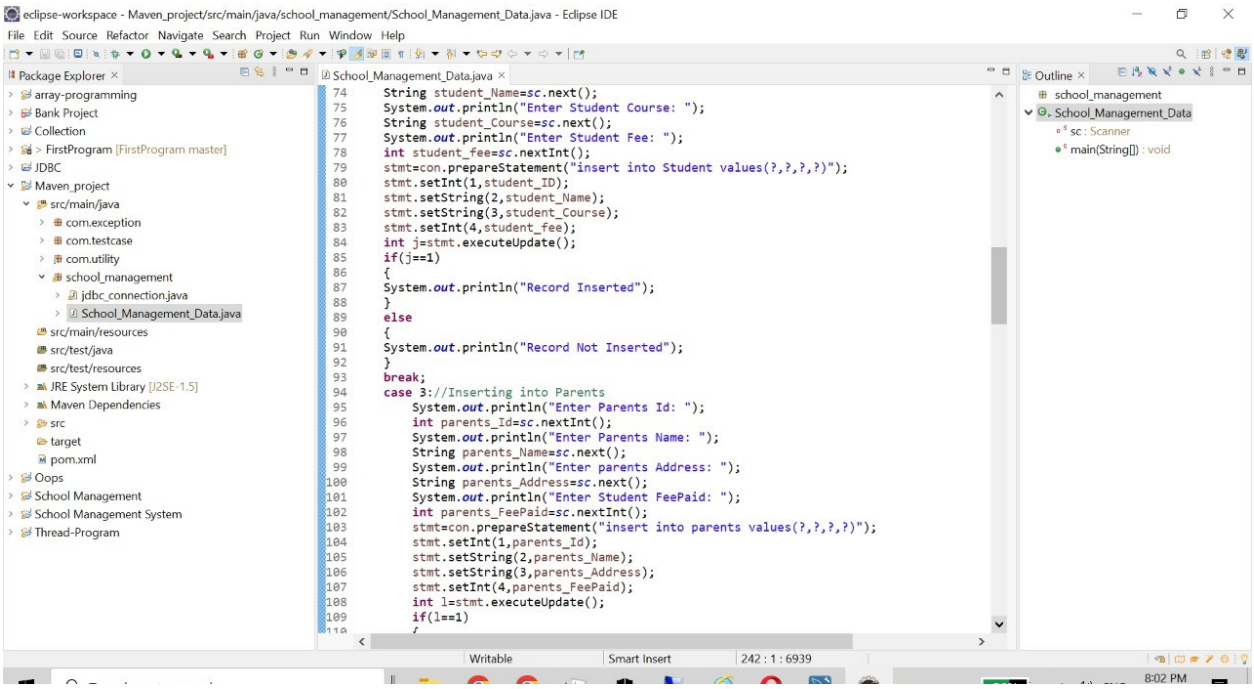
Creating Parent Table

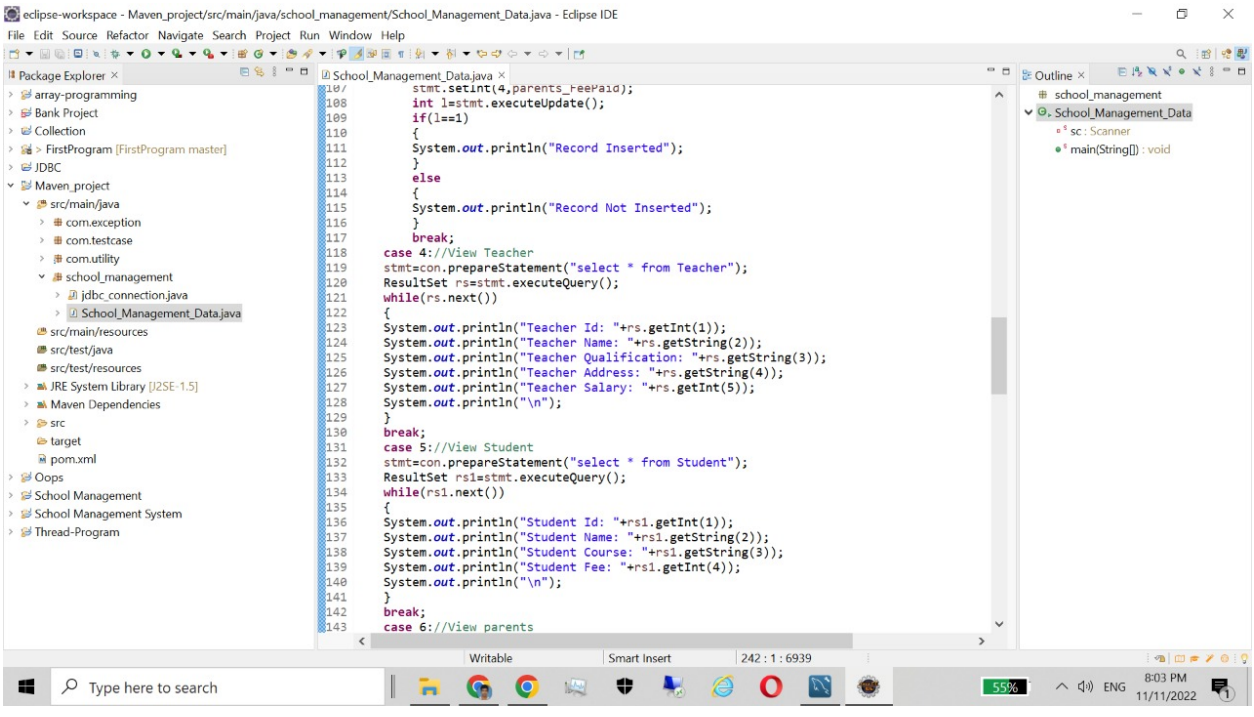


Code :


eclipse-workspace - Maven_project/src/main/java/school_management/School_Management_Data.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
array-programming
Bank Project
Collection
FirstProgram [FirstProgram master]
JDBC
Maven_project
src/main/java
com.exception
com.testcase
com.utility
school_management
jdbc_connection.java
School_Management_Data.java
src/main/resources
src/test/java
src/test/resources
JRE System Library [J2SE-1.5]
Maven Dependencies
src
target
pom.xml
Oops
School Management
School Management System
Thread-Program
School_Management_Data.java
1 package school_management;
2
3 import java.sql.Connection;
8 public class School_Management_Data {
9
10 private static Scanner sc;
11
12 public static void main(String args[])
13 {
14 try
15 {
16 sc = new Scanner(System.in);
17 Class.forName("com.mysql.cj.jdbc.Driver");
18 Connection con=DriverManager.getConnection("jdbc:mysql://localhost:3306/ school_manag
19 PreparedStatement stmt;
20 int choice;
21 ch:while(true)
22 {
23 System.out.println("==========
24 System.out.println("\n");
25 System.out.println("========== Management Record ==========
26 System.out.println("\n");
27 System.out.println("==========
28 System.out.println("1.Add Teacher: ");
29 System.out.println("2.Add Student: ");
30 System.out.println("3.Add Parents: ");
31 System.out.println("4.View Teacher: ");
32 System.out.println("5.View Student: ");
33 System.out.println("6.View Patents: ");
34 System.out.println("7.Update Teacher: ");
35 System.out.println("8.Update Student: ");
36 System.out.println("9.Delete Teacher: ");
37 System.out.println("10.Delete Student: ");
38 System.out.println("11.Exit ");
39 System.out.println("Enter your choice: ");
40 choice=sc.nextInt();
Outline
school_management
School_Management_Data
sc : Scanner
main(String[]) : void
Writable
Smart Insert
242 : 1 : 6939
7:59 PM

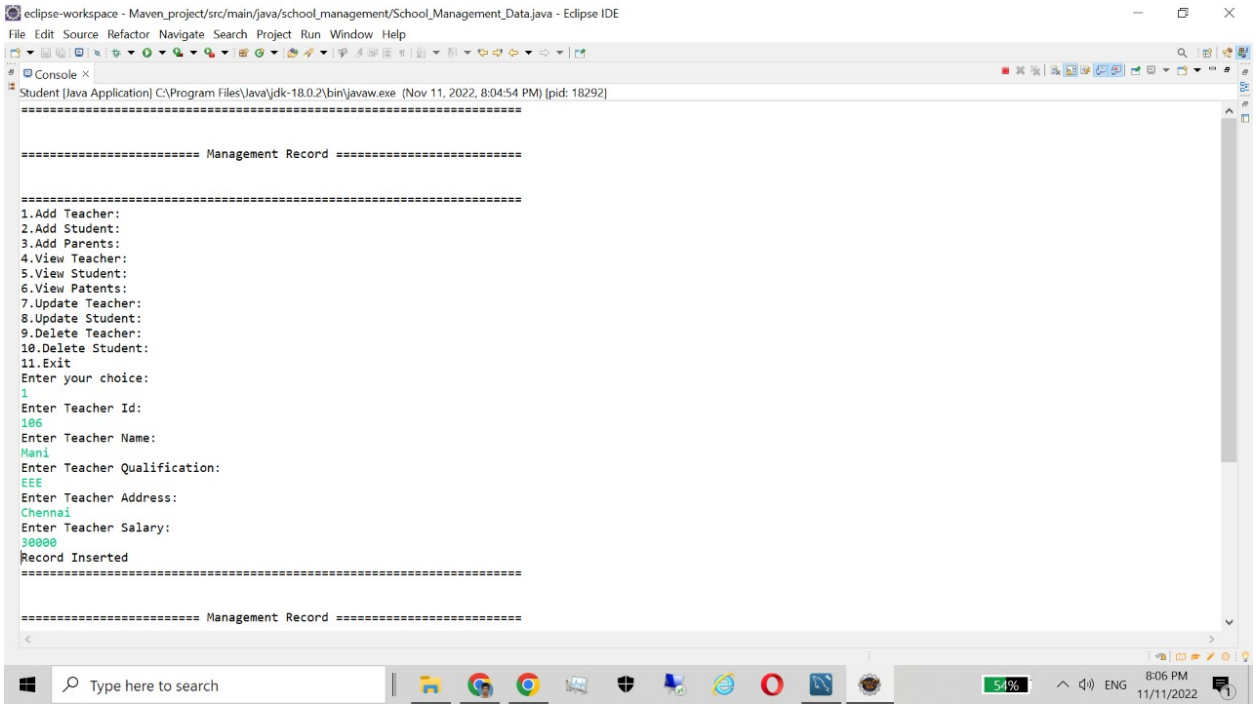

eclipse-workspace - Maven_project/src/main/java/school_management/School_Management_Data.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
array-programming
Bank Project
Collection
FirstProgram [FirstProgram master]
JDBC
Maven_project
src/main/java
com.exception
com.testcase
com.utility
school_management
jdbc_connection.java
School_Management_Data.java
src/main/resources
src/test/java
src/test/resources
JRE System Library [J2SE-1.5]
Maven Dependencies
src
target
pom.xml
Oops
School Management
School Management System
Thread-Program
School_Management_Data.java
41 switch(choice)
42 {
43 case 1://Inserting into Teacher
44 System.out.println("Enter Teacher Id: ");
45 int teacher_Id=sc.nextInt();
46 System.out.println("Enter Teacher Name: ");
47 String teacheName=sc.next();
48 System.out.println("Enter Teacher Qualification: ");
49 String teacherQualification=sc.next();
50 System.out.println("Enter Teacher Address: ");
51 String teacherAddress=sc.next();
52 System.out.println("Enter Teacher Salary: ");
53 int teacherSalary=sc.nextInt();
54 stmt=con.prepareStatement("insert into Teacher values(?,?,?,?,?)");
55 stmt.setInt(1,teacher_Id);
56 stmt.setString(2,teacheName);
57 stmt.setString(3,teacherQualification);
58 stmt.setString(4,teacherAddress);
59 stmt.setInt(5,teacherSalary);
60 int i=stmt.executeUpdate();
61 if(i==1)
62 {
63 System.out.println("Record Inserted");
64 }
65 else
66 {
67 System.out.println("Record Not Inserted");
68 }
69 break;
70 case 2://Inserting into Student
71 System.out.println("Enter Student Id: ");
72 int student_ID=sc.nextInt();
73 System.out.println("Enter Student Name: ");
74 String student_Name=sc.next();
75 System.out.println("Enter Student Course: ");
76 String student_Course=sc.next();
Outline
school_management
School_Management_Data
sc : Scanner
main(String[]) : void
Writable
Smart Insert
242 : 1 : 6939
Type here to search
56%
ENG
8:01 PM
11/11/2022


eclipse-workspace - Maven_project/src/main/java/school_management/School_Management_Data.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
array-programming
Bank Project
Collection
FirstProgram [FirstProgram master]
JDBC
Maven_project
src/main/java
com.exception
com.testcase
com.utility
school_management
jdbc_connection.java
School_Management_Data.java
src/main/resources
src/test/java
src/test/resources
JRE System Library [J2SE-1.5]
Maven Dependencies
src
target
pom.xml
Oops
School Management
School Management System
Thread-Program
School_Management_Data.java
74 String student_Name=sc.next();
75 System.out.println("Enter Student Course: ");
76 String student_Course=sc.next();
77 System.out.println("Enter Student Fee: ");
78 int student_fee=sc.nextInt();
79 stmt=con.prepareStatement("insert into Student values(?,?,?,?)");
80 stmt.setInt(1,student_ID);
81 stmt.setString(2,student_Name);
82 stmt.setString(3,student_Course);
83 stmt.setInt(4,student_fee);
84 int j=stmt.executeUpdate();
85 if(j==1)
86 {
87 System.out.println("Record Inserted");
88 }
89 else
90 {
91 System.out.println("Record Not Inserted");
92 }
93 break;
94 case 3://Inserting into Parents
95 System.out.println("Enter Parents Id: ");
96 int parents_Id=sc.nextInt();
97 System.out.println("Enter Parents Name: ");
98 String parents_Name=sc.next();
99 System.out.println("Enter parents Address: ");
100 String parents_Address=sc.next();
101 System.out.println("Enter Student FeePaid: ");
102 int parents_FeePaid=sc.nextInt();
103 stmt=con.prepareStatement("insert into parents values(?,?,?,?)");
104 stmt.setInt(1,parents_Id);
105 stmt.setString(2,parents_Name);
106 stmt.setString(3,parents_Address);
107 stmt.setInt(4,parents_FeePaid);
108 int l=stmt.executeUpdate();
109 if(l==1)
Outline
school_management
School_Management_Data
sc : Scanner
main(String[]) : void
Writable
Smart Insert
242 : 1 : 6939
8:02 PM

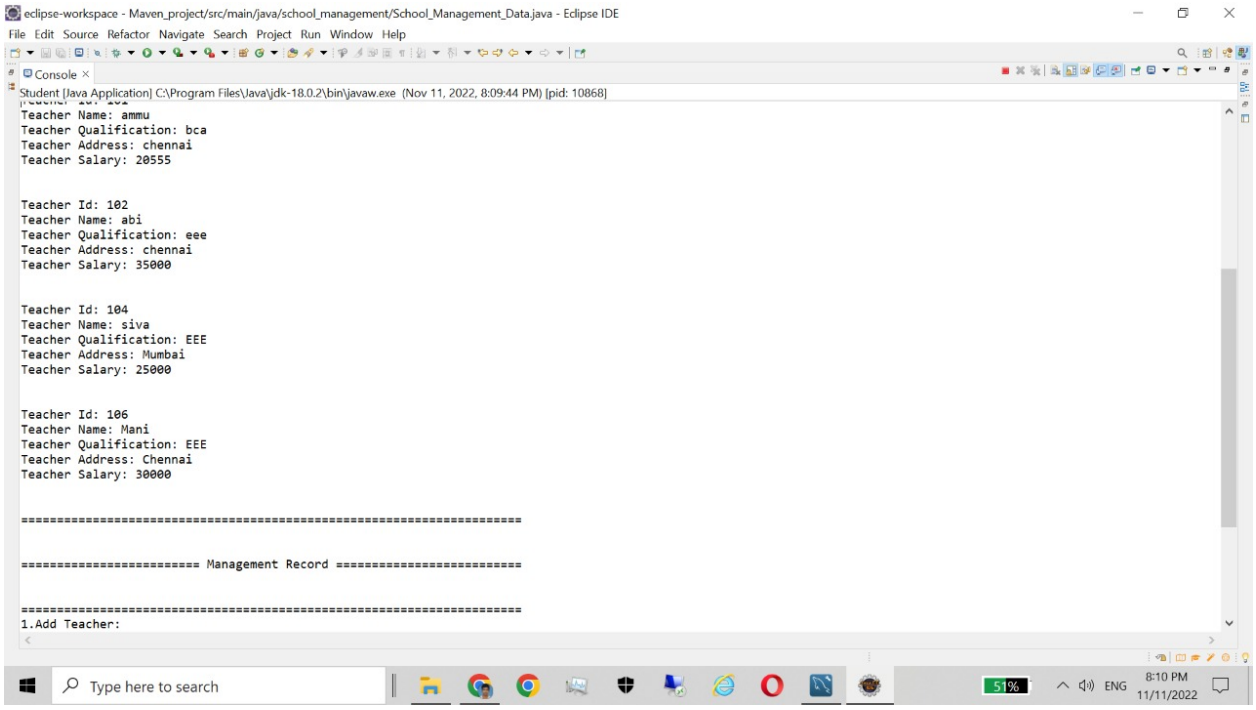

eclipse-workspace - Maven_project/src/main/java/school_management/School_Management_Data.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
array-programming
Bank Project
Collection
FirstProgram [FirstProgram master]
JDBC
Maven_project
src/main/java
com.exception
com.testcase
com.utility
school_management
jdbc_connection.java
School_Management_Data.java
src/main/resources
src/test/java
src/test/resources
JRE System Library [J2SE-1.5]
Maven Dependencies
src
target
pom.xml
Oops
School Management
School Management System
Thread-Program
School_Management_Data.java
stmt.setInt(4,parents_FeePaid);
int l=stmt.executeUpdate();
if(l==1)
{
System.out.println("Record Inserted");
}
else
{
System.out.println("Record Not Inserted");
}
break;
case 4://View Teacher
stmt=con.prepareStatement("select * from Teacher");
ResultSet rs=stmt.executeQuery();
while(rs.next())
{
System.out.println("Teacher Id: "+rs.getInt(1));
System.out.println("Teacher Name: "+rs.getString(2));
System.out.println("Teacher Qualification: "+rs.getString(3));
System.out.println("Teacher Address: "+rs.getString(4));
System.out.println("Teacher Salary: "+rs.getInt(5));
System.out.println("\n");
}
break;
case 5://View Student
stmt=con.prepareStatement("select * from Student");
ResultSet rs1=stmt.executeQuery();
while(rs1.next())
{
System.out.println("Student Id: "+rs1.getInt(1));
System.out.println("Student Name: "+rs1.getString(2));
System.out.println("Student Course: "+rs1.getString(3));
System.out.println("Student Fee: "+rs1.getInt(4));
System.out.println("\n");
}
break;
case 6://View parents
Outline
school_management
School_Management_Data
sc : Scanner
main(String[]) : void
Writable
Smart Insert
242 : 1 : 6939
Type here to search
55%
ENG
8:03 PM
11/11/2022

OUTPUT

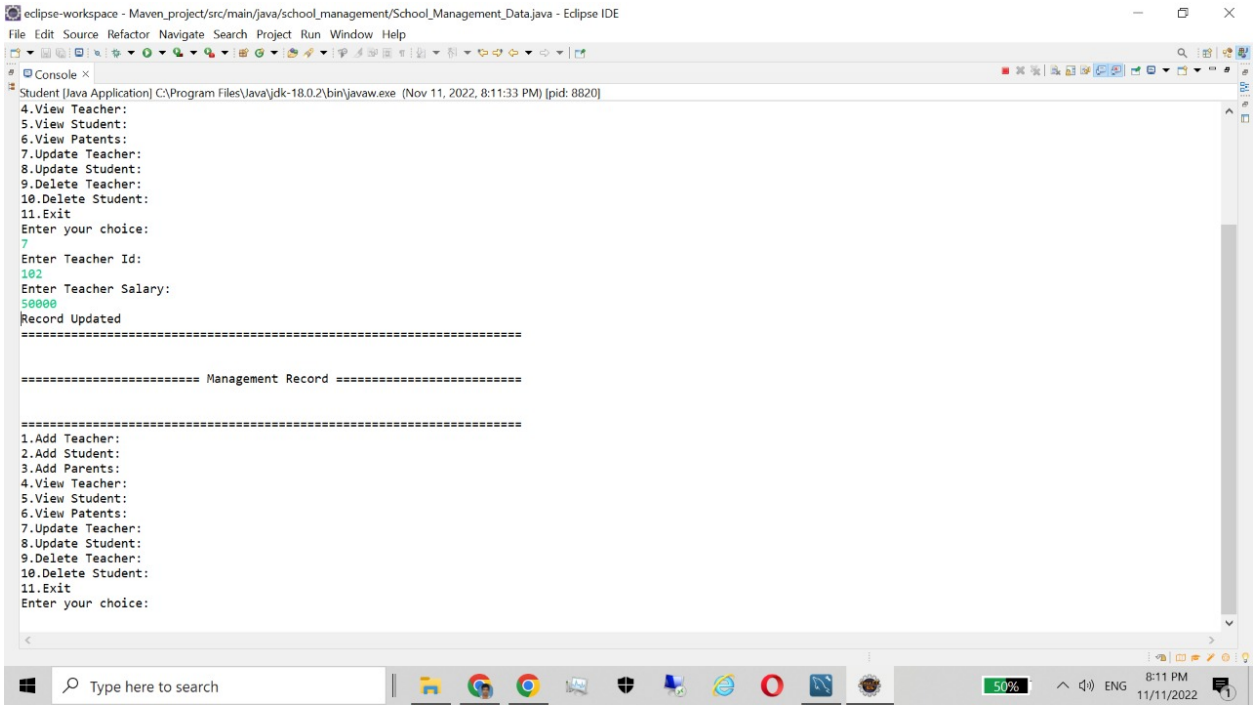
Output for Inserting



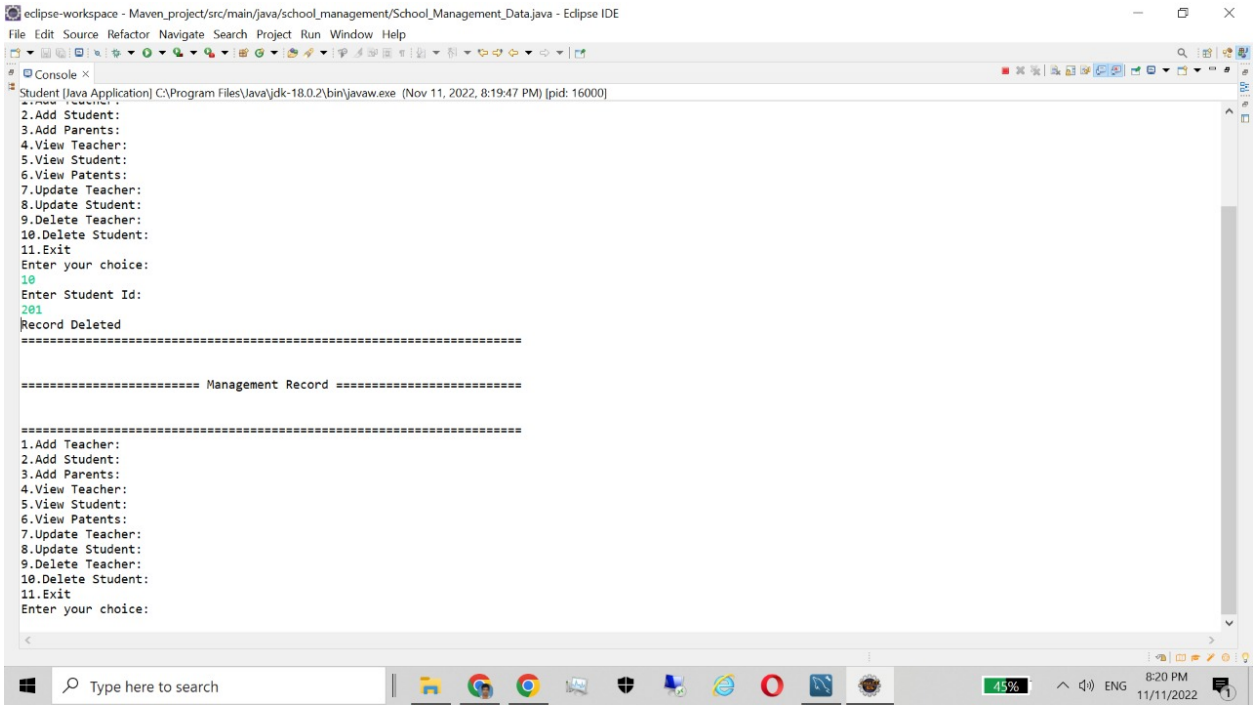
Output for View table



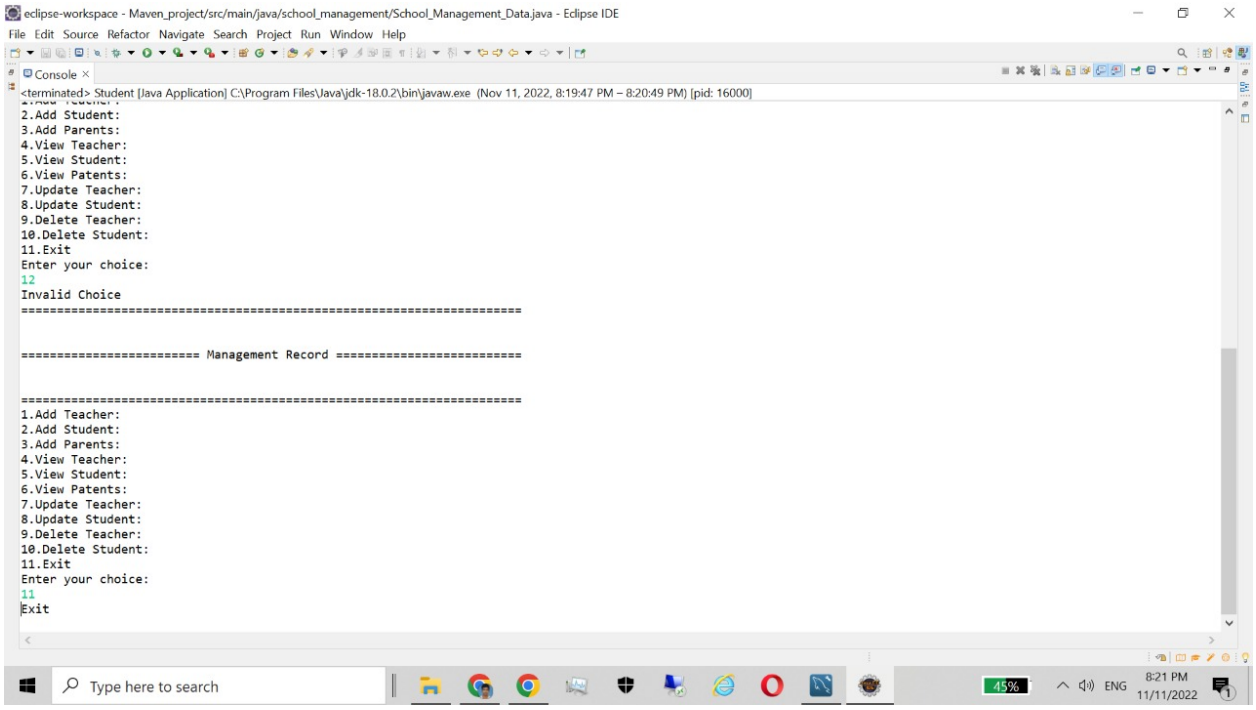
Output for Updating



Output for Deleting



Output for Exit



Conclusion:

In the offline system, it is an overhead to keep the records related to faculty, student, parents, and other school staff on the papers. Everything related to their progress in the system is marked manually. My other aim is to minimize the paperwork as minimum as I can so that there is no need to cut more and more trees.