

EduBridge



A Project Report

On

**“SCHOOL
MANAGEMENT
SYSTEM”**

By

BHARANIDHARAN B

Batch: 2022 – 7671

Center: Chennai chrompet

Under the Guidance of,

Chittaranjan Ghosh.

Technical Trainer

EduBridge

ABSTRACT

The school management system is a system which will use as a platform for interaction between student, teachers and parents. While the main objective of this project is to computerize the paperwork in the system and automate the work. The computerization is done so that the storage of all the details regarding students and teachers will be stored in the system which makes system centralized and the chance of duplication of any data is minimized. While by doing automation to the system will reduce the time for storing any data in the 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.

Software Requirements

- JVM(Eclipse IDE)
- MYSQL Workbench 8.0

Hardware Requirements

- Hard Disk – 2 GB.
- RAM required – 1 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

- From where should we start
- Where we have to go
- Where should we stop

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

The screenshot displays the MySQL Workbench interface for a local instance of MySQL 8.0. The left sidebar contains the 'MANAGEMENT' and 'INSTANCE' sections. The 'MANAGEMENT' section includes links to Server Status, Client Connections, Users and Privileges, Status and System Variables, Data Export, and Data Import/Restore. The 'INSTANCE' section includes links to Startup / Shutdown, Server Logs, Administration, and Schemas. The 'Schemas' section is currently selected, showing a list of databases: 'No object selected'.

The main window shows the 'Query 1' editor with the following SQL script:

```
1 • Show databases;
2 • use School;
3 • show tables;
4 • drop database School;
5 • create database School;
6
7 • CREATE TABLE `feesubmit` (
8   `id` int(255) NOT NULL,
9   `name` varchar(255) NOT NULL,
```

The 'Output' tab at the bottom shows the execution results of the queries. The first query, 'CREATE TABLE `reportcard`', was executed successfully. The second query, 'INSERT INTO `reportcard`', was also executed successfully. The third query, 'select * from reportcard LIMIT 0, 1000', returned 9 rows. The fourth query, 'INSERT INTO `Teacher`', was executed successfully. The fifth query, 'select * from Teacher LIMIT 0, 1000', returned 5 rows.

#	Time	Action	Message	Duration / Fetch
3	11:19:49	CREATE TABLE `reportcard` (`id` int(255) NOT NULL, `name` varchar(255) NOT NULL, ...	0 row(s) affected, 1 warning(s): 1681 Integer display width is deprecated and will be removed ...	0.250 sec
4	11:19:57	INSERT INTO `reportcard` (`id`, `name`, `std`, `rollnumber`, `Result`) VALUES (1, `Bharani`, `X...	9 row(s) affected Records: 9 Duplicates: 0 Warnings: 0	0.110 sec
5	11:20:05	select * from reportcard LIMIT 0, 1000	9 row(s) returned	0.000 sec / 0.000 sec
6	11:23:08	INSERT INTO `Teacher` (`Name`, `subject`, `City`, `Salary`) VALUES (Siva, physics, chennai, 54...	Error Code: 1054. Unknown column 'Siva' in field list	0.000 sec
7	11:25:02	INSERT INTO `Teacher` (`Name`, `subject`, `City`, `Salary`) VALUES (Siva, physics, chennai, ...	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0	0.079 sec
8	11:25:05	Select * from Teacher LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec

Creating Fee submit Table

The screenshot displays the MySQL Workbench interface. The left sidebar shows the 'MANAGEMENT' and 'INSTANCE' sections. The main query editor contains the following SQL code:

```
8 `id` int(255) NOT NULL,
9 `name` varchar(255) NOT NULL,
10 `age` int(255) NOT NULL,
11 `joiningMonth` varchar(255) NOT NULL,
12 `annualFee` int(255) NOT NULL,
13 `Status` varchar(255) NOT NULL
14 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
15 • drop table feesubmit;
16
```

The 'Result Grid' shows a table with the following data:

Name	Subject	City	Salary
Siva	physics	chennai	54000
Ramiya	maths	chennai	80000
Satish	computer	chennai	56000
Kumaran	Chemistry	chennai	67000
Priva	english	chennai	45000

The 'Output' section shows the execution of the following queries:

- 3 11:19:49 CREATE TABLE 'reportcard' (`id` int(255) NOT NULL, `name` varchar(255) NOT NULL, ... 0 row(s) affected, 1 warning(s): 1681 Integer display width is deprecated and will be removed ... 0.250 sec
- 4 11:19:57 INSERT INTO 'reportcard' ('id', 'name', 'std', 'rollnumber', 'Result') VALUES (1, 'Bharani', 'X... 9 row(s) affected Records: 9 Duplicates: 0 Warnings: 0 0.110 sec
- 5 11:20:05 select * from reportcard LIMIT 0, 1000 9 row(s) returned 0.000 sec / 0.000 sec
- 6 11:23:08 INSERT INTO 'Teacher' ('Name','subject','City','Salary') VALUES (Siva,physics,chennai,54... Error Code: 1054 Unknown column 'Siva' in field list' 0.000 sec
- 7 11:25:02 INSERT INTO 'Teacher' ('Name','subject','City','Salary') VALUES (Siva,'physics','chennai'... 5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0 0.079 sec
- 8 11:25:05 Select * from Teacher LIMIT 0, 1000 5 row(s) returned 0.000 sec / 0.000 sec

Creating Report card Table

The screenshot displays the MySQL Workbench interface. The left sidebar shows the 'MANAGEMENT' and 'INSTANCE' sections. The main editor window shows a SQL query to create a table named 'reportcard' with the following schema:

```
CREATE TABLE `reportcard` (  
  `id` int(255) NOT NULL,  
  `name` varchar(255) NOT NULL,  
  `std` varchar(255) NOT NULL,  
  `rollnumber` varchar(255) NOT NULL,  
  `Result` varchar(255) NOT NULL,  
  ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;  
)
```

The 'Result Grid' shows the following data:

Name	Subject	City	Salary
Siva	physics	chennai	54000
Ramya	maths	chennai	80000
Satish	computer	chennai	56000
Kumaran	Chemistry	chennai	67000
Priya	english	chennai	45000

The 'Output' window shows the execution of the following queries:

```
3 11:19:49 CREATE TABLE `reportcard` ( `id` int(255) NOT NULL, `name` varchar(255) NOT NULL, ... 0 row(s) affected, 1 warning(s): 1681 Integer display width is deprecated and will be removed ... 0.250 sec  
4 11:19:57 INSERT INTO `reportcard` (`id`, `name`, `std`, `rollnumber`, `Result`) VALUES (1, 'Bharani', 'X...', 9 row(s) affected Records: 9 Duplicates: 0 Warnings: 0 0.110 sec  
5 11:20:05 select * from reportcard LIMIT 0, 1000 9 row(s) returned 0.000 sec / 0.000 sec  
6 11:23:08 INSERT INTO `Teacher` (`Name`,`subject`,`City`,`Salary`) VALUES (Siva,physics,chennai,54... Error Code: 1054: Unknown column 'Siva' in field list 0.000 sec  
7 11:25:02 INSERT INTO `Teacher` (`Name`,`subject`,`City`,`Salary`) VALUES (Siva,'physics','chennai'... 5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0 0.079 sec  
8 11:25:05 Select * from Teacher LIMIT 0, 1000 5 row(s) returned 0.000 sec / 0.000 sec
```

Creating Teacher Table

The screenshot displays the MySQL Workbench interface. The left sidebar shows the 'MANAGEMENT' tree with 'Schemas' selected. The main editor window shows a SQL query to create a table named 'Teacher' with the following structure:

```
CREATE TABLE `Teacher` (  
  `Name` varchar(200) NOT NULL,  
  `Subject` varchar(200) NOT NULL,  
  `City` varchar(200) NOT NULL,  
  `Salary` int(255) NOT NULL,  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;  
  
Select * from Teacher;
```

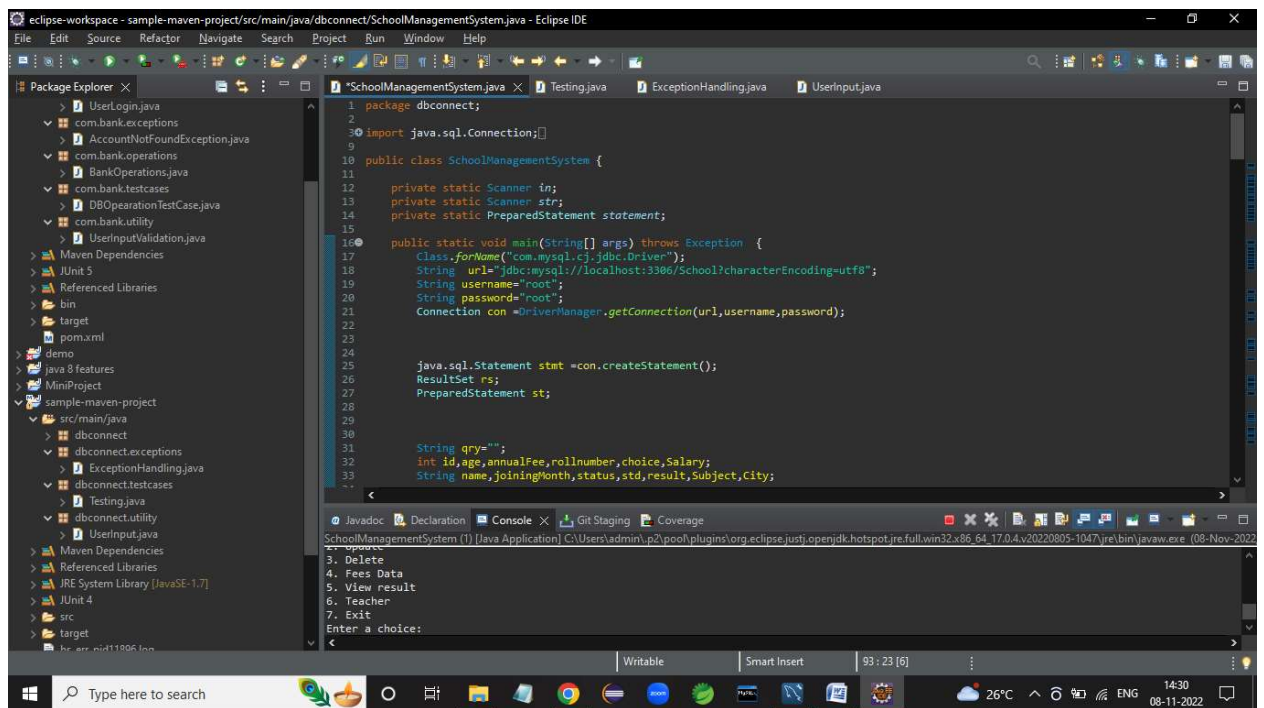
The 'Result Grid' shows the data returned by the query:

Name	Subject	City	Salary
Siva	physics	chennai	54000
Ramya	maths	chennai	80000
Satish	computer	chennai	56000
Kumaran	Chemistry	chennai	67000
Priya	english	chennai	45000

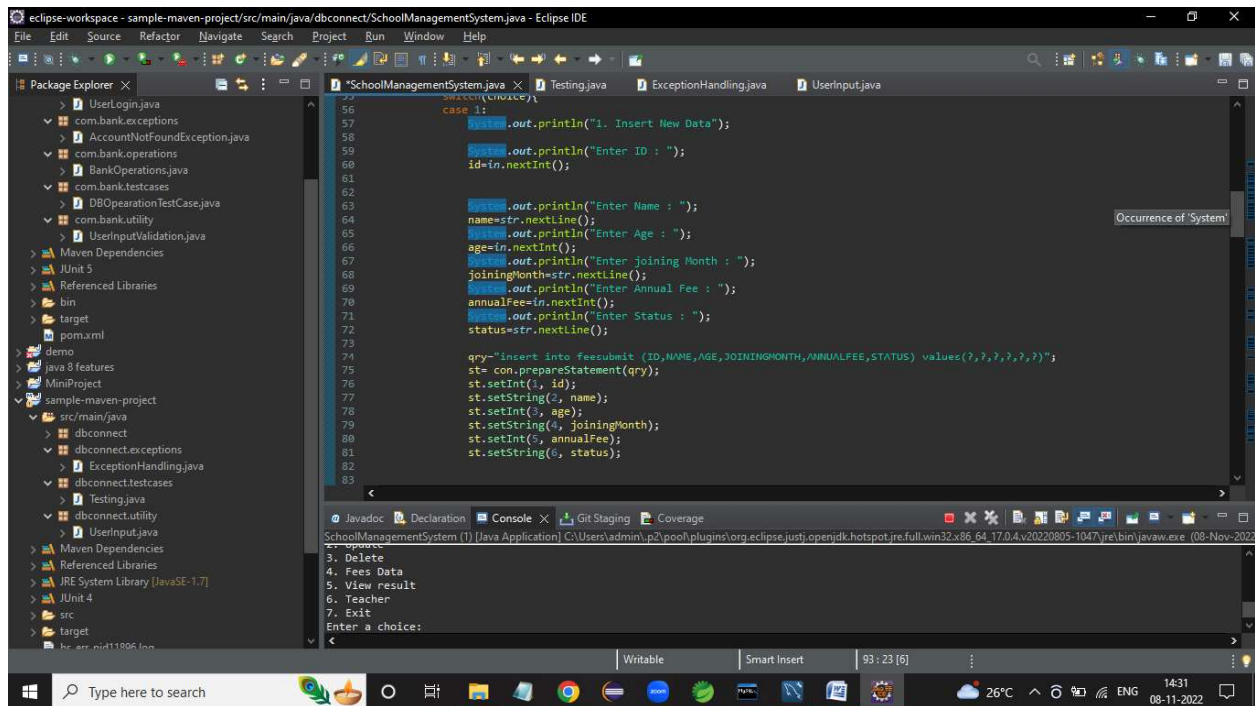
The 'Output' pane at the bottom shows the execution log:

#	Time	Action	Message	Duration / Fetch
3	11:19:49	CREATE TABLE 'reportcard' ('id' int(255) NOT NULL, 'name' varchar(255) NOT NULL, ...	0 row(s) affected, 1 warning(s): 1681 Integer display width is deprecated and will be removed ...	0.250 sec
4	11:19:57	INSERT INTO 'reportcard' ('id', 'name', 'std', 'rollnumber', 'Result') VALUES (1, 'Bharani', 'X...	9 row(s) affected Records: 9 Duplicates: 0 Warnings: 0	0.110 sec
5	11:20:05	select * from reportcard LIMIT 0, 1000	9 row(s) returned	0.000 sec / 0.000 sec
6	11:23:08	INSERT INTO 'Teacher' ('Name','subject','City','Salary') VALUES (Siva,physics,chennai,54...	Error Code: 1054. Unknown column 'Siva' in 'field list'	0.000 sec
7	11:25:02	INSERT INTO 'Teacher' ('Name','subject','City','Salary') VALUES (Siva,'physics','chennai'...	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0	0.079 sec
8	11:25:05	Select * from Teacher LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec

Implementing JDBC Connection



Code for Insertion

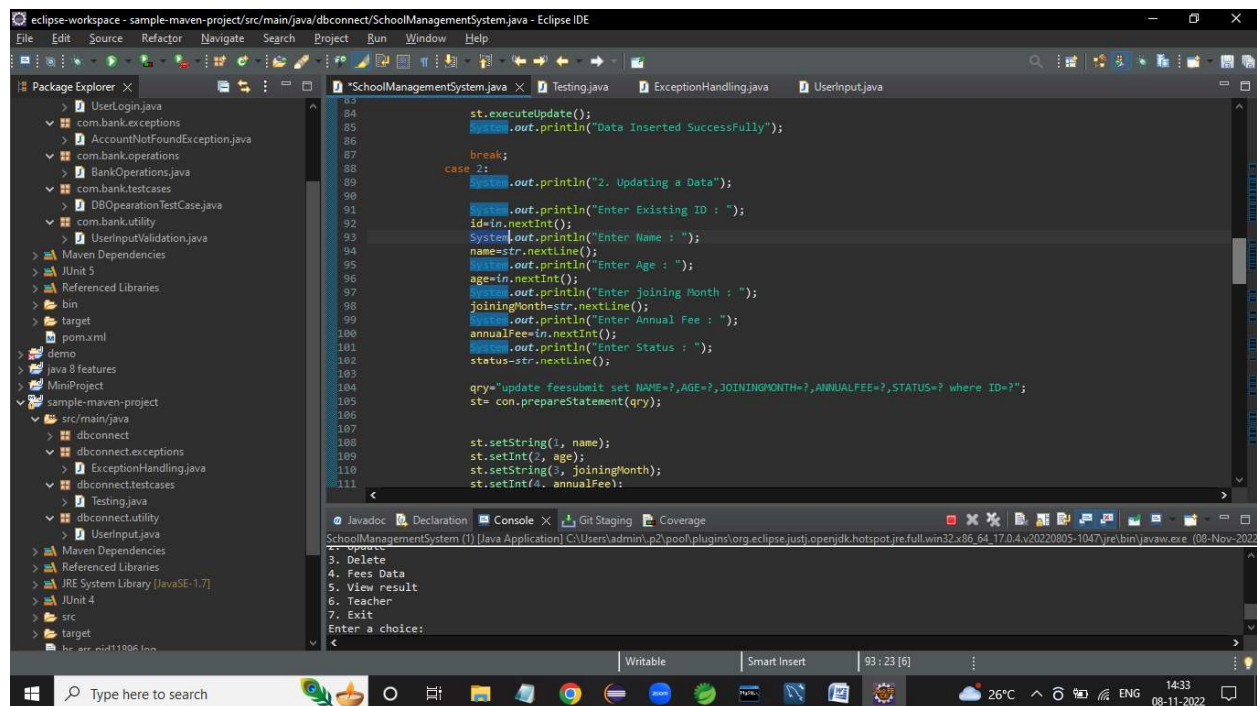


```
156 switch(choice)
157 case 1:
158     System.out.println("1. Insert New Data");
159     System.out.println("Enter ID : ");
160     id=in.nextInt();
161
162     System.out.println("Enter Name : ");
163     name=nr.nextLine();
164     System.out.println("Enter Age : ");
165     age=in.nextInt();
166     System.out.println("Enter joining Month : ");
167     joiningMonth=nr.nextLine();
168     System.out.println("Enter Annual Fee : ");
169     annualFee=in.nextInt();
170     System.out.println("Enter Status : ");
171     status=nr.nextLine();
172
173     qry="insert into feesubmit (ID,NAME,AGE,JOININGMONTH,ANNUALFEE,STATUS) values(?,?,?,?,?,?)";
174     st= con.prepareStatement(qry);
175     st.setInt(1, id);
176     st.setString(2, name);
177     st.setInt(3, age);
178     st.setString(4, joiningMonth);
179     st.setInt(5, annualFee);
180     st.setString(6, status);
181
182
183
```

SchoolManagementSystem (1) [Java Application] C:\Users\admin\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)

```
3. Delete
4. Fees Data
5. View result
6. Teacher
7. Exit
Enter a choice:
```

Code for Updating

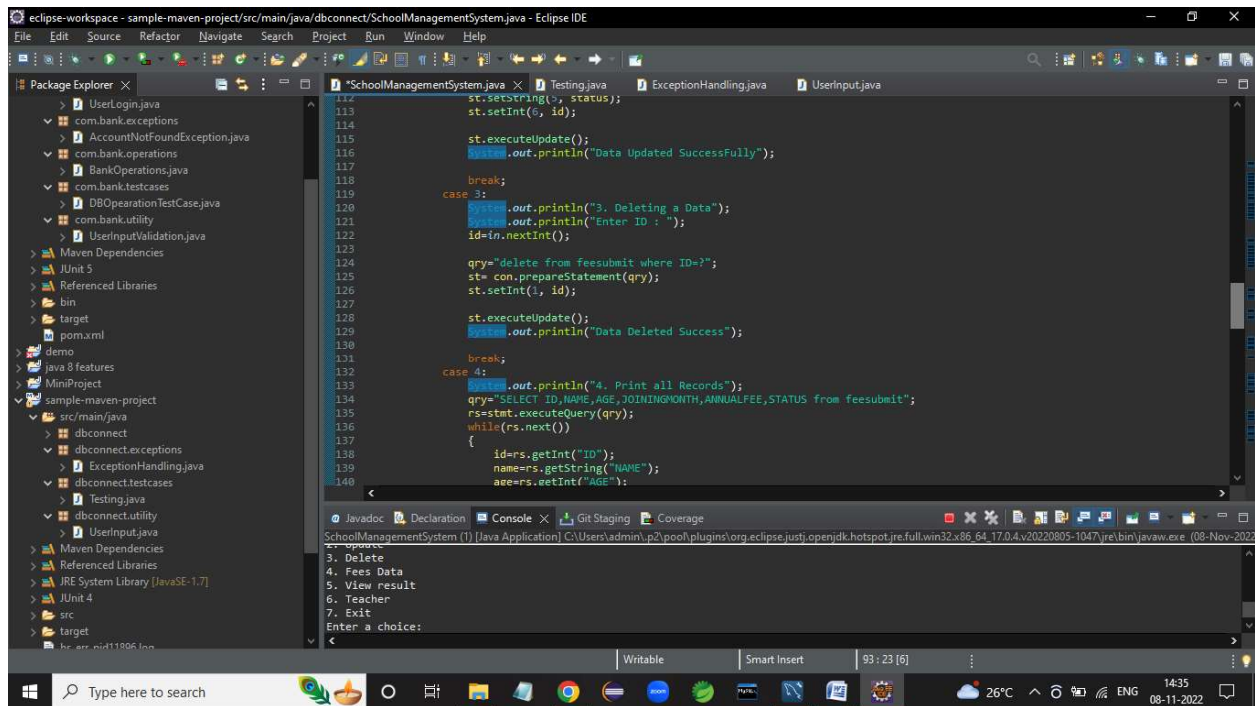


```
83
84     st.executeUpdate();
85     System.out.println("Data Inserted Successfully");
86
87     break;
88     case 2:
89         System.out.println("2. Updating a Data");
90         System.out.println("Enter Existing ID : ");
91         id=in.nextInt();
92         System.out.println("Enter Name : ");
93         name=st.nextLine();
94         System.out.println("Enter Age : ");
95         age=in.nextInt();
96         System.out.println("Enter joining Month : ");
97         joiningMonth=st.nextLine();
98         System.out.println("Enter Annual Fee : ");
99         annualFee=in.nextInt();
100        System.out.println("Enter Status : ");
101        status=st.nextLine();
102
103        qry="update feesubmit set NAME=?,AGE=?,JOININGMONTH=?,ANNUALFEE=?,STATUS=? where ID=?";
104        st= con.prepareStatement(qry);
105
106
107
108        st.setString(1, name);
109        st.setInt(2, age);
110        st.setString(3, joiningMonth);
111        st.setInt(4, annualFee);
```

SchoolManagementSystem (1) [Java Application] C:\Users\admin\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)

3. Delete
4. Fees Data
5. View result
6. Teacher
7. Exit
Enter a choice:

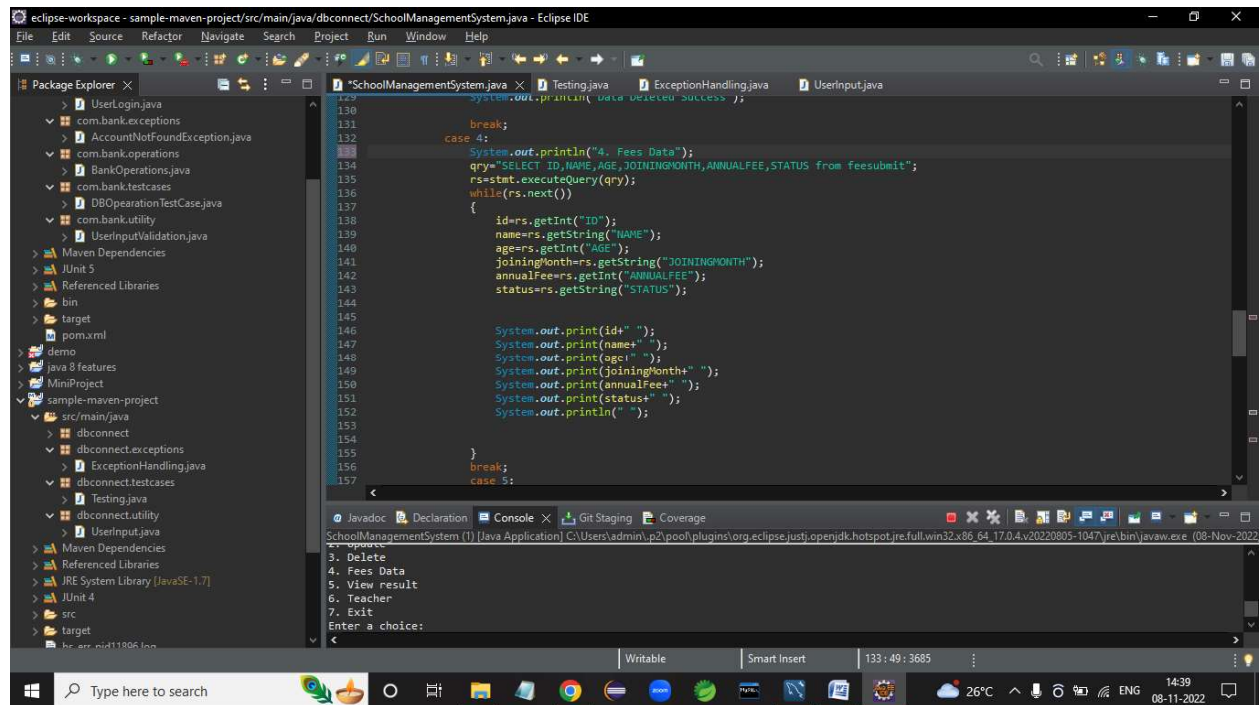
Code for Deletion



```
112 st.setString(5, status);
113 st.setInt(6, id);
114
115 st.executeUpdate();
116 System.out.println("Data Updated Successfully");
117
118 break;
119
120 case 3:
121     System.out.println("3. Deleting a Data");
122     System.out.println("Enter ID : ");
123     id=in.nextInt();
124
125     qry="delete from feesubmit where ID=?";
126     stmt.con.prepareStatement(qry);
127     st.setInt(1, id);
128
129     st.executeUpdate();
130     System.out.println("Data Deleted Success");
131
132     break;
133
134 case 4:
135     System.out.println("4. Print all Records");
136     qry="SELECT ID,NAME,AGE,JOININGMONTH,ANNUALFEE,STATUS from feesubmit";
137     rs=stmt.executeQuery(qry);
138     while(rs.next())
139     {
140         id=rs.getInt("ID");
141         name=rs.getString("NAME");
142         age=rs.getInt("AGE");
143     }
144 }
```

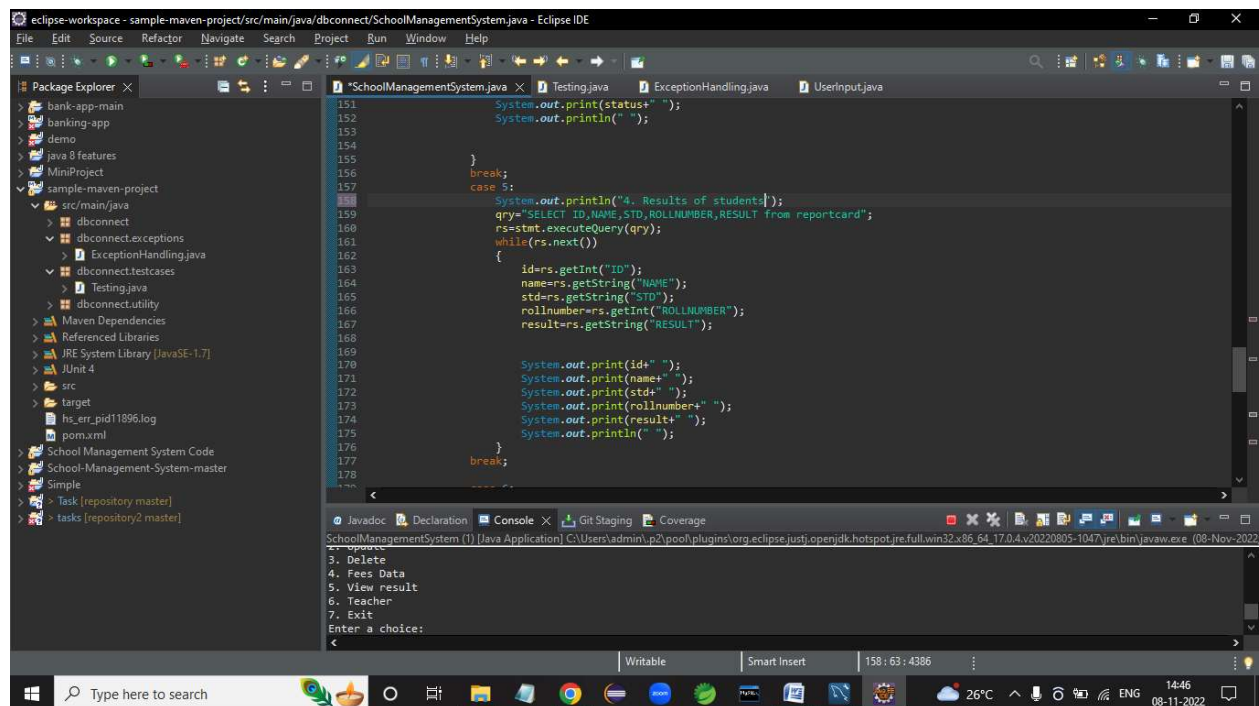
3. Delete
4. Fees Data
5. View result
6. Teacher
7. Exit
Enter a choice:

Code for View Fees Data



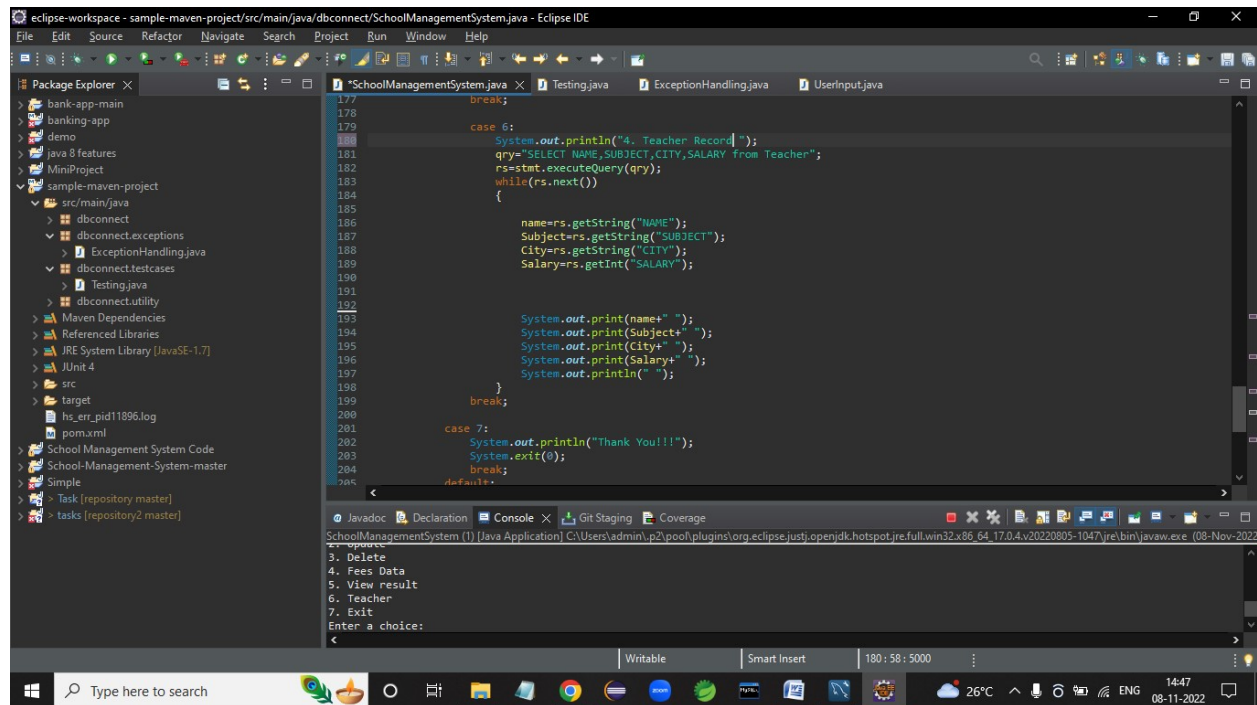
```
129      System.out.println("Data Deleted Success");
130
131      break;
132
133      case 4:
134          System.out.println("4. Fees Data");
135          qry="SELECT ID,NAME,AGE,JOININGMONTH,ANNUALFEE,STATUS from feesubmit";
136          rs=stmt.executeQuery(qry);
137          while(rs.next())
138          {
139              id=rs.getInt("ID");
140              name=rs.getString("NAME");
141              age=rs.getInt("AGE");
142              joiningMonth=rs.getString("JOININGMONTH");
143              annualFee=rs.getInt("ANNUALFEE");
144              status=rs.getString("STATUS");
145
146              System.out.print(id+" ");
147              System.out.print(name+" ");
148              System.out.print(age+" ");
149              System.out.print(joiningMonth+" ");
150              System.out.print(annualFee+" ");
151              System.out.print(status+" ");
152              System.out.println(" ");
153
154          }
155      }
156      break;
157      case 5:
```

Code for View Result



```
151      System.out.print(status+" ");
152      System.out.println(" ");
153
154      }
155
156      break;
157
158      case 5:
159          System.out.println("5. Results of student:");
160          qry="SELECT ID,NAME,STD,ROLLNUMBER,RESULT from reportcard";
161          rs=stat.executeQuery(qry);
162          while(rs.next())
163          {
164              id=rs.getInt("ID");
165              name=rs.getString("NAME");
166              std=rs.getString("STD");
167              rollnumber=rs.getInt("ROLLNUMBER");
168              result=rs.getString("RESULT");
169
170              System.out.print(id+" ");
171              System.out.print(name+" ");
172              System.out.print(std+" ");
173              System.out.print(rollnumber+" ");
174              System.out.print(result+" ");
175              System.out.println(" ");
176
177          }
178      }
179      break;
180      case 6:
```

Code for view Teacher table



The screenshot shows the Eclipse IDE with the following components:

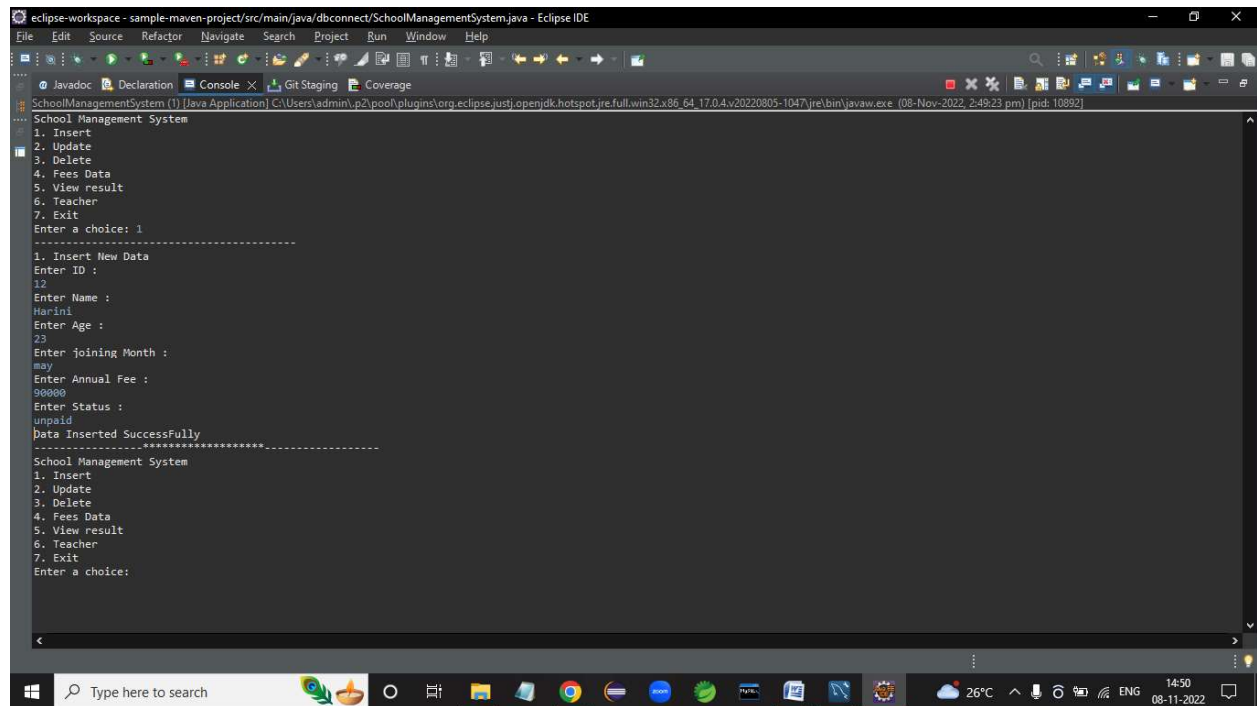
- Package Explorer:** Displays the project structure, including `src/main/java/dbconnect` and its sub-packages like `exceptions`, `testcases`, `utility`, and `utility`.
- Editor:** Shows the `SchoolManagementSystem.java` file. The code is as follows:

```
177 break;
178
179 case 6:
180     System.out.println("4. Teacher Record");
181     qry="SELECT NAME,SUBJECT,CITY,SALARY from Teacher";
182     rs=stat.executeQuery(qry);
183     while(rs.next())
184     {
185         name=rs.getString("NAME");
186         Subject=rs.getString("SUBJECT");
187         City=rs.getString("CITY");
188         Salary=rs.getInt("SALARY");
189
190         System.out.print(name+" ");
191         System.out.print(Subject+" ");
192         System.out.print(City+" ");
193         System.out.print(Salary+" ");
194         System.out.println(" ");
195     }
196     break;
197
198 case 7:
199     System.out.println("Thank You!!!");
200     System.exit(0);
201     break;
202
203 default:
204     break;
```
- Console:** Shows the output of the program, including the menu options and the result of the "View result" option:

```
3. Delete
4. Fees Data
5. View result
6. Teacher
7. Exit
Enter a choice:
4
```
- Bottom Bar:** Displays the status bar with the text "Writable", "Smart Insert", and the coordinates "180: 58: 5000".

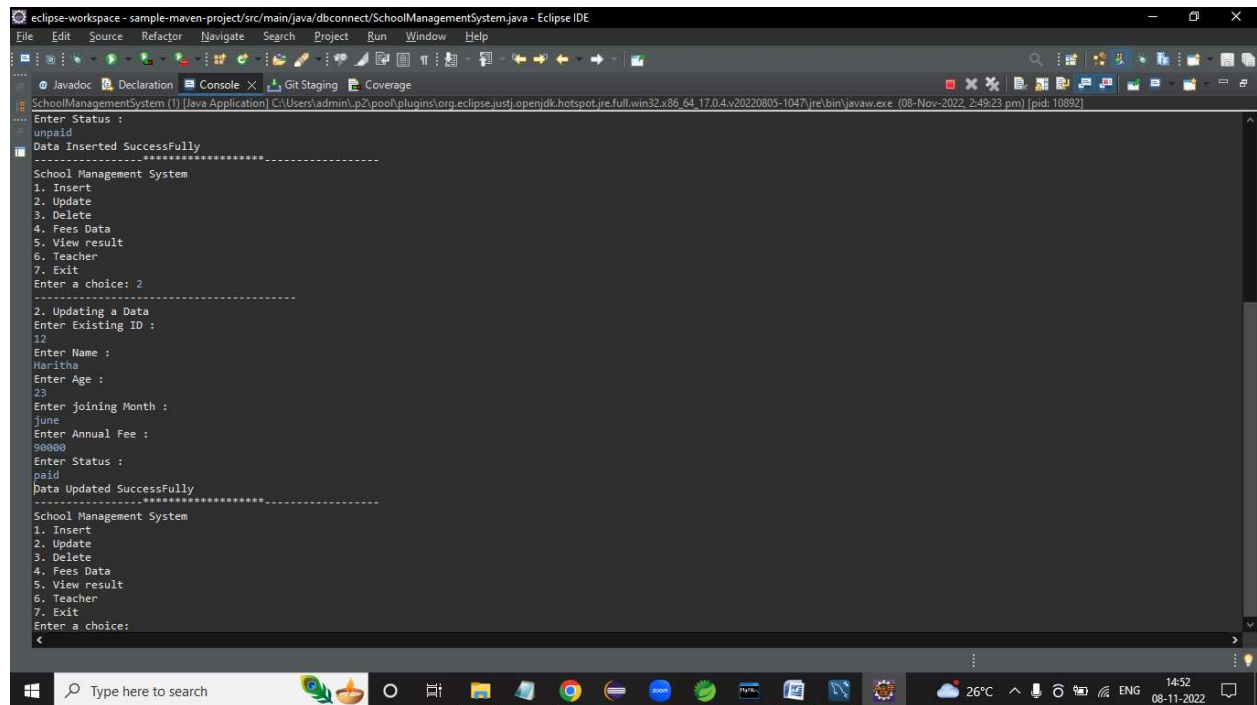
OUTPUT

Output for Insertion



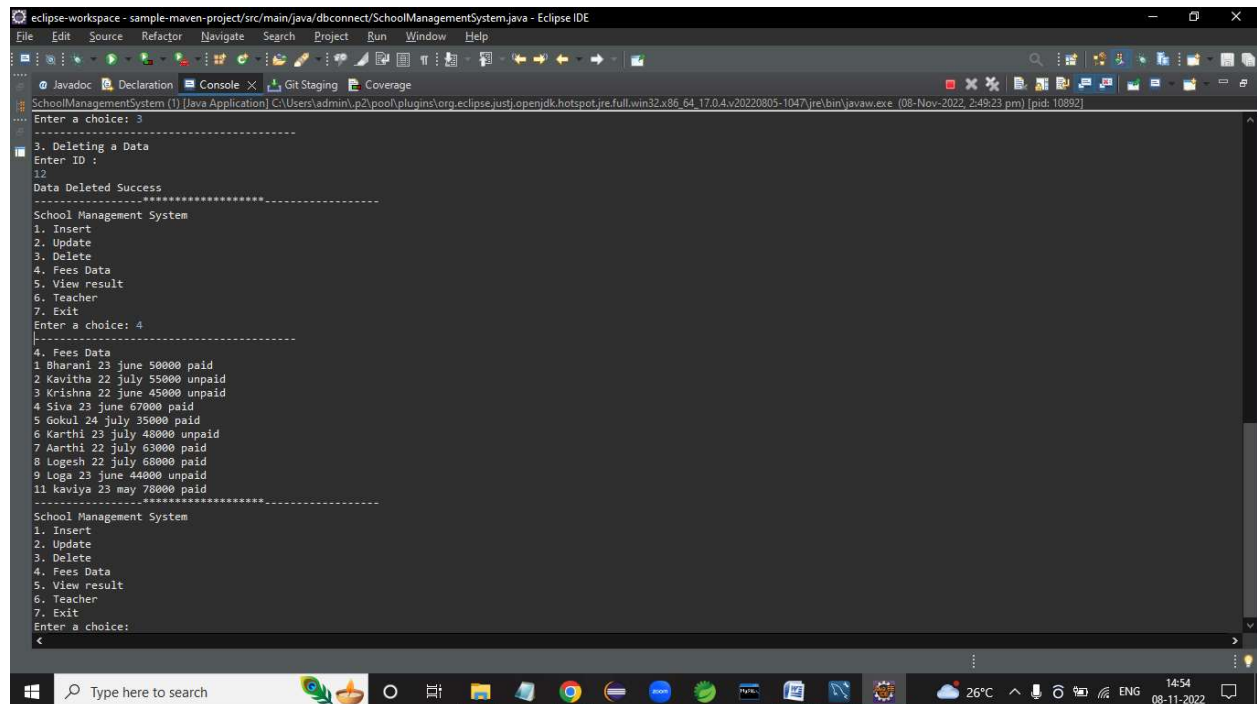
```
eclipse-workspace - sample-maven-project/src/main/java/dbconnect/SchoolManagementSystem.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
@ Javadoc Declaration Console X Git Staging Coverage
SchoolManagementSystem (1) [Java Application] C:\Users\admin\p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64.17.0.4.v20220805-1047\jre\bin\javaw.exe (08-Nov-2022, 2:49:23 pm) [pid: 10892]
School Management System
1. Insert
2. Update
3. Delete
4. Fees Data
5. View result
6. Teacher
7. Exit
Enter a choice: 1
-----
1. Insert New Data
Enter ID :
12
Enter Name :
Harini
Enter Age :
23
Enter joining Month :
may
Enter Annual Fee :
90000
Enter Status :
unpaid
Data Inserted Successfully
-----
School Management System
1. Insert
2. Update
3. Delete
4. Fees Data
5. View result
6. Teacher
7. Exit
Enter a choice:
```

Output for Updation



```
eclipse-workspace - sample-maven-project/src/main/java/dbconnect/SchoolManagementSystem.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
SchoolManagementSystem (1) [Java Application] C:\Users\admin\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, 2:49:23 pm) [pid: 10892]
Enter Status :
unpaid
Data Inserted Successfully
-----
School Management System
1. Insert
2. Update
3. Delete
4. Fees Data
5. View result
6. Teacher
7. Exit
Enter a choice: 2
-----
2. Updating a Data
Enter Existing ID :
12
Enter Name :
Haritha
Enter Age :
23
Enter joining Month :
june
Enter Annual Fee :
90000
Enter Status :
paid
Data Updated Successfully
-----
School Management System
1. Insert
2. Update
3. Delete
4. Fees Data
5. View result
6. Teacher
7. Exit
Enter a choice:
<
```

Output for View fees data



```
eclipse-workspace - sample-maven-project/src/main/java/dbconnect/SchoolManagementSystem.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
SchoolManagementSystem (1) [Java Application] C:\Users\admin\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, 2:49:23 pm) [pid: 10892]
Enter a choice: 3
-----
3. Deleting a Data
Enter ID :
12
Data Deleted Success
-----
School Management System
1. Insert
2. Update
3. Delete
4. Fees Data
5. View result
6. Teacher
7. Exit
Enter a choice: 4
-----
4. Fees Data
1. Bharani 23 june 50000 paid
2. Kavitha 22 july 55000 unpaid
3. Krishna 22 june 45000 unpaid
4. Siva 23 june 67000 paid
5. Gokul 24 july 35000 paid
6. Karthi 23 july 48000 unpaid
7. Aarthi 22 july 63000 paid
8. Logesh 22 july 68000 paid
9. Loga 23 june 44000 unpaid
10. kaviya 23 may 78000 paid
-----
School Management System
1. Insert
2. Update
3. Delete
4. Fees Data
5. View result
6. Teacher
7. Exit
Enter a choice:
<
```


Output for View Result

```
eclipse-workspace - sample-maven-project/src/main/java/dbconnect/SchoolManagementSystem.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help

SchoolManagementSystem (1) [Java Application] C:\Users\admin\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, 2:49:23 pm) [pid: 10892]

5 Gokul 24 july 35000 paid
6 Karthi 23 july 40000 unpaid
7 Aarthi 22 july 63000 paid
8 Logesh 22 july 68000 paid
9 Loga 23 june 44000 unpaid
11 kaviya 23 may 78000 paid
12 Haritha 23 may 89000 paid
-----
School Management System
1. Insert
2. Update
3. Delete
4. Fees Data
5. View result
6. Teacher
7. Exit
Enter a choice: 5
|-----
4. Results of students
1 Bharani X 101 Pass
2 Kavitha X 102 Pass
3 Krishna X 103 Fail
4 Siva X 104 Pass
5 Gokul X 105 Pass
6 Karthi XI 106 Pass
7 Aarthi XI 107 Pass
8 Logesh X 108 Pass
9 Loga XI 109 Fail
-----
School Management System
1. Insert
2. Update
3. Delete
4. Fees Data
5. View result
6. Teacher
7. Exit
Enter a choice:
<
```

Output for view teacher

```
eclipse-workspace - sample-maven-project/src/main/java/dbconnect/SchoolManagementSystem.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help

SchoolManagementSystem (1) [Java Application] C:\Users\admin\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, 2:49:23 pm) [pid: 10892]

4. Results of students
1 Bharani X 101 Pass
2 Kavitha X 102 Pass
3 Krishna X 103 Fail
4 Siva X 104 Pass
5 Gokul X 105 Pass
6 Karthi XI 106 Pass
7 Aarthi XI 107 Pass
8 Logesh X 108 Pass
9 Loga XI 109 Fail
-----
School Management System
1. Insert
2. Update
3. Delete
4. Fees Data
5. View result
6. Teacher
7. Exit
Enter a choice: 6
|-----
4. Teacher Record
Siva physics chennai 54000
Ramya maths chennai 80000
Satish computer chennai 56000
Kumaran Chemistry chennai 67000
Priya english chennai 45000
-----
School Management System
1. Insert
2. Update
3. Delete
4. Fees Data
5. View result
6. Teacher
7. Exit
Enter a choice:
<
```

Exit Operation

```
School Management System
1. Insert
2. Update
3. Delete
4. Fees Data
5. View result
6. Teacher
7. Exit
Enter a choice: 7|
-----
Thank You!!!
```

Conclusion

The project work titled “School Management System” has been designed using JAVA where in many user friendly form controls have been added in order to make it a user interactive Console-Based application. The system is developed in such a way that the user with common knowledge of computers can handle it easily. The System developed has proved to be user friendly and efficient in achieving basic goals. The system takes care of all the constraints which have specified. The system is found to be really beneficial for the concerned aspects. Application developed is realistic and secure.

