Core Java Project

Project Name : Student Management System

Student Name : Pravin Kolate

Guide By : MS. Indrakka Mali

Under By : EduBridge

Batch No : 8107

Introduction

Student 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 Student Management System deals with the various activities related to the students.

There are mainly two modules in this software

- Admin
- Student

In the Software we can register as a user and the user has two types, student and administrator. Administrator has the power to add new users and can edit and delete a user. A student can register as a user and can add, edit and delete his profile. The administrator can add edit and delete marks for the student. All the users can see the marks.

Requirements

- Programming Language : Java
- Database : MySQL
- IDE : Eclipse

Operations

- **Admin**

    1. Add new Student
    2. Update Student
    3. Delete Student
    4. Manage Course
    5. Show All Student
    6. DashBoard
    7. Get Status Of Student

- **Student**

    1. View All Account Details

Initial Setup

1. **Coding Setup**

Maven Project Name : Student Management System
Package Name : com.programs
Class Name : 1. StudentMain
2. ShowMenu
3. DataBaseConnection
4. DataBaseOperation
5. DashBoard
6. UpdateStudent
7. CheckValidation
8. ManageCourse

2. **DataBase Setup**

DataBase Name : javaproject
Table Name : student , course

1. StudentMain Class

```
package com.programs;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.Scanner;

public class StudentMain {
	private static Connection conn;
	private static PreparedStatement pst;
	private static ResultSet rs;

	public static void main(String[] args) throws SQLException {
		Scanner sc=new Scanner(System.in);
		// Show menu
		System.out.println("**************Student Management System******************");
		while(true) {
		System.out.println("Select Login Option");
		System.out.println("1.Admin	2.Student");
		int loginch=sc.nextInt();

		switch(loginch) {
		case 1:
			while(true) {
				try {
					conn=DatabaseConnection.getConnection();
					System.out.println("Enter UserName");
					String username=sc.next();
					System.out.println("Enter Password");
					String pass=sc.next();
					String s="select * from admin where username=? and password=?";
					pst=conn.prepareStatement(s);
					pst.setString(1, username);
					pst.setString(2, pass);
					rs=pst.executeQuery();
					if(rs.next()) {
						ShowMenu.showMenu();
						break;
					}
					else {
						System.out.println("Username and password Incorrect...");
					}
				}
				catch(Exception e) {
					e.printStackTrace();
```

```
                                        }
                                }
                                System.out.println();
                                break;

                        case 2:
                                DatabaseOperations.getStatus();
                                break;
                        default:
                                System.out.println("Enter Valid Choice");
                        }
                        System.out.println("Do You Want to Cotinue press y/n");
                        char ch=sc.next().charAt(0);
                        if(ch=='n') {
                                break;
                        }
                        }

        }


}
```

2. ShowMenu Class

```
package com.programs;

import java.sql.SQLException;
import java.util.Scanner;

public class ShowMenu {
   public static void showMenu() throws SQLException {
          Scanner sc=new Scanner(System.in);
          System.out.println("----------------------------------------");
          while(true) {
                  System.out.println("1.Add New Student         2.Show All Student");
                  System.out.println("3.Update Student         4.Delete Student");
                  System.out.println("5.Get Status Of Student    6.DashBoard");
                  System.out.println("7.Manage Course");
                  System.out.println("----------------------------------------");
                  System.out.println("Enter Your Choice");
            int choice =sc.nextInt();

            switch(choice) {
            case 1:
                    System.out.println("Add New Student");
                    DatabaseOperations.addStudent();
                    break;
            case 2:
                      System.out.println("Show All Student");
                      DatabaseOperations.displayAllStudent();
                      break;
            case 3:
                    System.out.println("Update Student ");
                    DatabaseOperations.updateStudent();
                    break;
            case 4:
                    System.out.println("Delete Student");
                    DatabaseOperations.deleteStudent();
                    break;
            case 5:
                    System.out.println("Status Of Student");
                    DatabaseOperations.getStatus();
                    break;
            case 6:
                      System.out.println("DashBoard");
                      DashBoard.getStatus();
                      break;
            case 7:
                      System.out.println("Manage Course");
                      ManageCourse.manageCourse();
                      break;

            }
```

```
            System.out.println();
            System.out.println("Do You Want To Continue... press y/n");
            char ch=sc.next().charAt(0);
            if(ch=='n') {
                  break;
            }

          }
    }
}
```

3. UpdateStudent Class

```
package com.programs;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.Scanner;

public class UpdateStudent {
	// set connection requirement
	private static Connection conn;
	private static PreparedStatement pst;
	private static ResultSet rs,rs1;
  static Scanner sc=new Scanner(System.in);

  // update Student Name
	public static void updateName() throws SQLException {
		try {
			conn=DatabaseConnection.getConnection();
			System.out.println("Enter Student ID");
			int sid=sc.nextInt();
			String s="Select * from student where sid=?";
			pst=conn.prepareStatement(s);
			pst.setInt(1, sid);
			rs=pst.executeQuery();
			if(rs.next()) {
				System.out.println("Enter Student Name");
				String name=sc.next();
				String uname="update student set sname=? where sid=?";
				pst=conn.prepareStatement(uname);
				pst.setString(1, name);
				pst.setInt(2, sid);
				int result=pst.executeUpdate();
				if(result>0) {
					System.out.println("Student Name Update.");
				}else {
					System.out.println("Student Name Not Update");
				}
			}else {
				System.out.println("Student ID Not Found.");
			}


		}catch(Exception e) {
			e.printStackTrace();
		}
		finally {
			conn.close();
			pst.close();
			rs.close();
```

```
		}

	}

	// Update Student Email
	public static void updateEmail() throws SQLException {
		try {
			conn=DatabaseConnection.getConnection();
			System.out.println("Enter Student ID");
			int sid=sc.nextInt();
			String s="Select * from student where sid=?";
			pst=conn.prepareStatement(s);
			pst.setInt(1, sid);
			rs=pst.executeQuery();
			if(rs.next()) {
				System.out.println("Enter Student Email");
				String email=sc.next();
				String uemail="update student set semail=? where sid=?";
				pst=conn.prepareStatement(uemail);
				pst.setString(1, email);
				pst.setInt(2, sid);
				int result=pst.executeUpdate();
				if(result>0) {
					System.out.println("Student Email Update.");
				}else {
					System.out.println("Student Email Not Update");
				}
			}else {
				System.out.println("Student ID Not Found.");
			}


		}catch(Exception e) {
			e.printStackTrace();
		}
		finally {
			conn.close();
			pst.close();
			rs.close();

		}

	}

	// Update student Mobile Number

	public static void updateMobile() throws SQLException {
		try {
			conn=DatabaseConnection.getConnection();
			System.out.println("Enter Student ID");
			int sid=sc.nextInt();
			String s="Select * from student where sid=?";
```

```
                pst=conn.prepareStatement(s);
                pst.setInt(1, sid);
                rs=pst.executeQuery();
                if(rs.next()) {
                        System.out.println("Enter Student Mobile");
                        String mobile=sc.next();
                        String umobile="update student set smobile=? where sid=?";
                        pst=conn.prepareStatement(umobile);
                        pst.setString(1, mobile);
                        pst.setInt(2, sid);
                        int result=pst.executeUpdate();
                        if(result>0) {
                                System.out.println("Student Mobile Update.");
                        }else {
                                System.out.println("Student Mobile Not Update");
                        }
                }else {
                        System.out.println("Student ID Not Found.");
                }


        }catch(Exception e) {
                e.printStackTrace();
        }
        finally {
                conn.close();
                pst.close();
                rs.close();

        }

}
// update student Course

public static void updateCourse() throws SQLException {
        try {
                conn=DatabaseConnection.getConnection();
                System.out.println("Enter Student ID");
                int sid=sc.nextInt();
                String s="Select * from student where sid=?";
                pst=conn.prepareStatement(s);
                pst.setInt(1, sid);
                rs=pst.executeQuery();
                if(rs.next()) {
                        System.out.println("Choose Course");
                  System.out.println("11. Java Full Stack (RS.5000)     22. C++ (RS.2000)");
             System.out.println("33. Python       (RS.3000)     44. DS (RS.4000)");
                   int cid=sc.nextInt();
                        String ucourse="update student set cid=? where sid=?";
                        pst=conn.prepareStatement(ucourse);
                        pst.setInt(1, cid);
                        pst.setInt(2, sid);
                        int result=pst.executeUpdate();
```

```
                        if(result>0) {
                                System.out.println("Student Course Update.");
                        }else {
                                System.out.println("Student Course Not Update");
                        }
                }else {
                        System.out.println("Student ID Not Found.");
                }


        }catch(Exception e) {
                e.printStackTrace();
        }
        finally {
                conn.close();
                pst.close();
                rs.close();

        }

    }
    //update Student Fees.

    public static void updateFees() throws SQLException {
        try {
                conn=DatabaseConnection.getConnection();
                System.out.println("Enter Student ID");
                int sid=sc.nextInt();
                String s="Select * from student where sid=?";
                pst=conn.prepareStatement(s);
                pst.setInt(1, sid);
                rs=pst.executeQuery();
                if(rs.next()) {
                        System.out.println("Enter Student Fees");
                        float fees=sc.nextFloat();
                        if(fees<0) {
                                System.out.println("Enter Valid Fees");
                        }else {
                                fees+=rs.getFloat(6);
                                String s1="select cfees from course where cid=?";
                                pst=conn.prepareStatement(s1);
                                pst.setFloat(1, rs.getInt(5));
                                rs1=pst.executeQuery();
                                if(rs1.next()) {
                                        if(fees>rs1.getFloat(1)) {
                                                System.out.println("Enter Valid Fees");
                                        }
                                        else {
                                                String ufees="update student set balance=? where
sid=?";
                                                pst=conn.prepareStatement(ufees);
                                                pst.setFloat(1, fees);
                                                pst.setInt(2, sid);
```

```
                                                int result=pst.executeUpdate();
                                                if(result>0) {
                                                        System.out.println("Student Fees Update.");
                                                }else {
                                                        System.out.println("Student Fees Not
Update");
                                                }
                                        }
                                }
                        }



                }else {
                        System.out.println("Student ID Not Found.");
                }


        }catch(Exception e) {
                e.printStackTrace();
        }
        finally {
                conn.close();
                pst.close();
                rs.close();

        }


    }
}
```

4. ManageCourse Class

```
package com.programs;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.Scanner;

public class ManageCourse {
	// set connection requirement
	private static Connection conn;
	private static PreparedStatement pst;
	private static ResultSet rs;
	static Scanner sc=new Scanner(System.in);

	public static void manageCourse() throws SQLException {
		conn=DatabaseConnection.getConnection();

		while(true) {
			System.out.println("1.Add New Course     2.Update Course");
			System.out.println("3.Delete Course      4.Show All Course");
			System.out.println("Enter Your Choice");
			int ch=sc.nextInt();

			switch(ch) {
			case 1:
				System.out.println("Add New Course");
				addnewCourse();
				break;
			case 2:
				System.out.println("Update Course");
				updateCourse();
				break;
			case 3:
				System.out.println("Delete Course");
				deleteCourse();
				break;
			case 4:
				System.out.println("Show All Course");
				showCourse();
				break;
			default:
				System.out.println("Enter Valid choice");
			}
			System.out.println("Do You Want To Continue press y/n");
			char choice=sc.next().charAt(0);
			if(choice=='n') {
				break;
```

```
                }
        }

}

// update Course

private static void updateCourse() {
        try {
                while(true) {
                        System.out.println("1.Update Cname");
                        System.out.println("2.Update Fees");
                        System.out.println("Enter Your Choice");
                        int ch=sc.nextInt();

                        switch(ch) {
                        case 1:
                                System.out.println("Enter CID");
                                int cid=sc.nextInt();
                                String s="select * from course where cid=?";
                                pst=conn.prepareStatement(s);
                                pst.setInt(1, cid);
                                rs=pst.executeQuery();
                                if(rs.next()) {
                                        System.out.println("Enter Cname");
                                        String cname=sc.next();
                                        String s1="update course set cname=? where cid=?";
                                        pst=conn.prepareStatement(s1);
                                        pst.setString(1, cname);
                                        pst.setInt(2, cid);
                                        int i=pst.executeUpdate();
                                        if(i>0) {
                                                System.out.println("Course Name Update.");
                                        }else {
                                                System.out.println("Course Name Not Update");
                                        }
                                }
                                break;
                        case 2:
                                System.out.println("Enter CID");
                                int cid1=sc.nextInt();
                                String s1="select * from course where cid=?";
                                pst=conn.prepareStatement(s1);
                                pst.setInt(1, cid1);
                                rs=pst.executeQuery();
                                if(rs.next()) {
                                        System.out.println("Enter Cfees");
                                        float cfees=sc.nextFloat();
                                        String s2="update course set cfees=? where cid=?";
                                        pst=conn.prepareStatement(s2);
                                        pst.setFloat(1, cfees);
                                        pst.setInt(2, cid1);
                                        int i=pst.executeUpdate();
```

```
                                        if(i>0) {
                                                System.out.println("Course Fees Update.");
                                        }else {
                                                System.out.println("Course Fees Not Update");
                                        }
                                }
                                break;
                        default:
                                System.out.println("Enter Valid Choice");
                        }

                        System.out.println("Do You Want To Contiune press y/n");
                        char cho=sc.next().charAt(0);
                        if(cho=='n') {
                                break;
                        }


                }

        }catch(Exception e) {
                e.printStackTrace();
        }
}

// Delete Course
private static void deleteCourse() throws SQLException {
        try {
                conn=DatabaseConnection.getConnection();
                System.out.println("Are You Sure To Delete Student Press y/n");
                char ch=sc.next().charAt(0);
                if(ch=='n') {

                }else {
                        System.out.println("Enter Course ID");
                        int cid=sc.nextInt();

                        String s="select * from course where cid=?";
                        pst=conn.prepareStatement(s);
                        pst.setInt(1, cid);
                        rs=pst.executeQuery();
                        if(rs.next()) {
                                String del="delete from course where cid=?";
                                pst=conn.prepareStatement(del);
                                pst.setInt(1, cid);
                                int result=pst.executeUpdate();
                                if(result>0) {
                                        System.out.println("Course Deleted.");
                                }else {
                                        System.out.println("Course Not Delete.");
                                }
                        }else {
                                System.out.println("Course ID Not Found.");
```

```
                        }
                }

        }catch(Exception e) {
                e.printStackTrace();
        }



}
// Add New Course

private static void addnewCourse() throws SQLException {
        int cid=0;
  String s="select max(cid) from course";
          pst=conn.prepareStatement(s);
          rs=pst.executeQuery();
          if(rs.next()) {
                  cid=rs.getInt(1)+1;
          }else {
                  System.out.println("Record Not fetch error occure");
          }
          System.out.println("Enter Course Name");
          String cname=sc.next();

          String c="select * from course where cname=?";
          pst=conn.prepareStatement(c);
          pst.setString(1, cname);
          rs=pst.executeQuery();
          if(rs.next()) {
                  System.out.println("Course Already Register");
          }
          else {
                  System.out.println("Enter Course Fees");
                  float cfees=sc.nextFloat();

                  String ins="insert into course values(?,?,?)";
                  pst=conn.prepareStatement(ins);
                  pst.setInt(1, cid);
                  pst.setString(2, cname);
                  pst.setFloat(3, cfees);
                  int i=pst.executeUpdate();
                  if(i>0) {
                          System.out.println("Course Register.");
                  }else {
                          System.out.println("Course Not Register.");
                  }
          }



}
```

```
	// Show All Course
	private static void showCourse() throws SQLException {
		String course="select * from course";
		  pst=conn.prepareStatement(course);
		  rs=pst.executeQuery();
		  System.out.printf("%-6s%-20s%6s\n","CID","CNAME","CFEES");
		  System.out.println("---------------------------------");
		  while(rs.next()) {
			  System.out.printf("%-6s%-20s%6s\n",rs.getInt(1),rs.getString(2),rs.getFloat(3));
		  }

	}

}
```

5. DataBaseOperation Class

```
package com.programs;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.Scanner;

public class DatabaseOperations {
	// set connection requirement
	private static Connection conn;
	private static PreparedStatement pst;
	private static ResultSet rs;
	private static ResultSet rs1;
  static Scanner sc=new Scanner(System.in);

	 // Display All Student
	public static void displayAllStudent() throws SQLException {
		try {
			conn=DatabaseConnection.getConnection();
			String s="Select * from student";
			pst=conn.prepareStatement(s);
			rs=pst.executeQuery();

System.out.println("**************************************************************************");

System.out.printf("%-6s%-10s%-20s%-15s%-3s%8s\n","SID","SNAME","SEMAIL","SMOBILE","CID","TFEES");

			System.out.println("--------------------------------------------------------------------------");
			while(rs.next()) {

System.out.printf("%-6s%-10s%-20s%-15s%-3s%8s\n",rs.getInt(1),rs.getString(2),rs.getString(3),rs.getBigDecimal(4),rs.getInt(5),rs.getFloat(6));


			}

System.out.println("**************************************************************************");
		}catch(Exception e) {
			e.printStackTrace();
		}
	  finally {
		conn.close();
		pst.close();
		rs.close();
	}


	}
```

```
// get Status of student

public static void getStatus() throws SQLException {
  try {
                conn=DatabaseConnection.getConnection();
          System.out.println("Enter Student ID");
          int id=sc.nextInt();

          String s="select * from student where sid=?";
          pst=conn.prepareStatement(s);
          pst.setInt(1, id);
          rs=pst.executeQuery();
          if(rs.next()) {
                String sts="select * from course where cid=?";
                pst=conn.prepareStatement(sts);
                pst.setInt(1, rs.getInt(5));
                rs1=pst.executeQuery();
                if(rs1.next()) {
                        System.out.println("-----------------------------------------------");
                        System.out.println("SID     : "+rs.getInt(1));
                        System.out.println("SName    : "+rs.getString(2));
                        System.out.println("SEmail   : "+rs.getString(3));
                        System.out.println("SMobile  : "+rs.getBigDecimal(4));
   System.out.println("CID     : "+rs1.getInt(1));
                        System.out.println("CName    : "+rs1.getString(2));
                        System.out.println("Cfees    : "+rs1.getFloat(3));
                        System.out.println("Total Pay : "+rs.getFloat(6));

                        // calculate student pending fees
                        float cfees=rs1.getFloat(3);
                        float sfees=rs.getFloat(6);
                        float total=cfees-sfees;
                        System.out.println("Fees Pending : "+total);
                        System.out.println("-----------------------------------------------");


                }else {
                        System.out.println("Record Not Found");
                }


          }else {
                System.out.println("ID Not Found");
          }


  }catch(Exception e) {
          e.printStackTrace();
  }
  finally {
                conn.close();
                pst.close();
```

```
                rs.close();
        }

}



 // Delete Student
public static void deleteStudent() throws SQLException {
        try {
                conn=DatabaseConnection.getConnection();
                System.out.println("Are You Sure To Delete Student Press y/n");
                char ch=sc.next().charAt(0);
                if(ch=='n') {

                }else {
                        System.out.println("Enter Student ID");
                        int sid=sc.nextInt();

                        String s="select * from student where sid=?";
                        pst=conn.prepareStatement(s);
                        pst.setInt(1, sid);
                        rs=pst.executeQuery();
                        if(rs.next()) {
                                String del="delete from student where sid=?";
                                pst=conn.prepareStatement(del);
                                pst.setInt(1, sid);
                                int result=pst.executeUpdate();
                                if(result>0) {
                                        System.out.println("Student Deleted.");
                                }else {
                                        System.out.println("Student Not Delete.");
                                }
                        }else {
                                System.out.println("Student ID Not Found.");
                        }
                }

        }catch(Exception e) {
                e.printStackTrace();
        }
        finally {
                conn.close();
                pst.close();
                rs.close();
        }

}

// add new Student

public static void addStudent() throws SQLException {
```

```
	try {
		conn=DatabaseConnection.getConnection();
		// get max id from table
		int sid=0,cid=0;
		String s="select max(sid) from student";
		pst=conn.prepareStatement(s);
		rs=pst.executeQuery();
		if(rs.next()) {
			sid=rs.getInt(1)+1;
		}else {
			System.out.println("Record Not fetch error occure");
		}
		System.out.println("Enter Name");
		String name=sc.next();
		System.out.println("Enter Email");
		String email=sc.next();
		System.out.println("Enter Mobile");
		long mobile=sc.nextLong();

		// check student already exists or not
	 int check=CheckValidation.checkStudent(mobile,email);
	 if(check==0) {
		while(true) {
			System.out.println("Choose Course");
			System.out.println("------------------------------");
			String course="select * from course";
			pst=conn.prepareStatement(course);
			rs=pst.executeQuery();
			System.out.printf("%-6s%-20s%6s\n","CID","CNAME","CFEES");
			System.out.println("---------------------------------");
			while(rs.next()) {

System.out.printf("%-6s%-20s%6s\n",rs.getInt(1),rs.getString(2),rs.getFloat(3));
			}
			 cid=sc.nextInt();
			if(cid==11 || cid==22 || cid==33 || cid==44) {
				 break;
			}else {
				System.out.println("Please Choose Valid Course.");
			}
		}
		while(true) {
			float amt=CheckValidation.validFees(cid);
			if(amt==0) {
				amt=CheckValidation.validFees(cid);
			}else {
				 String ins="insert into student values(?,?,?,?,?,?)";
					pst=conn.prepareStatement(ins);
					pst.setInt(1, sid);
					pst.setString(2, name);
					pst.setString(3, email);
					pst.setLong(4, mobile);
					pst.setInt(5, cid);
```

```
                                        pst.setFloat(6, amt);

                                        int result=pst.executeUpdate();
                                        if(result>0) {
                                                System.out.println("Student Add Successfully....");
                                                break;
                                        }else {
                                                System.out.println("Student Not Added");
                                        }
                        }
                        }
         }

        }catch(Exception e) {
                e.printStackTrace();
        }
        finally {
                        conn.close();
                        pst.close();
                        rs.close();
                }

       }

        // Update student
       public static void updateStudent() throws SQLException {
                while(true) {
                        System.out.println("----------------------------------------");
                        System.out.println("1.Update Name       2.Update Email");
                        System.out.println("3.Update Mobile     4.Update Course");
                        System.out.println("5.Update Fees");
                        System.out.println("----------------------------------------");
                        System.out.println("Enter You Choice");
                        int ch=sc.nextInt();
                        switch(ch) {
                        case 1:
                                System.out.println("Update Name");
                                UpdateStudent.updateName();
                                break;
                        case 2:
                                System.out.println("Update Email");
                                UpdateStudent.updateEmail();
                                break;
                        case 3:
                                System.out.println("Update Mobile");
                                UpdateStudent.updateMobile();
                                break;
                        case 4:
                                System.out.println("Update Course");
                                UpdateStudent.updateCourse();
                                break;
                        case 5:
                                System.out.println("Update Fees");
```

```
                                UpdateStudent.updateFees();
                                break;
                        default:
                                System.out.println("Please Enter Valid Choice");
                        }
                        System.out.println("Do You Want To Update More press y/n");
                        char choice=sc.next().charAt(0);
                        if(choice=='n') {
                                break;
                        }
                }

        }
}
```

6. DataBaseConnection Class

```
package com.programs;

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

public class DatabaseConnection {

	private static String driver="com.mysql.cj.jdbc.Driver";
	private static String un="root";
	private static String  up="root";
	private static String url="jdbc:mysql://localhost:3306/javaproject";
	private static Connection conn=null;

	public static Connection getConnection() {
		try {
			Class.forName(driver);
			conn=DriverManager.getConnection(url,un,up);

			if(conn==null) {
				System.out.println("Connection not establish");
				System.exit(0);
			}
		}catch(Exception e) {
			e.printStackTrace();
		}
		return conn;
	}


}
```

7. CheckValidationClass

```
package com.programs;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.util.Scanner;

public class CheckValidation {
	// set connection requirement
	private static Connection conn;
	private static PreparedStatement pst;
	private static ResultSet rs;
	 static Scanner sc=new Scanner(System.in);

	// check course fees.
	public static float validFees(int cid) {
		float amt=0.0f;

		 try {
			conn=DatabaseConnection.getConnection();
			System.out.println("Enter Amount To Pay");
			amt=sc.nextFloat();
			String s="select * from course where cid=?";
			pst=conn.prepareStatement(s);
			pst.setInt(1, cid);
			rs=pst.executeQuery();
			if(rs.next()) {
				if(amt>rs.getFloat(3)) {
					System.out.println("Enter Valid Course Fees.");
				}else if(amt<0) {
					System.out.println("Enter Valid Course Fees.");
				}
				else {
					return amt;

				}
			}

		  }catch(Exception e) {
			e.printStackTrace();
		 }
			return 0;
	  }



	// check student exists or not
```

```
	public static int checkStudent(long mobile,String email) {
		try {
			conn=DatabaseConnection.getConnection();
			String s="select * from student where semail=? and smobile=?;";
			 pst=conn.prepareStatement(s);
			 pst.setString(1, email);
			 pst.setLong(2, mobile);
			 rs=pst.executeQuery();
			 if(rs.next()) {
				System.out.println("Student Already Exists");
				return -1;
			 }
		}catch(Exception e) {
			e.printStackTrace();
		}
		return 0;
	}

}
```

8. DashBoard Class

```
package com.programs;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;

public class DashBoard {
	private static Connection conn;
	private static PreparedStatement pst;
	private static ResultSet rs,rs1;
	// get current status
	public static void getStatus() {
		int total_std = 0,total_crs=0;
		float pending_fees = 0,total_fees=0;

		try {
			conn=DatabaseConnection.getConnection();
			// total student
			String s="select count(sid) from student";
			pst=conn.prepareStatement(s);
			rs=pst.executeQuery();
			if(rs.next()) {
				total_std=rs.getInt(1);
			}else {
				System.out.println("Record Not Found");
			}

			// total course
			String s1="select count(cid) from course";
			pst=conn.prepareStatement(s1);
			rs=pst.executeQuery();
			if(rs.next()) {
				total_crs=rs.getInt(1);
			}else {
				System.out.println("Record Not Found");
			}

			// total fees
			String s2="select sum(balance) from student";
			pst=conn.prepareStatement(s2);
			rs=pst.executeQuery();
			if(rs.next()) {
				total_fees=rs.getInt(1);
			}else {
				System.out.println("Record Not Found");
			}

			// calculate pending fees;
			String s3="select * from student";
```

```
                        pst=conn.prepareStatement(s3);
                        rs=pst.executeQuery();
                        while(rs.next()) {
                                String s4="select cfees from course where cid=?";
                                pst=conn.prepareStatement(s4);
                                pst.setInt(1, rs.getInt(5));
                                rs1=pst.executeQuery();
                                if(rs1.next()) {
                                        pending_fees+=rs1.getFloat(1)-rs.getFloat(6);
                                }

                        }
                        System.out.println("---------------------------------");
                        System.out.println("Total Student     : "+total_std);
                        System.out.println("Total Course      : "+total_crs);
                        System.out.println("Total Fees Collect : "+total_fees);
                        System.out.println("Pending Fees      : "+pending_fees);

                }catch(Exception e) {
                        e.printStackTrace();
                }
        }

}
```

Input/output

- **Student Login**

********** Student Management System**********
Select Login Option
1.Admin 2.Student
2
Enter Student ID
112

SID : 112
SName : Vishal
SEmail : vishal@gmail.com
SMobile : 8978676767
CID : 33
CName : Python
Cfees : 3000.0
Total Pay : 1000.0
Fees Pending : 2000.0

Do You Want to Cotinue press y/n

- **Admin Login**

********** Student Management System**********
Select Login Option
1.Admin 2.Student
1
Enter UserName
pravin
Enter Password
pravin123

1.Add New Student 2.Show All Student
3.Update Student 4.Delete Student
5.Get Status Of Student 6.DashBoard
7.Manage Course

Enter Your Choice

- **Insert New Student**

```
Enter Your Choice
1
Add New Student

Enter Name
Pravin
Enter Email
pravin@gmai.com
Enter Mobile
9286868686
Choose Course
------------------------------
CID  CNAME          CFEES
---------------------------------
11   Java_FullStack    8000.0
22   C++            2000.0
33   Python         3000.0
44   DS             4000.0
11
Enter Amount To Pay
4000
Student Add Successfully....

Do You Want To Continue... press y/n
```

- **Show All Student**

```
Enter Your Choice
2
Show All Student
**************************************************************************
SID  SNAME   SEMAIL          SMOBILE     CID  TFEES
--------------------------------------------------------------------------
112  Vishal  vishal@gmail.com  8978676767   33  1000.0
113  Mahesh  mahesh@gmail.com  8978675656   44  2000.0
114  Pravin  pravin@gmail.com  9876565656   11  4000.0
115  Pravin  pravin@gmai.com   9286868686   11  4000.0
**************************************************************************

Do You Want To Continue... press y/n
```

- **DashBoard**

```
Enter Your Choice
6
DashBoard
---------------------------------
Total Student     : 4
Total Course      : 4
Total Fees Collect : 11000.0
Pending Fees      : 12000.0

Do You Want To Continue... press y/n
```

- **Get Status Of Student**

```
Enter Your Choice
5
Status Of Student
Enter Student ID
112
-------------------------------------------------
SID     : 112
SName    : Vishal
SEmail   : vishal@gmail.com
SMobile  : 8978676767
CID     : 33
CName    : Python
Cfees    : 3000.0
Total Pay : 1000.0
Fees Pending : 2000.0
-------------------------------------------------

Do You Want To Continue... press y/n
```

- **Update Student**

```
Enter Your Choice
3
Update Student
```

```
----------------------------------------
1.Update Name       2.Update Email
3.Update Mobile     4.Update Course
5.Update Fees
----------------------------------------
Enter You Choice
1
Update Name
Enter Student ID
112
Enter Student Name
Pravin
Student Name Update.
Do You Want To Update More press y/n
```