

# EduBridge



A  
Project Report  
On  
“SCHOOL  
MANAGEMENT  
SYSTEM”

By  
**ABILASHA T**

**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**

# **SCHOOL MANAGEMENT SYSTEM**

The school management system is developed to help staff do less work, help students and teachers save time, cut costs, and improve data security. All of these things will eventually make an institution more cost-effective and productive. It is also able to secure important data of students and educators that every educational institution needs.

Data security and management are very important parts of the school management system. This makes sure that every student in the school is kept up to date. Educators' and students' names, parents' names, transaction history, and other achievements are all stored. Monitoring and keeping track of notes, remarks, incidents, and other actions such as internal documents and

workflows are included in the system. It is individuals who make and manage timetables. Keeping track of exams, assessments, grades, and progress in school is all managed by the software.

- There is a lot of paper used to keep track of student information, their results, and their performance.
- Manual systems are difficult to locate and update with information about students, their grades, and how well they performed in class.
- The manual system doesn't protect the academic information that could be lost if it were lost.
- There are some things that the school doesn't tell parents or teachers about, so they don't know about them.

## **PROJECT SCOPE**

This system is designed to be easy for people to use and efficient for a wide range of different tasks. These tasks may include things like registering new students, managing fees, managing exams, and all the other things that make the administrative department of a school work well.

- **School Information Management:** This module will show the information about a specific school whether it is primary, secondary, or tertiary. The school information includes the number of rooms and labs they have as well as the count of teachers and students.
- **Student Management:** The information of the student must also be managed in school management because the student information was the most important data that the system should process.
- **Grading System:** The grading system module will sum up and calculate the given output by the students and the result would be the academic performance of every student. This will monitor the student's performance and behaviour throughout the period.

## **EXISTING SYSTEM**

In the present School Management System, it is uneasy to store the information related to students, faculty and parents on the paper. As there is too much information when someone tries to access any of stored information

it becomes a difficult and time-consuming task. While these days parents and faculty have more work than just take care of students it is difficult for both parents and faculty to monitor them. Whereas the storing and retrieving an information is a difficult task, it also requires much amount of unnecessary worker to do the task.

## **PROPOSED SYSTEM**

The school management system will manage all the work in any school in particular order so that the time requirement and complexity of the system will be reduced, at first it will focus on student related information. As a student gets the admission in the school system will start managing the details regarding the students. It will manage the fee details, and if the full payment has not done, then it will notify about the fee to a staff of the school. School Management System will then display the date of the test and when the test completes it will display the results of the students. While the parents can use it to monitor their children's performance, also they can contact with the teachers.

# **SCHOOL MANAGEMENT SYSTEM MODULE**

## **HARDWARE REQUIREMENTS:**

- Processor : Intel Pentium 4
- Hard disk : 2GB
- Ram : 512 MB
- Operating System: Windows 10

## **SOFTWARE REQUIREMENTS:**

- Language used : Java
- Database: Mysql
- Tools used: Eclipse (IDE), Mysql Workbench 8.0

# CREATION OF TABLES IN A DATABASE

The screenshot displays the MySQL Workbench interface. On the left, the 'SCHEMAS' pane shows a tree view with 'school' selected, containing 'Tables', 'Views', 'Stored Procedures', and 'Functions'. The 'Tables' pane shows a table named 'student2' with columns 'id', 'name', 'city', and 'standard'. The 'Result Grid' pane shows the following data:

id	name	city	standard
1	Ablasha	Chennai	11
2	Revathi	Bangalore	11
4	Robert	Chennai	11
5	Sheela	Bangalore	11
6	Manik	Hyderabad	11
7	Akila	Hyderabad	11
8	Beena	Hyderabad	11
9	Moorthy	Hyderabad	11

The 'SQL File 1' pane shows the following SQL code:

```
28 (5,'Sheela','Bangalore',11),
29 (6,'Manik','Hyderabad',11),
30 (7,'Akila','Hyderabad',11),
31 (8,'Beena','Hyderabad',11),
32 (9,'Moorthy','Hyderabad',11);
33 select * from student2;
```

The 'Output' pane shows the following log:

#	Time	Action	Message	Duration / Fetch
6	17:11:53	use school	0 row(s) affected	0.000 sec
7	17:11:54	use school	0 row(s) affected	0.000 sec
8	17:11:57	use school	0 row(s) affected	0.000 sec
9	17:12:02	show databases	10 row(s) returned	0.000 sec / 0.000 sec
10	17:12:07	use school	0 row(s) affected	0.016 sec
11	17:12:14	select * from student2 LIMIT 0, 1000	13 row(s) returned	0.016 sec / 0.000 sec

## CREATION OF STUDENT TABLE

The screenshot displays the MySQL Workbench interface. The left sidebar shows the 'SCHEMAS' panel with a tree view containing 'customer', 'employees', 'sampledb', 'school', 'student2', and 'Administration'. The 'school' schema is selected, and the 'Tables' folder is expanded. The main editor window shows the SQL File 1 with the following SQL code:

```
28 (5, 'Sheela', 'Bangalore', 11),
29 (6, 'Manik', 'Hyderabad', 11),
30 (7, 'Akila', 'Hyderabad', 11),
31 (8, 'Beena', 'Hyderabad', 11),
32 (9, 'Moorthy', 'Hyderabad', 11);
33 select * from student2;
```

The 'Result Grid' shows the output of the query, displaying a table with 16 rows and 4 columns: id, name, city, and standard. The data is as follows:

id	name	city	standard
7	Akila	Hyderabad	11
8	Beena	Hyderabad	11
9	Moorthy	Hyderabad	11
0			
0			
10	Seetha	Hyderabad	11
15	Vivin	Bangalore	11
16	Ramsu...	Chennai	11

The bottom panel shows the 'Output' tab with the 'Action Output' table, which lists the actions performed and their results:

#	Time	Action	Message	Duration / Fetch
7	17:11:54	use school	0 row(s) affected	0.000 sec
8	17:11:57	use school	0 row(s) affected	0.000 sec
9	17:12:02	show databases	10 row(s) returned	0.000 sec / 0.000 sec
10	17:12:07	use school	0 row(s) affected	0.016 sec
11	17:12:14	select * from student2 LIMIT 0, 1000	13 row(s) returned	0.016 sec / 0.000 sec
12	17:12:47	select * from student2 LIMIT 0, 1000	12 row(s) returned	0.031 sec / 0.000 sec



# CREATION OF STUDENT ASSESSMENT TABLE

The screenshot displays the MySQL Workbench interface. The left sidebar shows the 'SCHEMAS' panel with a tree view containing 'customer', 'employees', 'sampledb', 'school', and 'Information'. The 'school' schema is selected. The main editor window shows a SQL script with the following content:

```
48 (5,'Sheela','11',88,100,'005','Pass'),
49 (6,'Manik','11',88,75,'006','Pass'),
50 (7,'Beena','11',66,22,'008','Pass'),
51 (8,'Beena','11',66,22,'008','Pass'),
52 (9,'Moorthy','11',20,17,'009','Fail');
53 select* from assessmentresult1;
```

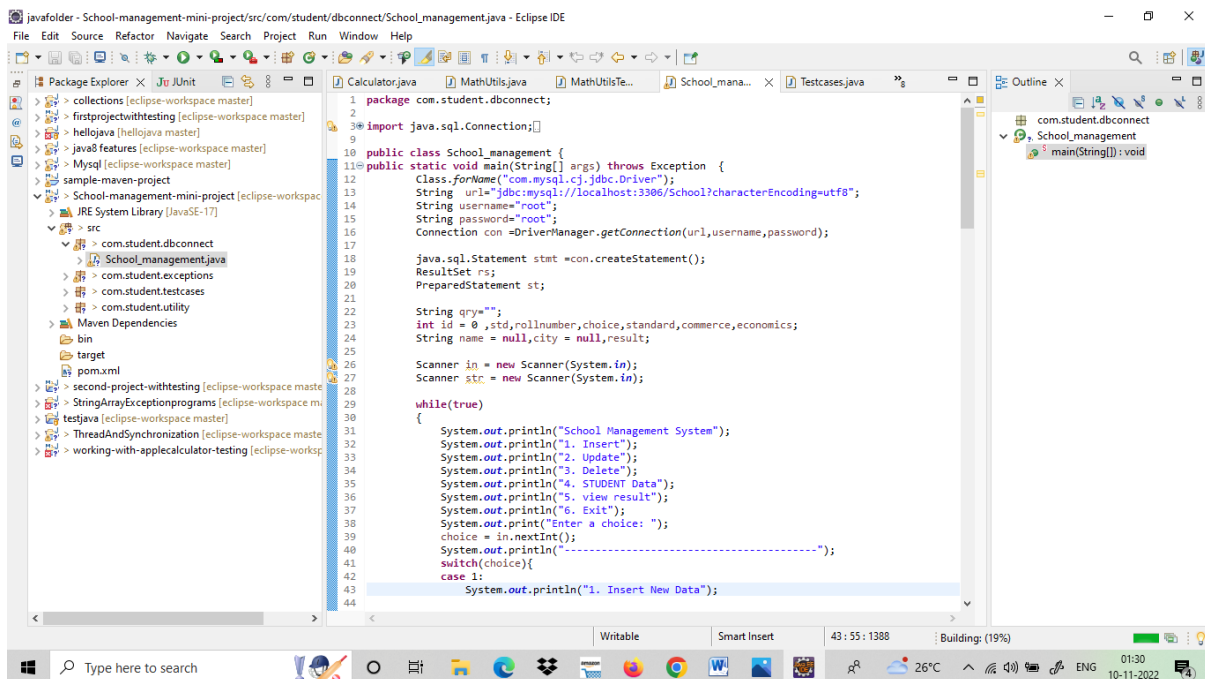
Below the SQL editor, the 'Result Grid' shows a table with the following data:

id	name	std	commerce	economics	rollnumber	Result
2	Revathi	11	99	98	002	Pass
3	Krishna	11	40	22	003	Fail
4	Robert	11	70	70	004	Pass
5	Sheela	11	80	100	005	Pass
6	Manik	11	80	75	006	Pass
7	Beena	11	66	22	008	Pass
8	Beena	11	66	22	008	Pass
9	Moorthy	11	20	17	009	Fail

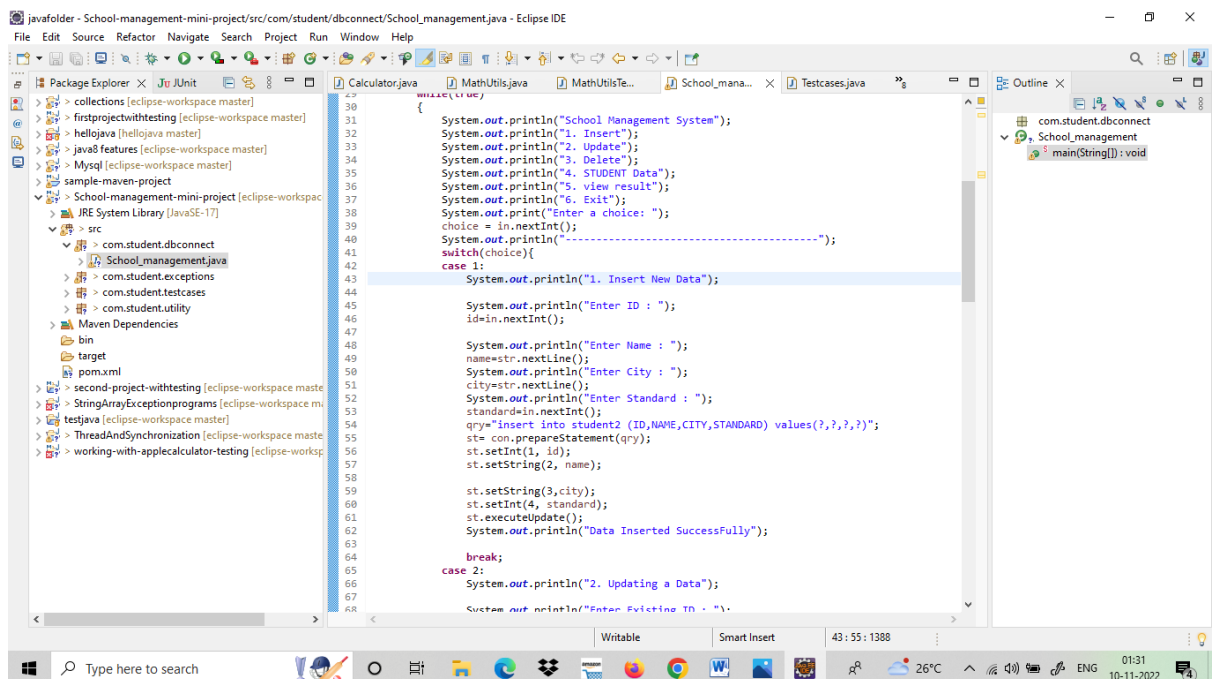
The 'Output' panel at the bottom shows the execution log:

#	Time	Action	Message	Duration / Fetch
1	01:12:53	use school	0 row(s) affected	0.000 sec
2	01:13:04	select* from AssessmentResult1	INSERT INTO teacher1(id,'name','subject')VALUES(1... Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds t...	0.062 sec
3	01:13:35	select * from student2 LIMIT 0, 1000	12 row(s) returned	0.328 sec / 0.000 sec
4	01:15:14	CREATE TABLE 'AssessmentResult1' (id int(255) NOT NULL, 'name' varchar(255) NOT NU...	Error Code: 1050. Table 'assessmentresult1' already exists	0.047 sec
5	01:16:15	select* from assessmentresult1 LIMIT 0, 1000	9 row(s) returned	0.031 sec / 0.000 sec

# CONNECTION OF ECLIPSE WITH MYSQL WORKBENCH USING JDBC CONNECTION

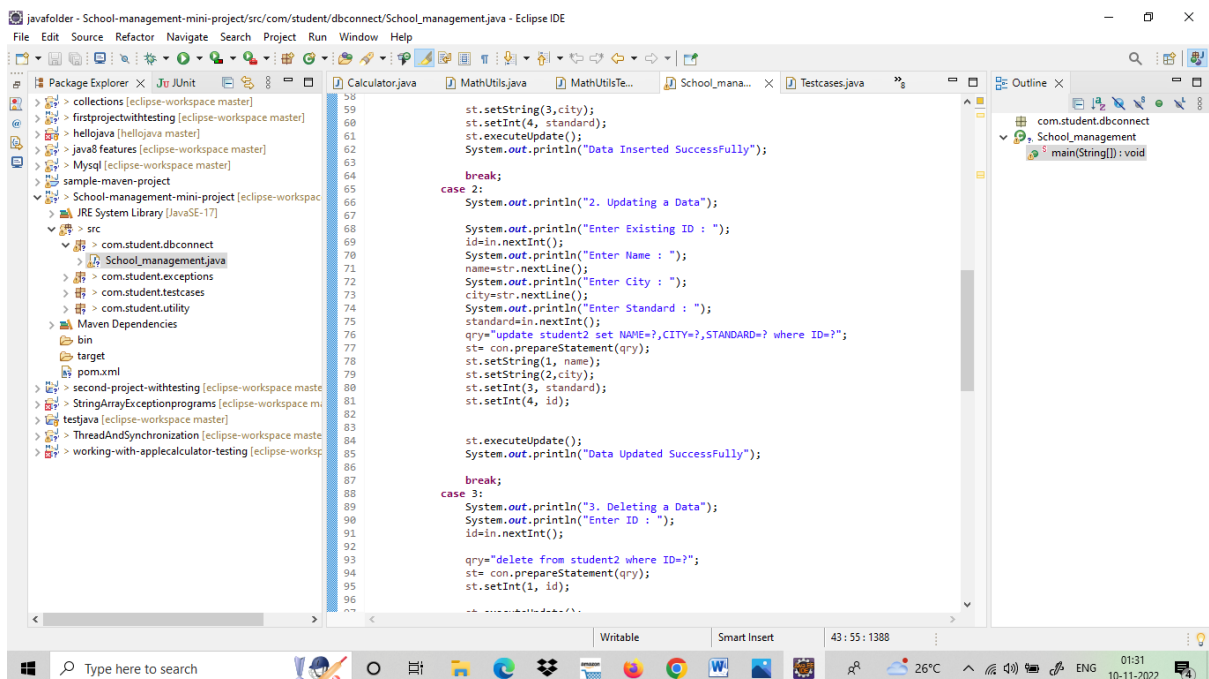


## INSERTING A NEW RECORD IN THE DATABASE-INSERTION



```
429
430
431 {
432     System.out.println("School Management System");
433     System.out.println("1. Insert");
434     System.out.println("2. Update");
435     System.out.println("3. Delete");
436     System.out.println("4. STUDENT Data");
437     System.out.println("5. view result");
438     System.out.println("6. Exit");
439     choice = in.nextInt();
440     System.out.println("-----");
441     switch(choice){
442     case 1:
443         System.out.println("1. Insert New Data");
444
445         System.out.println("Enter ID : ");
446         id=in.nextInt();
447
448         System.out.println("Enter Name : ");
449         name=nr.nextLine();
450         System.out.println("Enter City : ");
451         city=nr.nextLine();
452         System.out.println("Enter Standard : ");
453         standard=nr.nextInt();
454         qry="insert into student2 (ID,NAME,CITY,STANDARD) values(?,?,?,?)";
455         st= con.prepareStatement(qry);
456         st.setInt(1, id);
457         st.setString(2, name);
458
459         st.setString(3,city);
460         st.setInt(4, standard);
461         st.executeUpdate();
462         System.out.println("Data Inserted Successfully");
463
464         break;
465     case 2:
466         System.out.println("2. Updating a Data");
467         System.out.println("Enter Existing ID : ");
```

## UPDATING A RECORD IN THE DATABASE-UPDATION



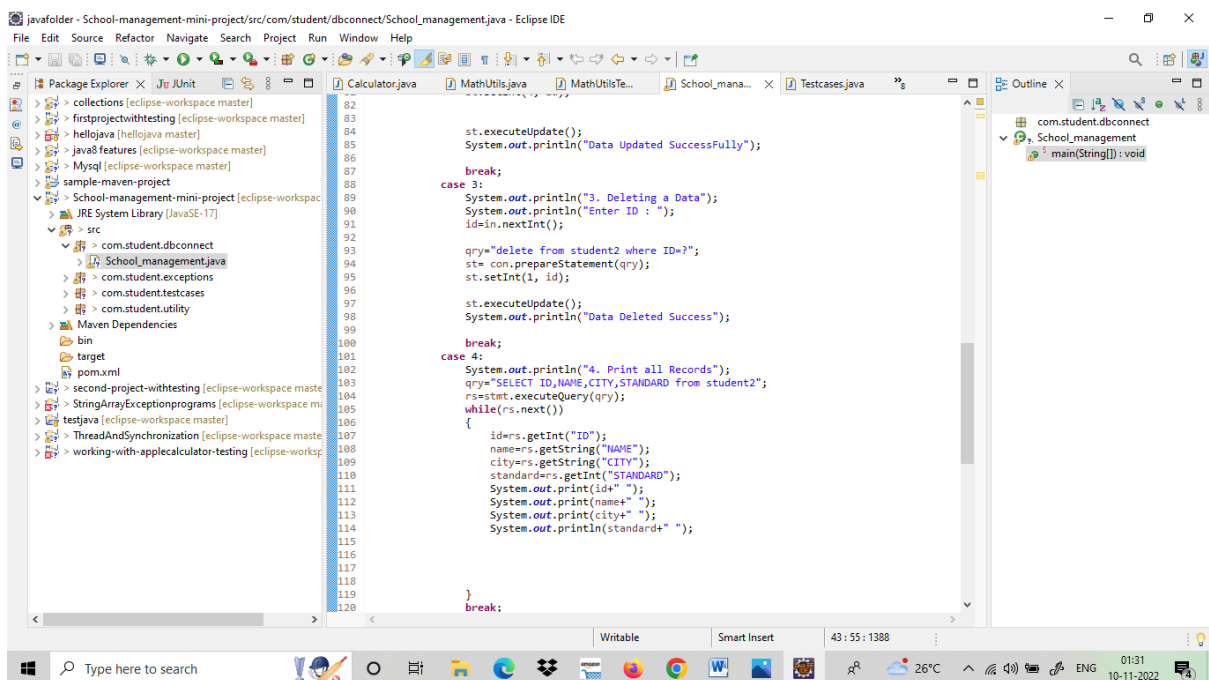
## DELETING A RECORD FROM THE DATABASE -DELETION

```
82  
83  
84  
85  
86  
87  
88  
89  
90  
91  
92  
93  
94  
95  
96  
97  
98  
99  
100  
101  
102  
103  
104  
105  
106  
107  
108  
109  
110  
111  
112  
113  
114  
115  
116  
117  
118  
119  
120  
break;
```

```
st.executeUpdate();  
System.out.println("Data Updated Successfully");  
  
break;  
case 3:  
    System.out.println("3. Deleting a Data");  
    System.out.println("Enter ID : ");  
    id=in.nextInt();  
  
    qry="delete from student2 where ID=?";  
    st= con.prepareStatement(qry);  
    st.setInt(1, id);  
  
    st.executeUpdate();  
    System.out.println("Data Deleted Success");  
  
    break;  
case 4:  
    System.out.println("4. Print all Records");  
    qry="SELECT ID,NAME,CITY,STANDARD from student2";  
    rs=stmt.executeQuery(qry);  
    while(rs.next())  
    {  
        id=rs.getInt("ID");  
        name=rs.getString("NAME");  
        city=rs.getString("CITY");  
        standard=rs.getInt("STANDARD");  
        System.out.print(id+" ");  
        System.out.print(name+" ");  
        System.out.print(city+" ");  
        System.out.println(standard+" ");  
    }  
}
```

com.student.dbconnect  
School\_management  
main(String[]): void

## CODE FOR STUDENT

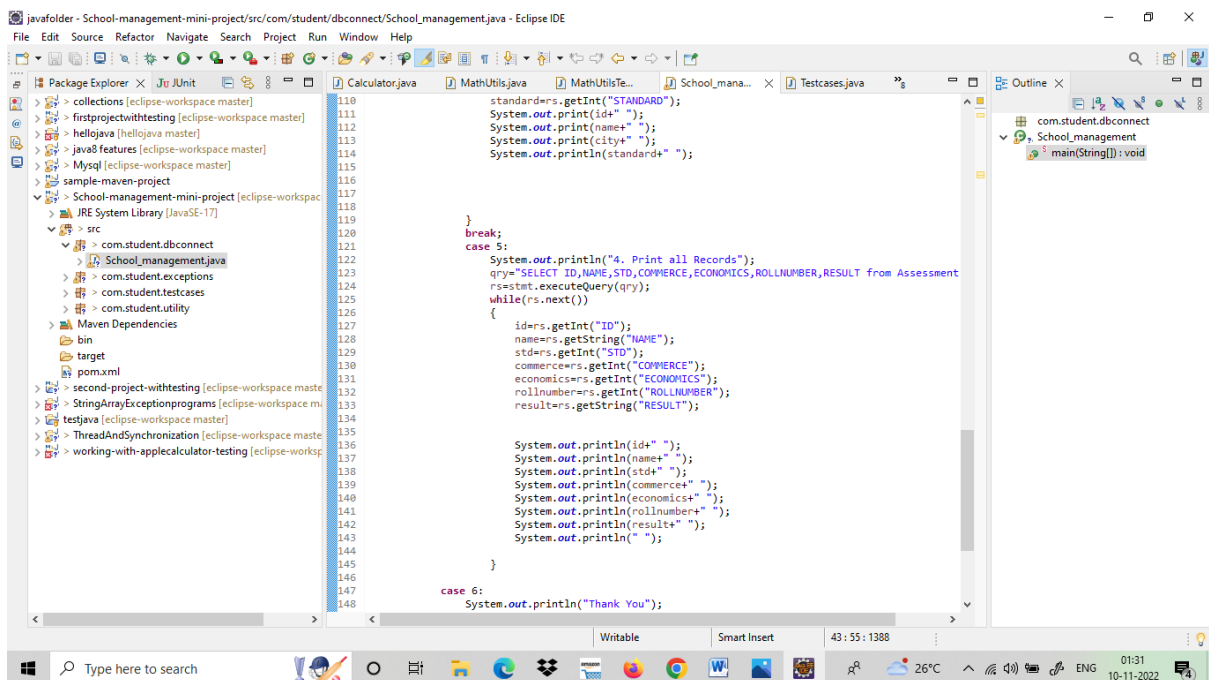


```
82  
83  
84  
85  
86  
87  
88  
89  
90  
91  
92  
93  
94  
95  
96  
97  
98  
99  
100  
101  
102  
103  
104  
105  
106  
107  
108  
109  
110  
111  
112  
113  
114  
115  
116  
117  
118  
119  
120  
break;
```

```
st.executeUpdate();  
System.out.println("Data Updated Successfully");  
  
break;  
case 3:  
    System.out.println("3. Deleting a Data");  
    System.out.println("Enter ID : ");  
    id=in.nextInt();  
  
    qry="delete from student2 where ID=?";  
    st= con.prepareStatement(qry);  
    st.setInt(1, id);  
  
    st.executeUpdate();  
    System.out.println("Data Deleted Success");  
  
    break;  
case 4:  
    System.out.println("4. Print all Records");  
    qry="SELECT ID,NAME,CITY,STANDARD from student2";  
    rs=stmt.executeQuery(qry);  
    while(rs.next())  
    {  
        id=rs.getInt("ID");  
        name=rs.getString("NAME");  
        city=rs.getString("CITY");  
        standard=rs.getInt("STANDARD");  
        System.out.print(id+" ");  
        System.out.print(name+" ");  
        System.out.print(city+" ");  
        System.out.println(standard+" ");  
    }  
}
```

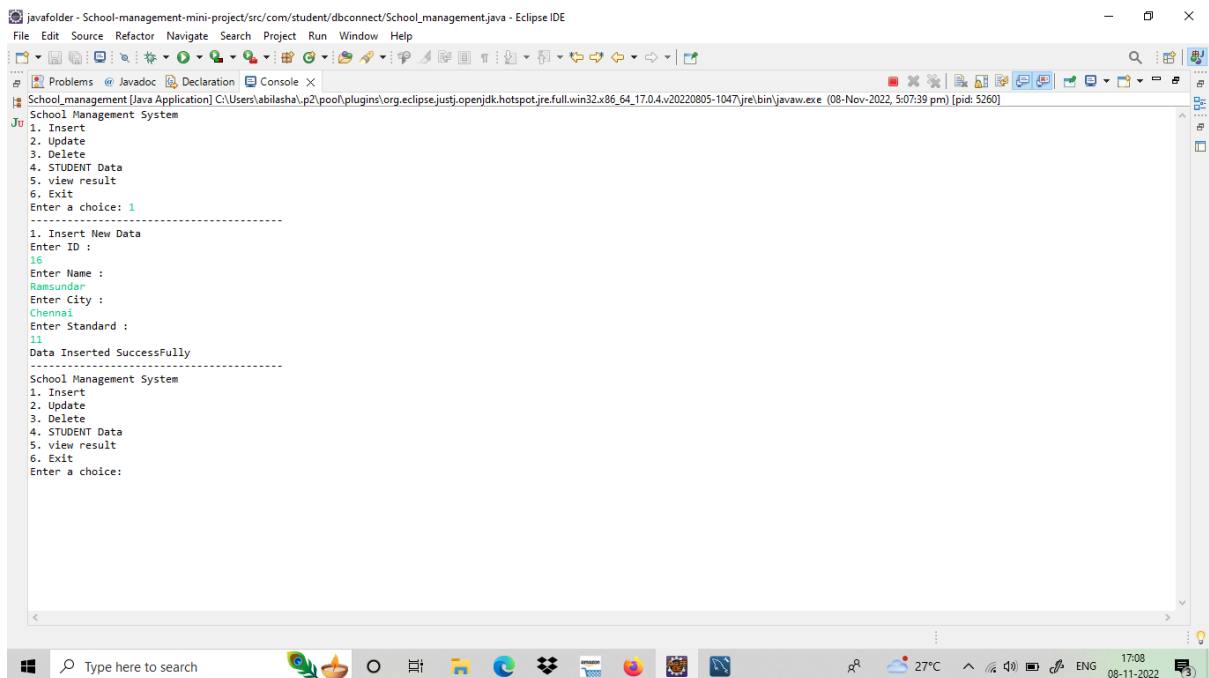
com.student.dbconnect  
School\_management  
main(String[] args): void

# CODE FOR STUDENT ASSESSMENT



```
110         standard=rs.getInt("STANDARD");
111         System.out.print(id+" ");
112         System.out.print(name+" ");
113         System.out.print(city+" ");
114         System.out.println(standard+" ");
115
116
117
118
119     }
120     break;
121     case 5:
122         System.out.println("4. Print all Records");
123         qry="SELECT ID,NAME,STD,COMMERCE,ECONOMICS,ROLLNUMBER,RESULT from Assessment";
124         rs=stmt.executeQuery(qry);
125         while(rs.next())
126         {
127             id=rs.getInt("ID");
128             name=rs.getString("NAME");
129             std=rs.getInt("STD");
130             commerce=rs.getInt("COMMERCE");
131             economics=rs.getInt("ECONOMICS");
132             rollnumber=rs.getInt("ROLLNUMBER");
133             result=rs.getString("RESULT");
134
135             System.out.println(id+" ");
136             System.out.println(name+" ");
137             System.out.println(std+" ");
138             System.out.println(commerce+" ");
139             System.out.println(economics+" ");
140             System.out.println(rollnumber+" ");
141             System.out.println(result+" ");
142             System.out.println(" ");
143         }
144
145     case 6:
146         System.out.println("Thank You");
147
148
```

## OUTPUT FOR DATA INSERTION



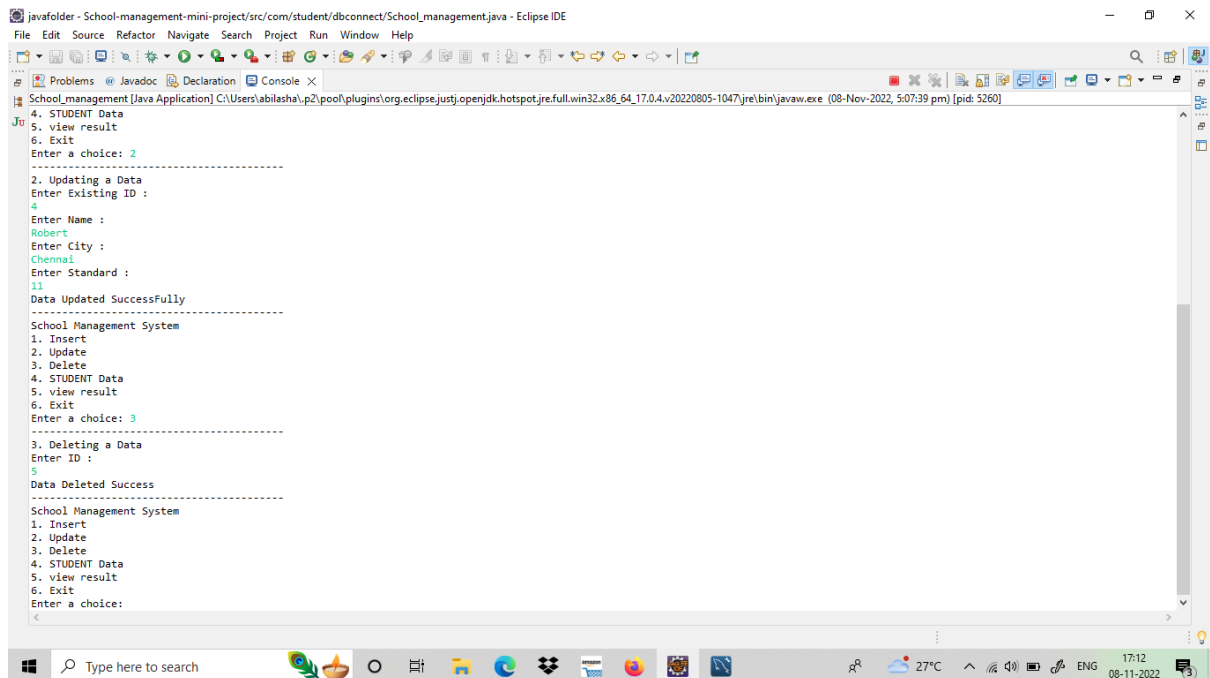
```
javafolder - School-management-mini-project/src/com/student/dbconnect/School_management.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help

School_management [Java Application] C:\Users\abilasha\p2\poo\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.4.v20220805-1047\jre\bin\javaw.exe (08-Nov-2022, 5:07:39 pm) [pid: 5260]

School Management System
1. Insert
2. Update
3. Delete
4. STUDENT Data
5. view result
6. Exit
Enter a choice: 1
-----
1. Insert New Data
Enter ID :
16
Enter Name :
Ramsundar
Enter City :
Chennai
Enter Standard :
11
Data Inserted Successfully
-----
School Management System
1. Insert
2. Update
3. Delete
4. STUDENT Data
5. view result
6. Exit
Enter a choice:
```



## OUTPUT FOR DATA UPDATION

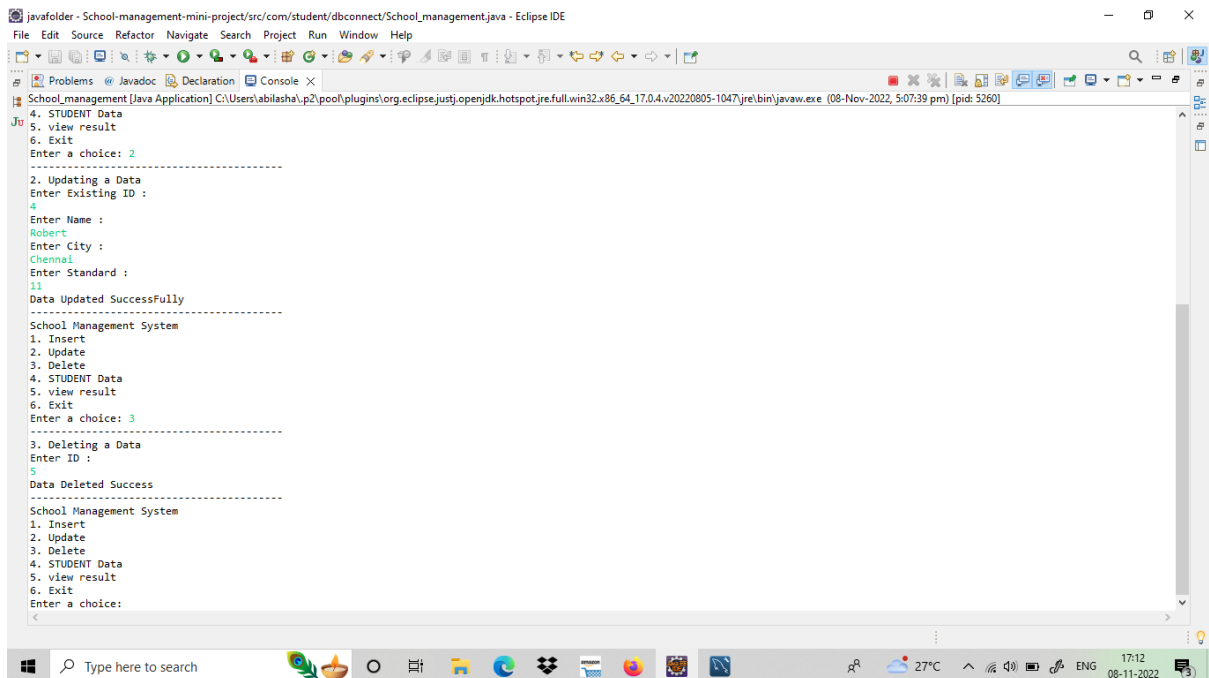


```
java folder - School-management-mini-project/src/com/student/dbconnect/School_management.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help

School_management [Java Application] C:\Users\abilasha\p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64_17.0.4.v20220805-1047\jre\bin\javaw.exe (08-Nov-2022, 5:07:39 pm) [pid: 5260]

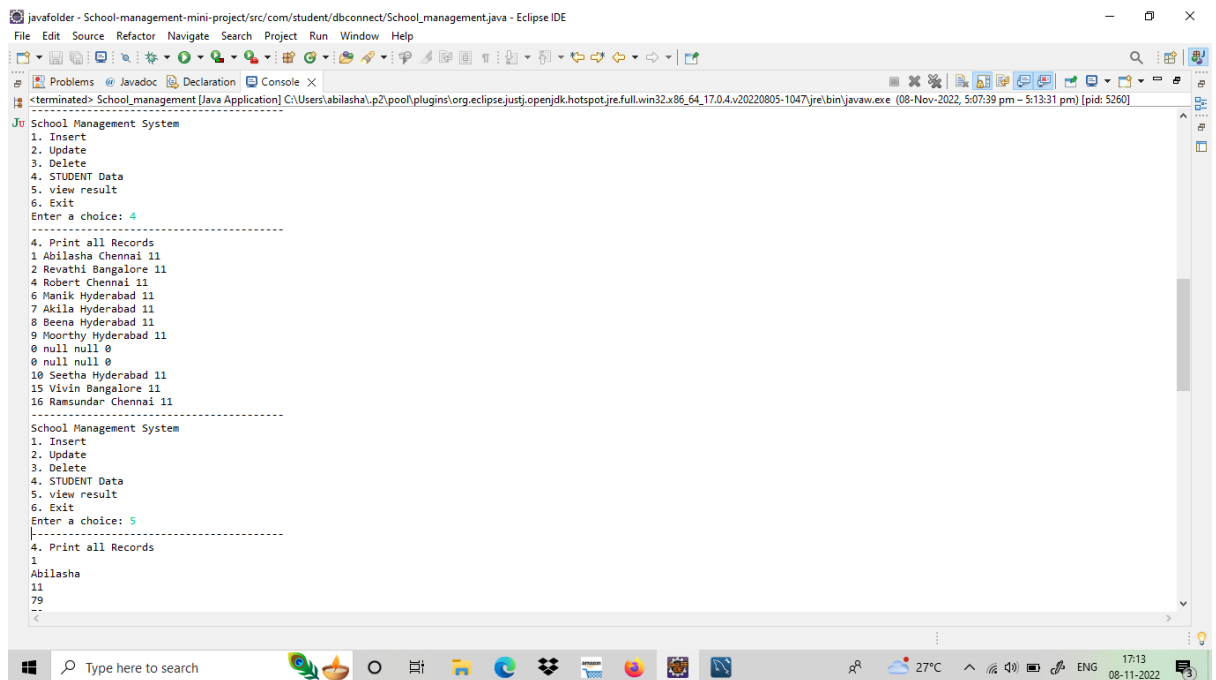
4. STUDENT Data
5. view result
6. Exit
Enter a choice: 2
-----
2. Updating a Data
Enter Existing ID :
4
Enter Name :
Robert
Enter City :
Chennai
Enter Standard :
11
Data Updated Successfully
-----
School Management System
1. Insert
2. Update
3. Delete
4. STUDENT Data
5. view result
6. Exit
Enter a choice: 3
-----
3. Deleting a Data
Enter ID :
5
Data Deleted Success
-----
School Management System
1. Insert
2. Update
3. Delete
4. STUDENT Data
5. view result
6. Exit
Enter a choice:
```

## OUTPUT FOR DATA DELETION



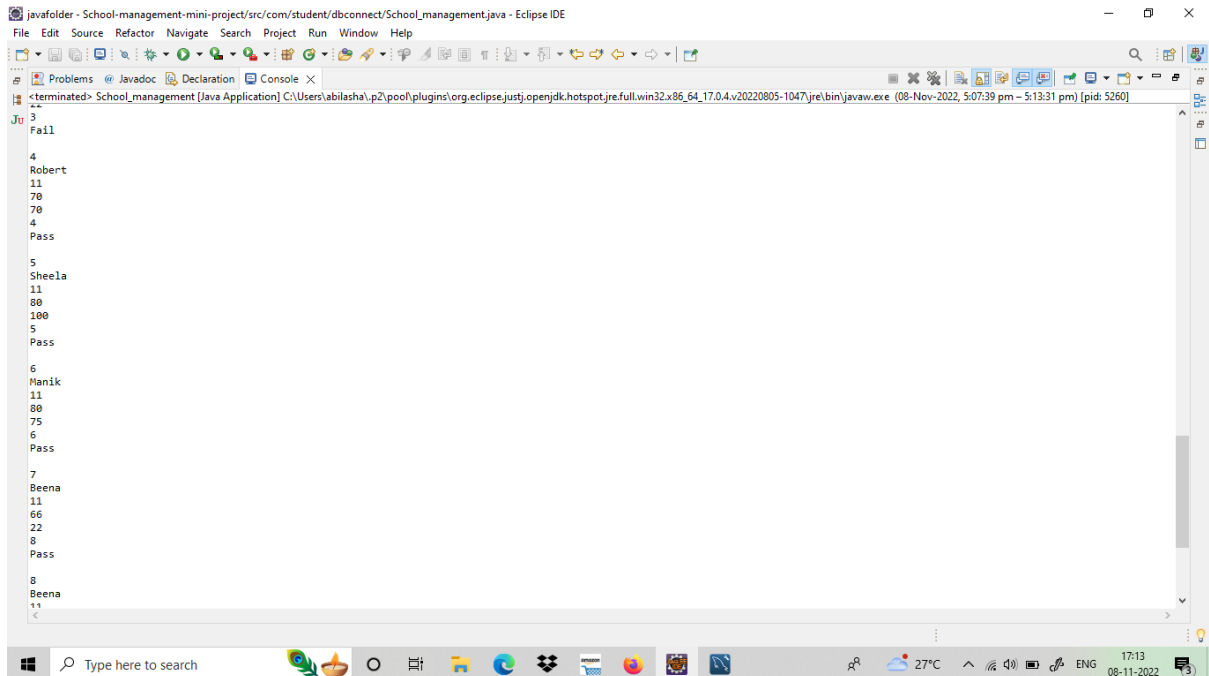
```
javafolder - School-management-mini-project/src/com/student/dbconnect/School_management.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
School_management [Java Application] C:\Users\abilasha\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.4.v20220805-1047\jre\bin\javaw.exe (08-Nov-2022, 5:07:39 pm) [pid: 5260]
4. STUDENT Data
5. view result
6. Exit
Enter a choice: 2
-----
2. Updating a Data
Enter Existing ID :
4
Enter Name :
Robert
Enter City :
Chennai
Enter Standard :
11
Data Updated Successfully
-----
School Management System
1. Insert
2. Update
3. Delete
4. STUDENT Data
5. view result
6. Exit
Enter a choice: 3
-----
3. Deleting a Data
Enter ID :
5
Data Deleted Success
-----
School Management System
1. Insert
2. Update
3. Delete
4. STUDENT Data
5. view result
6. Exit
Enter a choice:
```

## OUTPUT FOR STUDENT DATA



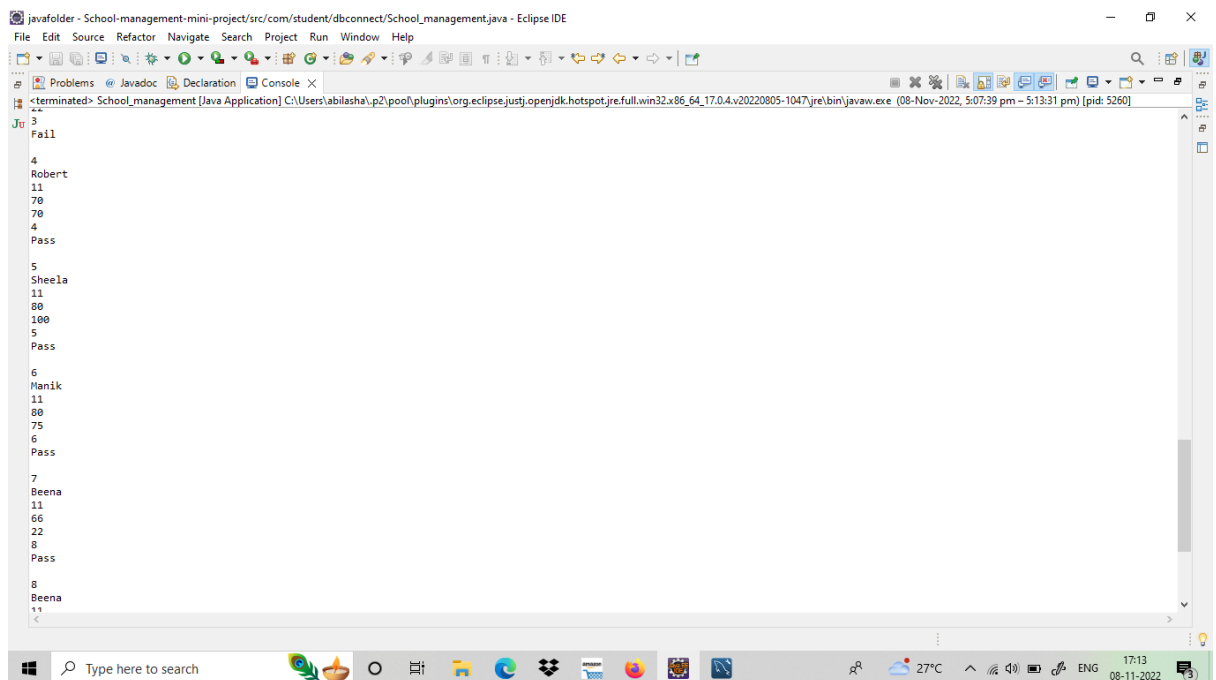
```
javafolder - School-management-mini-project/src/com/student/dbconnect/School_management.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
<terminated> School_management [Java Application] C:\Users\abilasha\p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64.17.0.4.v20220805-1047\jre\bin\javaw.exe (08-Nov-2022, 5:07:39 pm - 5:13:31 pm) [pid: 5260]
School Management System
1. Insert
2. Update
3. Delete
4. STUDENT Data
5. view result
6. Exit
Enter a choice: 4
-----
4. Print all Records
1 Abilasha Chennai 11
2 Revathi Bangalore 11
4 Robert Chennai 11
6 Manik Hyderabad 11
7 Akila Hyderabad 11
8 Beena Hyderabad 11
9 Moorthy Hyderabad 11
0 null null 0
0 null null 0
10 Seetha Hyderabad 11
15 Vivin Bangalore 11
16 Ramsundar Chennai 11
-----
School Management System
1. Insert
2. Update
3. Delete
4. STUDENT Data
5. view result
6. Exit
Enter a choice: 5
-----
4. Print all Records
1
Abilasha
11
79
```

## OUTPUT FOR STUDENT ASSESSMENT DATA



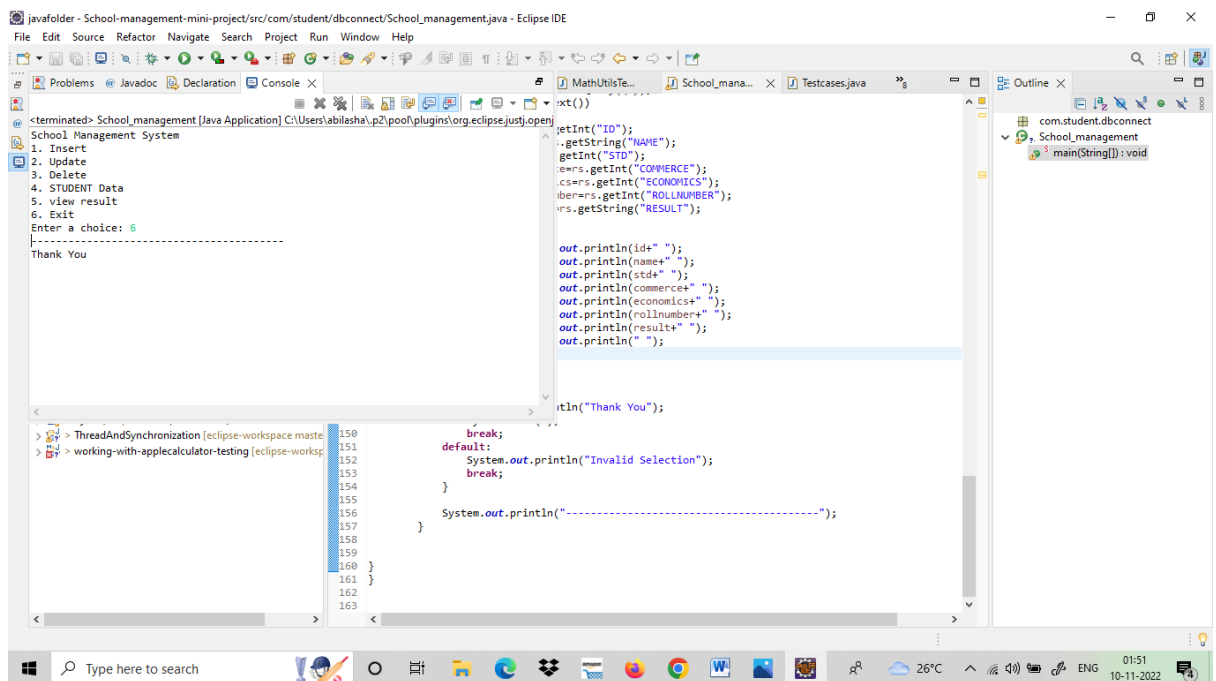
```
<terminated> School_management [Java Application] C:\Users\abilasha\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64.17.0.4.v20220805-1047\jre\bin\javaw.exe (08-Nov-2022, 5:07:39 pm - 5:13:31 pm) [pid: 5260]
3
Fall
4
4
Robert
11
70
70
4
Pass
5
Sheela
11
80
100
5
Pass
6
Manik
11
80
75
6
Pass
7
Beena
11
66
22
8
Pass
8
Beena
11
<
```

## OUTPUT FOR VIEWING THE RESULT



```
java folder - School-management-mini-project/src/com/student/dbconnect/School_management.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
<terminated> School_management [Java Application] C:\Users\abilasha\p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64.17.0.4.v20220805-1047\jre\bin\javaw.exe (08-Nov-2022, 5:07:39 pm - 5:13:31 pm) [pid: 5260]
Jg 3
Fail
4
Robert
11
70
70
4
Pass
5
Sheela
11
80
100
5
Pass
6
Manik
11
80
75
6
Pass
7
Beena
11
66
22
8
Pass
8
Beena
11
```

## OUTPUT FOR EXIT OPERATION



The screenshot displays the Eclipse IDE interface. The top toolbar includes icons for File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, and Help. The menu bar lists File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, and Help. The toolbar also contains icons for Problems, Javadoc, Declaration, Console, and a search icon. The main editor area shows the file `School_management.java` with the following code:

```
getInt("ID");
getString("NAME");
getInt("STD");
e=rs.getInt("COMMERCE");
cs=rs.getInt("ECONOMICS");
rber=rs.getInt("ROLLNUMBER");
rs.getString("RESULT");

out.println(id+" ");
out.println(name+" ");
out.println(std+" ");
out.println(commerce+" ");
out.println(economics+" ");
out.println(rollnumber+" ");
out.println(result+" ");
out.println(" ");

println("Thank You");
```

The Console view at the bottom shows the output of the application:

```
<terminated> School_management [Java Application] C:\Users\abilasha\p2\pool\plugins\org.eclipse.justi.open
School Management System
1. Insert
2. Update
3. Delete
4. STUDENT Data
5. view result
6. Exit
Enter a choice: 6
-----
Thank You
```

The bottom status bar shows the Windows taskbar with the search bar, system tray, and date/time (10-11-2022, 01:51).