



VIT-AP

UNIVERSITY

A PROJECT REPORT
FOR
CSE2005
SCOPE
WINTER SEMESTER 2023-24
(Freshers)

Submitted by: Satyajit Panda

Regd: 23BCE7155

Course Code: CSE2005

Course Name: Object Oriented Programming

Slot: C1+TC1

Submitted To: Dr. D Santhadevi

INDEX:

Sno.	Topic
1	Project Introduction
2	What Drivers and JDK Used?
3	Classes, Methods and Objects
4	Databases Used
5	Source Code - Main
6	Source Code - DBconnect
7	Source Code - Admin
8	Output's Screenshot.
9	References Used
10	Conclusion

Acknowledgment:

I would like to express our gratitude to Dr. D Santhadevi Ma'am for her help and support throughout the development of the "Email and Registration Number Administration System" project. Her insights have been invaluable in shaping this system into a functional and efficient tool for managing student registrations and email addresses at our institution.

Sincerely,

Satyajit Panda

23BCE7155

Introduction –

Student admissions and email account management is a substantial undertaking for universities. The Software Development Cell (SDC) Email & Registration Management System alleviates this burden for VIT database admins. This powerful system displays lists of registered students, seamlessly manages new admissions, and efficiently processes registration cancellations. With its robust functionality and intuitive interface, the system streamlines critical administrative tasks.

What drivers and JDK used?

- 1 - JDBC (Java Database Connectivity) drivers are used to establish a connection between the Java application and the SQL database.
- 2 - J-Connector (SQL Server) is a Microsoft-provided driver that allows Java applications to communicate with Microsoft SQL Server databases specifically.
- 3 - The project is built using JDK (Java Development Kit) version 22, which is the most recent version of Java at the time of development.
- 4 - JDK 22 provides the latest Java features, improvements, and bug fixes, ensuring optimal performance and compatibility for the application.

CLASSES:

1. Main Class:

- This class is responsible for the menu window or user interface.
- It serves as the entry point for the application.
- It handles user input and navigation through the menu options.

2. Admin Class:

- This class contains all the core operations and functionalities.
- It provides methods for adding new student records, deleting existing records, and displaying data from the database in a tabular format.
- All the data manipulation operations are implemented in this class.

3. DBconnect Class:

- This class is dedicated to establishing the connection with the MySQL database.
- It encapsulates the logic for connecting to the database using the appropriate JDBC drivers or J-Connector.
- This class acts as an interface between the application and the database, ensuring a separation of concerns.

METHODS:

1. ``Admin`` constructor: Initializes a new ``Admin`` object with the provided name, department, and application number. It also generates a default password.
2. ``getName``: Returns the name of the student.
3. ``getFirstName``: Extracts and returns the first name (lowercase) of the student from their full name.
4. ``getDept``: Returns the department of the student.
5. ``getApplicationNo``: Returns the application number of the student.
6. ``getRegNo``: Generates and returns a unique registration number for the student based on their application number, department, and the college they belong to.
7. ``getEmail``: Generates and returns the student's email address based on their registration number and the college they belong to.

8. ``addStudent`` : Adds a new student record to the corresponding database table based on their department and college.
9. ``getTable`` : Displays all student records from a specified database table in a tabular format.
10. ``getTable`` (overloaded): Displays student records from a specified database table and college in a tabular format.
11. ``deleteStudent`` : Deletes a student record from the corresponding database table based on their registration number.
12. ``updatePassword`` : Updates the password of a student in the corresponding database table based on their registration number and the provided old and new passwords.
13. ``getNoofStudents`` : Retrieves and displays the total number of students, as well as the count of students in the CSE and ECE departments, for a specified college.

OBJECTS –

Admin Class with :

- `DBconnect db = new DBconnect();`

Main Class with :

- 1. `Admin cseVellore;`
- 2. `Admin eceVellore;`
- 3. `Admin cseAP;`
- 4. `Admin eceAP;`

DATABASE USED:

The database named **admin** used in the Admin class. It has 2 tables namely:

```
mysql> show tables;
+-----+
| Tables_in_admin |
+-----+
| cancelled_students |
| csestudents |
| ecestudents |
+-----+
3 rows in set (0.04 sec)
```

1. CSEstudents:

```
mysql> desc CSEstudents;
+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+
| RegNo      | varchar(250)  | NO   | PRI | NULL    |       |
| CollegeID  | int           | NO   |     | NULL    |       |
| ApplicationNo | varchar(200)  | YES  |     | NULL    |       |
| Name       | varchar(210)  | YES  |     | NULL    |       |
| Email      | varchar(230)  | YES  |     | NULL    |       |
| Password   | varchar(250)  | YES  |     | NULL    |       |
+-----+
```

2. ECEstudents:

```
mysql> desc ECEstudents;
+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+
| RegNo      | varchar(250)  | NO   | PRI | NULL    |       |
| CollegeID  | int           | NO   |     | NULL    |       |
| ApplicationNo | varchar(200)  | YES  |     | NULL    |       |
| Name       | varchar(210)  | YES  |     | NULL    |       |
| Email      | varchar(230)  | YES  |     | NULL    |       |
| Password   | varchar(250)  | YES  |     | NULL    |       |
+-----+
6 rows in set (0.00 sec)
```

3. Cancelled_Students:

```
mysql> desc cancelled_students;
+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+
| RegNo      | varchar(250)  | NO   | PRI | NULL    |       |
| CollegeID  | int           | NO   |     | NULL    |       |
| ApplicationNo | varchar(200)  | YES  |     | NULL    |       |
| Name       | varchar(210)  | YES  |     | NULL    |       |
| Email      | varchar(230)  | YES  |     | NULL    |       |
| Password   | varchar(250)  | YES  |     | NULL    |       |
| PhoneNo    | varchar(15)   | YES  |     | NULL    |       |
| ranks      | int           | YES  |     | NULL    |       |
+-----+
8 rows in set (0.04 sec)
```


SOURCE CODE:

Main Class –

```
import java.sql.*;
import java.util.Scanner;

public class Main {
    static Scanner sc = new Scanner(System.in);

    private static void printMenu() {
        System.out.println();
        System.out.println("-----");
        System.out.println();
        System.out.println("1. About the System.");
        System.out.println("2. Show Details of Registered Students.");
        System.out.println("3. Show Detail of a individual Student.");
        System.out.println("4. Number of Students at Campuses.");
        System.out.println("5. Admit a student.");
        System.out.println("6. Cancel the admission.");
        System.out.println("7. Show Past Students.");
        System.out.println("8. Update password of a student.");
        System.out.println("0. Exit");
        System.out.println();
        System.out.println("-----");
        System.out.println();
    }

    private static void aboutSystem() {
        System.out.println("This system assists the SDC admin in managing VIT's student email addresses and registrations. \n" +
            "It provides functionalities such as displaying registered students, \n" +
            "managing student admissions, and handling registration cancellations.\n");
    }

    private static void showRegisteredStudents() {
        try {
            System.out.println("-----");
        };

        System.out.println("Show Details of Registered Students");
        System.out.println("-----");
        ");

        System.out.println("Choose the college to view registered students:");
        System.out.println("1. VIT Vellore");
        System.out.println("2. VIT AP");
        System.out.println("0. Back");
        System.out.print(" >> ");
        int collegeChoice = sc.nextInt();
        System.out.println();

        switch (collegeChoice) {
```

```

        case 1:
            System.out.println("Registered Students at VIT Vellore:");
            System.out.println("-----");
            Admin.getTable("csestudents", 1);
            Admin.getTable("ecestudents", 1);
            break;
        case 2:
            System.out.println("Registered Students at VIT AP:");
            System.out.println("-----");
            Admin.getTable("csestudents", 2);
            Admin.getTable("ecestudents", 2);
            break;
        case 0:
            return;
        default:
            System.out.println("Invalid choice. Please choose again.");
    }
} catch (SQLException e) {
    System.out.println("An error occurred: " + e.getMessage());
}
}

private static void numberOfStudentsAtCampuses() {
    try {
        System.out.println("-----");
        System.out.println("Number of Students at Campuses");
        System.out.println("-----");

        System.out.println("1. VIT Vellore");
        System.out.println("2. VIT AP");
        System.out.println("0. Back");
        System.out.print(" >> ");
        int campusChoice = sc.nextInt();
        System.out.println();

        switch (campusChoice) {
            case 1:
                Admin.getNoofStudents(1);
                break;
            case 2:
                Admin.getNoofStudents(2);
                break;
            case 0:
                return;
            default:
                System.out.println("Invalid choice. Please choose again.");
        }
    } catch (SQLException e) {
        System.out.println("An error occurred: " + e.getMessage());
    }
}

private static void admitStudent() {
    while (true) {

```

```

        System.out.println("Admit a student");

        System.out.print("Application number: ");
        long applicationNo = sc.nextLong();
        System.out.println();

        System.out.print("Rank: ");
        int rank = sc.nextInt();
        System.out.println();

        sc.nextLine();

        System.out.print("Name of student: ");
        String name = sc.nextLine();
        System.out.println();

        System.out.print("Phone Number: ");
        String phoneNumber = sc.nextLine();
        System.out.println();

        if (rank < 10000) {
            System.out.println("You have two college options:");
            System.out.println("1. VIT Vellore");
            System.out.println("2. VIT AP");
            System.out.println("0. Go Back");
            System.out.print("Choose a college (1, 2, or 0 to go back): ");
            int collegeChoice = sc.nextInt();
            System.out.println();

            switch (collegeChoice) {
                case 1:
                    admitStudentToVellore(name, applicationNo, rank,
phoneNumber);
                    System.out.println("The student has been admitted.");
                    return;
                case 2:
                    admitStudentToAP(name, applicationNo, rank, phoneNumber);
                    return;
                case 0:
                    // Go back to main menu
                    return;
                default:
                    System.out.println("Invalid choice. Please try again.");
            }
        } else if (rank >= 10000 && rank < 20000) {
            System.out.println("You have VIT AP as only option:");
            System.out.println("1. VIT AP");
            System.out.println("0. Go back");
            System.out.print("Choose a college (1 or 0 to go back): ");
            int collegeChoice = sc.nextInt();
            System.out.println();
            switch (collegeChoice) {
                case 1:
                    admitStudentToAP(name, applicationNo, rank, phoneNumber);
                    return;

```

```

        case 0:
            return;
    }
} else {
    System.out.println("Sorry, the student is not eligible for
admission.");
    return; // Return to main menu
}
}

private static void admitStudentToVellore(String name, long applicationNo, int
rank, String phn) {
    try {
        while (true) {
            System.out.println("VIT Vellore");
            if (rank <= 5000) {
                System.out.println("You have two options:");
                System.out.println("1. CSE");
                System.out.println("2. ECE");
                System.out.println("0. Go Back");
                System.out.print("Choose a branch (1, 2, or 0 to go back): ");
                int branchChoice = sc.nextInt();
                System.out.println();

                switch (branchChoice) {
                    case 1:
                        System.out.println();
                        Admin cseVellore = new Admin(name, "bce",
applicationNo, phn, rank);
                        cseVellore.addStudent(1, phn, rank);
                        System.out.println();
                        System.out.println("Admitted to CSE in VIT Vellore");
                        return;
                    case 2:
                        System.out.println();
                        Admin eceVellore = new Admin(name, "bee",
applicationNo, phn, rank);
                        eceVellore.addStudent(1, phn, rank);
                        System.out.println();
                        System.out.println("Admitted to ECE in VIT Vellore");
                        return;
                    case 0:
                        // Go back to college selection
                        return;
                    default:
                        System.out.println("Invalid choice. Please try
again.");
                }
            } else if (rank <= 10000) {
                System.out.println();
                System.out.println("You are not eligible for CSE. \n" +
                    "You can try ECE if you want to go back.\n" +
                    "press 0 to go back" +

```

```

        "\nPress 1 for ECE...");
        int choice = sc.nextInt();
        if (choice == 0) {
            return;
        }
        System.out.println();
        Admin eceVellore = new Admin(name, "bee", applicationNo, phn,
rank);

        eceVellore.addStudent(1, phn, rank);
        System.out.println();
        System.out.println("Admitted to ECE in VIT Vellore");
        return; // Go back to admitStudent
    } else {
        System.out.println("Sorry, student is not eligible for
admission to VIT Vellore. ");
        return; // Go back to main menu
    }
}
} catch (SQLException e) {
    System.out.println("An error occurred: " + e.getMessage());
}
}

private static void admitStudentToAP(String name, long applicationNo, int
rank, String phn) {
    try {
        while (true) {
            System.out.println("VIT AP");
            if (rank <= 10000) {
                System.out.println("You have two options:");
                System.out.println("1. CSE");
                System.out.println("2. ECE");
                System.out.println("0. Go Back");
                System.out.print("Choose a branch (1, 2, or 0 to go back): ");
                int branchChoice = sc.nextInt();
                System.out.println();

                switch (branchChoice) {
                    case 1:
                        System.out.println();
                        Admin cseAP = new Admin(name, "bce", applicationNo,
phn, rank);

                        cseAP.addStudent(2, phn, rank);
                        System.out.println();
                        System.out.println("Admitted to CSE in VIT AP");

                        return; // Go back to admitStudent
                    case 2:
                        System.out.println();
                        Admin eceAP = new Admin(name, "bee", applicationNo,
phn, rank);

                        eceAP.addStudent(2, phn, rank);
                        System.out.println();
                        System.out.println("Admitted to ECE in VIT AP");
                        return; // Go back to admitStudent

```

```

        case 0:
            // Go back to college selection
            return;
        default:
            System.out.println("Invalid choice. Please try
again.");
    }
} else if (rank <= 20000) {
    System.out.println();
    System.out.println("If student dont want to participate fo
ECE");

    System.out.println("0. To Go back");
    System.out.println("Press 1 for ECE...");
    int choice = sc.nextInt();
    if (choice == 0) {
        return;
    }
    System.out.println();
    Admin eceAP = new Admin(name, "bee", applicationNo, phn,
rank);

    eceAP.addStudent(2, phn, rank);
    System.out.println();
    System.out.println("Admitted to ECE in VIT AP");
    return; // Go back to admitStudent
} else {
    System.out.println("Sorry, the student is not eligible for
admission to VIT AP.");
    return; // Go back to main menu
}
}
} catch (SQLException e) {
    System.out.println("An error occurred: " + e.getMessage());
}
}

private static void cancelRegistration() {
    try {
        System.out.println("Note down the registration no.**");
        System.out.println("Enter the College :");
        System.out.println("1 for VIT-Vellore");
        System.out.println("2 for VIT-AP");
        int col = sc.nextInt();
        if (col == 1) {
            System.out.println("Enter the branch :");
            System.out.println("1 for CSE");
            System.out.println("2 for ECE");
            int branch = sc.nextInt();
            if (branch == 1) {
                Admin.getTable("csestudents", 1);
            } else if (branch == 2) {
                Admin.getTable("ecestudents", 1);
            }
        } else if (col == 2) {
            System.out.println("Enter the branch :");
            System.out.println("1 for CSE");

```

```

        System.out.println("2 for ECE");
        int branch = sc.nextInt();
        if (branch == 1) {
            Admin.getTable("csestudents", 2);

        } else if (branch == 2) {
            Admin.getTable("ecestudents", 2);
        }

    }

    System.out.println();
    System.out.println("Check the Registration Number of which is to be
cancelled.");
    System.out.println();

    System.out.print("Give the Registration No. >> ");
    String reg = sc.next();
    System.out.println();

    Admin.deleteStudent(reg);

    System.out.println("The admission has been cancelled.");
} catch (SQLException e) {
    System.out.println("An error occurred: " + e.getMessage());
}
}

private static void pastStudents() {
    try {
        Admin.getTable("Cancelled_Students");
    } catch (SQLException e) {
        System.out.println("An error occurred: " + e.getMessage());
    }
}

private static void updatePass() {
    try {
        System.out.println();

        System.out.println("Note down the registration no. and password**");
        System.out.println("Enter the College :");
        System.out.println("1 for VIT-Vellore");
        System.out.println("2 for VIT-AP");
        int col = sc.nextInt();
        if (col == 1) {
            System.out.println("Enter the branch :");
            System.out.println("1 for CSE");
            System.out.println("2 for ECE");
            int branch = sc.nextInt();
            if (branch == 1) {
                Admin.getTable("csestudents", 1);
            } else if (branch == 2) {
                Admin.getTable("ecestudents", 1);
            }
        }
    }
}

```

```

    } else if (col == 2) {
        System.out.println("Enter the branch :");
        System.out.println("1 for CSE");
        System.out.println("2 for ECE");
        int branch = sc.nextInt();
        if (branch == 1) {
            Admin.getTable("csestudents", 2);

        } else if (branch == 2) {
            Admin.getTable("ecestudents", 2);
        }
    }

    System.out.println();
    System.out.print("Enter the registration no : ");
    String reg = sc.next();
    System.out.println();

    System.out.println("Enter the old password : ");
    String oldPass = sc.next();
    System.out.println();
    System.out.println("Enter the new password : ");
    String newPass = sc.next();
    System.out.println();
    Admin.updatePassword(reg, oldPass, newPass);
    System.out.println();
    System.out.println("The Login password has been updated.");
    System.out.println();
} catch (SQLException e) {
    System.out.println("An error occurred: " + e.getMessage());
}
}

private static void showDetails() {
    try {
        while (true) {
            System.out.println("Enter the College:");
            System.out.println("1 for VIT-Vellore");
            System.out.println("2 for VIT-AP");
            int col = sc.nextInt();
            if (col == 1 || col == 2) {
                System.out.println("Enter the branch:");
                System.out.println("1 for CSE");
                System.out.println("2 for ECE");
                int branch = sc.nextInt();
                if (branch == 1 || branch == 2) {
                    String tableName = (branch == 1) ? "csestudents" :
"ecestudents";

                    Admin.getTable(tableName, col);
                    System.out.println();
                    Admin.showStudent(tableName, col);
                    break;
                } else {
                    System.out.println("Invalid branch number! Please enter 1

```



```

        for CSE or 2 for ECE.");
        }
        } else {
            System.out.println("Invalid college number! Please enter 1 for
VIT-Vellore or 2 for VIT-AP.");
        }
    }
} catch (Exception e) {
    System.out.println("Invalid input! Please enter a valid integer.");
    sc.nextLine();
}
}

public static void main(String[] args) {
    System.out.println("VIT Email and Registration Management");
    boolean exit = false;
    while (!exit) {
        printMenu();
        System.out.print(" >> ");
        int choice = sc.nextInt();
        System.out.println();
        switch (choice) {
            case 0:
                System.out.println("Thank you for using our system");
                exit = true;
                break;
            case 1:
                aboutSystem();
                break;
            case 2:
                showRegisteredStudents();
                break;

            case 3:
                showDetails();
                break;
            case 4:
                numberOfStudentsAtCampuses();
                break;
            case 5:
                admitStudent();
                break;
            case 6:
                cancelRegistration();
                break;
            case 7:
                pastStudents();
                break;
            case 8:
                updatePass();
                break;
            default:
                System.out.println("Invalid choice. Please choose again.");
        }
    }
}

```

```
        sc.close(); // Closing the scanner  
    }  
}
```

DBConnect Class –

```
import java.sql.*;

public class DBconnect {
    Connection con = null;
    PreparedStatement pst = null;
    ResultSet rs = null;

    public DBconnect() {
    }

    public Connection connect() {

        try {
            String url = "jdbc:mysql://localhost:3306/admin";
            con = DriverManager.getConnection(url, "root", "abcd35621");
            System.out.println();
            return con;
        } catch (Exception e) {
            System.out.println(e.getMessage());
        }
        return null;
    }
}
```

Admin –

```
import java.sql.*;
import java.util.Scanner;

interface AdminInterface {
    String getName();
    String getDept();
    long getApplicationNo();
    String getRegNo(int college) throws SQLException;
    String getEmail(int college) throws SQLException;
    void addStudent(int college,String phn, int rank) throws SQLException;
}

public class Admin implements AdminInterface {
    static DBconnect db=new DBconnect();
    static Scanner sc=new Scanner(System.in);

    String password;
    String name, dept, regNo;
    long applicationNo;
    String phoneNo;
    int Rank;
    public int getRank() {
        return Rank;
    }

    public String getPhoneNo() {
        return phoneNo;
    }

    public Admin(String name, String dept, long applicationNo, String phone, int Rank){
        this.name = name;
        this.dept =dept;
        this.applicationNo = applicationNo;
        password = getFirstName()+"123";
    }

    public String getName() {
        return name;
    }

    public String getFirstName(){
        String newName = getName()+ ' ';
        return newName.substring(0,newName.indexOf(' ')).toLowerCase();
    }

    public String getDept() {
        return dept;
    }

    public long getApplicationNo() {
        return applicationNo;
    }
}
```

```

    }

    public String getRegNo(int college) throws SQLException {
        Connection conn = db.connect();
        PreparedStatement pst = null;
        ResultSet rs = null;
        String tableName = "";

        if(getDept().equals("bce")){
            tableName = "csestudents";
        }
        else if(getDept().equals("bee")){
            tableName = "ecestudents";
        }

        String query = "SELECT MAX(RegNo) FROM " + tableName + " WHERE CollegeID = ? AND
RegNo IS NOT NULL AND RegNo != ''";
        ;
        pst = conn.prepareStatement(query);

        pst.setInt(1, college);
        rs = pst.executeQuery();

        int id = 0;
        if(rs.next()){
            id = Integer.parseInt((rs.getString(1)).substring(5))+1;
        }

        regNo =
String.valueOf(getApplicationNo()).substring(2,4)+getDept()+String.valueOf(id);

        conn.close();
        return regNo;
    }

    public String getEmail(int college) throws SQLException {

        String id = getRegNo(college);
        String email="";
        if(college==1){
            email = getFirstName()+ "." +id+"@vitstudent.ac.in";
        }
        else if(college==2){
            email = getFirstName()+ "." +id+"@vitapstudent.ac.in";
        }
        return email;
    }

    public void addStudent(int college, String phn, int rank) throws SQLException {
        Connection conn = null;
        PreparedStatement ps = null;

        try {
            conn = db.connect();
            String tableName = "";

            if (getDept().equalsIgnoreCase("bee")) {
                tableName = "ecestudents";
            } else if (getDept().equalsIgnoreCase("bce")) {
                tableName = "csestudents";
            }
        }
    }

```

```

        if (!tableName.isEmpty()) {
            String query = "INSERT INTO " + tableName + " (RegNo, CollegeID,
ApplicationNo, Name, Email, Password, PhoneNo, ranks) VALUES (?, ?, ?, ?, ?, ?, ?, ?)";
            String emailAddress = getEmail(college);

            ps = conn.prepareStatement(query);
            ps.setString(1, regNo.toUpperCase());
            ps.setInt(2, college);
            ps.setString(3, String.valueOf(getApplicationNo()));
            ps.setString(4, getName());
            ps.setString(5, emailAddress);
            ps.setString(6, password);
            ps.setString(7, phn);
            ps.setInt(8, rank);

            ps.executeUpdate();
        }
    } finally {
        if (ps != null) {
            try {
                ps.close();
            } catch (SQLException e) {
                e.printStackTrace();
            }
        }
        if (conn != null) {
            try {
                conn.close();
            } catch (SQLException e) {
                e.printStackTrace();
            }
        }
    }
}

public static void getTable(String tableName) throws SQLException {
    Connection conn = null;
    PreparedStatement ps = null;
    ResultSet rs = null;

    try {
        conn = db.connect();

        // Prepare the SQL query
        String query = "SELECT RegNo, CollegeID, ApplicationNo, Name, Email, Password,
PhoneNo, ranks FROM " + tableName;
        ps = conn.prepareStatement(query);

        // Execute the query
        rs = ps.executeQuery();
        System.out.println();
        // Print table header
        System.out.println("Table: " + tableName);
        System.out.println("-----");
        System.out.println("-----");
        System.out.printf("%-20s %-12s %-20s %-25s %-45s %-15s %-15s %-10s\n", "RegNo",
"CollegeID", "ApplicationNo", "Name", "Email", "Password", "PhoneNo", "Rank");
        System.out.println("-----");
    }
}

```

```

-----");
-----");

    // Print table data
    while (rs.next()) {
        String regNo = rs.getString("RegNo");
        int collegeID = rs.getInt("CollegeID");
        String applicationNo = rs.getString("ApplicationNo");
        String name = rs.getString("Name");
        String email = rs.getString("Email");
        String password = rs.getString("Password");
        String phoneNo = rs.getString("PhoneNo");
        int rank = rs.getInt("ranks");

        System.out.printf("%-20s %-12d %-20s %-25s %-45s %-15s %-15s %-10d\n",
regNo, collegeID, applicationNo, name, email, password, phoneNo, rank);
    }
    System.out.println("-----");
-----");

    System.out.println();

} catch (SQLException e) {
    e.printStackTrace();
} finally {
    if (rs != null) {
        rs.close();
    }
    if (ps != null) {
        ps.close();
    }
    if (conn != null) {
        conn.close();
    }
}
}

public static void getTable(String tableName, int college) throws SQLException {
    Connection conn = null;
    PreparedStatement ps = null;
    ResultSet rs = null;

    try {
        conn = db.connect();

        // Prepare the SQL query
        String query = "SELECT RegNo, ApplicationNo, Name, Email, Password, PhoneNo,
ranks FROM " + tableName + " WHERE CollegeID = ?";
        ps = conn.prepareStatement(query);
        ps.setInt(1, college);

        // Execute the query
        rs = ps.executeQuery();
        System.out.println();
        // Print table header
        if (college == 1) {
            System.out.print("VIT-Vellore - ");
        } else if (college == 2) {
            System.out.print("VIT-AP - ");
        }
    }
}

```

```

        if (tableName.equalsIgnoreCase("ecestudents")) {
            System.out.println("ECE Students");
        } else if (tableName.equalsIgnoreCase("csestudents")) {
            System.out.println("CSE Students");
        }

        System.out.println("\n-----\n");
        System.out.printf("%-25s %-20s %-25s %-45s %-45s %-20s %-8s\n", "Registration
No", "Application No", "Name", "Email", "Password", "Phone No", "Rank");
        System.out.println("\n-----\n");

        // Print table data
        while (rs.next()) {
            String regNo = rs.getString("RegNo");
            String applicationNo = rs.getString("ApplicationNo");
            String name = rs.getString("Name");
            String email = rs.getString("Email");
            String password = rs.getString("Password");
            String phoneNo = rs.getString("PhoneNo");
            int rank = rs.getInt("ranks");

            System.out.printf("%-25s %-20s %-25s %-45s %-45s %-20s %-8d\n", regNo,
applicationNo, name, email, password, phoneNo, rank);
        }
        System.out.println("-----");

        System.out.println();

    } catch (SQLException e) {
        e.printStackTrace();
    } finally {
        // Close resources in reverse order of their creation to avoid resource leaks
        if (rs != null) {
            rs.close();
        }
        if (ps != null) {
            ps.close();
        }
        if (conn != null) {
            conn.close();
        }
    }
}

public static void deleteStudent(String regNo) throws SQLException {
    Connection conn = db.connect();
    PreparedStatement psSelect = null;
    String tableName = "";

    if (regNo.substring(2, 5).equalsIgnoreCase("bce")) {
        tableName = "csestudents";
    } else if (regNo.substring(2, 5).equalsIgnoreCase("bee")) {
        tableName = "ecestudents";
    }
}

```



```

String query = "SELECT * FROM " + tableName + " WHERE RegNo = ?";
psSelect = conn.prepareStatement(query);
psSelect.setString(1, regNo);
ResultSet rs = psSelect.executeQuery();

String reg = "", app = "", name1="", email1="", password1="", phn = "";
int ranks = 0;
int col = 0;
if (rs.next()) {
    reg = rs.getString("RegNo");
    app = rs.getString("ApplicationNo");
    name1 = rs.getString("Name");
    email1 = rs.getString("Email");
    password1 = rs.getString("Password");
    col = rs.getInt("CollegeID");
    ranks =rs.getInt("ranks");
    phn = rs.getString("PhoneNo");
}

try {
    String insertQuery = "INSERT INTO Cancelled_Students VALUES
(?,?,?,?,?,?,?,?)";
    PreparedStatement psInsert = conn.prepareStatement(insertQuery);
    psInsert.setString(1, reg);
    psInsert.setInt(2, col);
    psInsert.setString(3, app);
    psInsert.setString(4, name1);
    psInsert.setString(5, email1);
    psInsert.setString(6, password1);
    psInsert.setString(7, phn);
    psInsert.setInt(8, ranks);
    psInsert.executeUpdate();
    psInsert.close();
} catch (SQLException e) {
    // If insertion into Cancelled_Students fails, revert changes
    String newDel = "DELETE FROM Cancelled_Students WHERE RegNo = ?";
    PreparedStatement psDel = conn.prepareStatement(newDel);
    psDel.setString(1, reg);
    psDel.executeUpdate();
    psDel.close();

    String newInsertQuery = "INSERT INTO Cancelled_Students VALUES
(?,?,?,?,?,?,?,?)";
    PreparedStatement psInsert = conn.prepareStatement(newInsertQuery);
    psInsert.setString(1, reg);
    psInsert.setInt(2, col);
    psInsert.setString(3, app);
    psInsert.setString(4, name1);
    psInsert.setString(5, email1);
    psInsert.setString(6, password1);
    psInsert.setString(7, phn);
    psInsert.setInt(8, ranks);
    psInsert.executeUpdate();
    psInsert.close();
}

PreparedStatement psDelete = conn.prepareStatement("DELETE FROM " + tableName + "
WHERE RegNo = ?");
psDelete.setString(1, regNo);
psDelete.executeUpdate();

```

```

        psDelete.close();

        rs.close();
        psSelect.close();
        conn.close();
    }

    public static void updatePassword(String regNo, String oldPassword, String newPassword)
    throws SQLException {
        Connection conn = db.connect();
        String tableName = "";
        if (regNo.substring(2, 5).equalsIgnoreCase("bce")) {
            tableName = "csestudents";
        }
        else if (regNo.substring(2, 5).equalsIgnoreCase("bee")) {
            tableName = "ecestudents";
        }

        // Check old password
        String selectQuery = "SELECT Password FROM " + tableName + " WHERE RegNo = ?";
        PreparedStatement psSelect = conn.prepareStatement(selectQuery);
        psSelect.setString(1, regNo);
        ResultSet rs = psSelect.executeQuery();
        if (rs.next()) {
            String savedPassword = rs.getString("Password");
            if (!savedPassword.equals(oldPassword)) {
                System.out.println("Old password is incorrect.");
                return;
            }
        }
        else {
            System.out.println("Student with registration number " + regNo + " not
found.");
            return;
        }

        String updateQuery = "UPDATE " + tableName + " SET Password = ? WHERE RegNo = ?";
        PreparedStatement psUpdate = conn.prepareStatement(updateQuery);
        psUpdate.setString(1, newPassword); // Set the new password
        psUpdate.setString(2, regNo); // Set the registration number to identify the
student
        psUpdate.executeUpdate();
        psUpdate.close();
        psSelect.close();
        rs.close();
    }

    public static void getNoofStudents(int collegeID) throws SQLException {

        Connection conn = null;
        PreparedStatement psTotal = null;
        PreparedStatement psCSE = null;
        PreparedStatement psECE = null;
        ResultSet rsTotal = null;
        ResultSet rsCSE = null;
        ResultSet rsECE = null;

        try {
            conn = db.connect();

```

```

        if(collegeID==1){
            System.out.println("Students at VIT-Vellore");
        }
        else
            if(collegeID==2){
                System.out.println("Students at VIT-AP");
            }
        // Query to get the total number of students in the specified college
        String queryTotal = "SELECT (SELECT COUNT(*) FROM csestudents WHERE CollegeID =
?) + " +
            "(SELECT COUNT(*) FROM ecestudents WHERE CollegeID = ?) AS
TotalStudents";
        psTotal = conn.prepareStatement(queryTotal);
        psTotal.setInt(1, collegeID);
        psTotal.setInt(2, collegeID);
        rsTotal = psTotal.executeQuery();

        int totalStudents = 0;
        if (rsTotal.next()) {
            totalStudents = rsTotal.getInt("TotalStudents");
        }

        // Query to get the number of CSE students in the specified college
        String queryCSE = "SELECT COUNT(*) AS CSECount FROM csestudents WHERE CollegeID
= ?";

        psCSE = conn.prepareStatement(queryCSE);
        psCSE.setInt(1, collegeID);
        rsCSE = psCSE.executeQuery();

        int cseCount = 0;
        if (rsCSE.next()) {
            cseCount = rsCSE.getInt("CSECount");
        }

        // Query to get the number of ECE students in the specified college
        String queryECE = "SELECT COUNT(*) AS ECECount FROM ecestudents WHERE CollegeID
= ?";

        psECE = conn.prepareStatement(queryECE);
        psECE.setInt(1, collegeID);
        rsECE = psECE.executeQuery();

        int eceCount = 0;
        if (rsECE.next()) {
            eceCount = rsECE.getInt("ECECount");
        }

        // Print the results
        System.out.println("Total Students: " + totalStudents);
        System.out.println("Branch | Student Count");
        System.out.println("-----");
        System.out.println("CSE | " + cseCount);
        System.out.println("ECE | " + eceCount);
    } finally {
        // Close resources
        if (rsTotal != null) {
            rsTotal.close();
        }
        if (psTotal != null) {
            psTotal.close();
        }
    }

```

```

        if (rsCSE != null) {
            rsCSE.close();
        }
        if (psCSE != null) {
            psCSE.close();
        }
        if (rsECE != null) {
            rsECE.close();
        }
        if (psECE != null) {
            psECE.close();
        }
        if (conn != null) {
            conn.close();
        }
    }
}

private static boolean isRegNoPresent(String tableName, String regNo) throws
SQLException {
    Connection conn = db.connect();
    String query = "SELECT RegNo FROM " + tableName + " WHERE RegNo = ?";
    try (PreparedStatement statement = conn.prepareStatement(query)) {
        statement.setString(1, regNo);
        try (ResultSet resultSet = statement.executeQuery()) {
            return resultSet.next(); // Returns true if the registration number is
present
        }
    }
}

public static void showStudent(String tableName, int collegeID) {
    while (true) {
        try {
            System.out.println("Enter the registration number:");
            String regno = sc.next();
            if (isRegNoPresent(tableName, regno)) {
                Connection conn = db.connect();
                PreparedStatement psTotal = null;

                String query = "SELECT * FROM " + tableName + " WHERE CollegeID = ? AND
RegNo = ?";

                psTotal = conn.prepareStatement(query);
                psTotal.setInt(1, collegeID);
                psTotal.setString(2, regno);
                ResultSet rsTotal = psTotal.executeQuery();

                if (rsTotal.next()) {
                    String regNo = rsTotal.getString("RegNo");
                    int col = rsTotal.getInt("CollegeID");
                    String app = rsTotal.getString("ApplicationNo");
                    String name = rsTotal.getString("Name");
                    String email = rsTotal.getString("Email");
                    String pass = rsTotal.getString("Password");
                    String phone = rsTotal.getString("PhoneNo");
                    int rank = rsTotal.getInt("ranks");
                    String d="";
                    if (email.contains("bce")){
                        d = "Computer Science and Engineering.";
                    }
                    else

```


Output:

1. About the system

```
VIT Email and Registration Management
-----
1. About the System.
2. Show Details of Registered Students.
3. Show Detail of a individual Student.
4. Number of Students at Campuses.
5. Admit a student.
6. Cancel the admission.
7. Show Past Students.
8. Update password of a student.
0. Exit
-----
>> 1
```

2. Registered Student List.

```
-----
1. About the System.
2. Show Details of Registered Students.
3. Show Detail of a individual Student.
4. Number of Students at Campuses.
5. Admit a student.
6. Cancel the admission.
7. Show Past Students.
8. Update password of a student.
0. Exit
-----
>> 2
-----
Show Details of Registered Students
-----
Choose the college to view registered students:
1. VIT Vellore
2. VIT AP
0. Back
>> 2
```

Registered Students at VIT AP:						

VIT-AP - CSE Students						

Registration No	Application No	Name	Email	Password	Phone No	Rank
23BCE7000	2023170010	Aarav Patel	aarav.23bce7000@vitapstudent.ac.in	Aarav123	9747916769	5176
23BCE7001	2023170110	Ravi Singla	ravi.23bce7001@vitapstudent.ac.in	Ravi123	9621040960	5094
23BCE7002	2023170112	Vivek Reddy	vivek.23bce7002@vitapstudent.ac.in	Vivek123	9876543220	5943
23BCE7003	2023170113	Sneha Gupta	sneha.23bce7003@vitapstudent.ac.in	Sneha123	9876543221	4430
23BCE7004	2023170114	Kiran Kumar	kiran.23bce7004@vitapstudent.ac.in	Kiran123	9876543222	4326
23BCE7005	2023170115	Neha Singh	neha.23bce7005@vitapstudent.ac.in	Neha123	9876543223	4338
23BCE7006	2023170116	Rahul Reddy	rahul.23bce7006@vitapstudent.ac.in	Rahul123	9876543224	4713

VIT-AP - ECE Students						

Registration No	Application No	Name	Email	Password	Phone No	Rank
23BEE7001	202345829	Saptarishi Patnaik	saptarishi.23bee7003@vitapstudent.ac.in	saptarishi123	9235467702	5890
23BEE7002	2023452121	Sankhojyoti Halder	sankhojyoti.23bee7004@vitapstudent.ac.in	sankhojyoti123	9183574618	5920
23BEE7003	2023450001	Aruna Menon	aruna.23bee7003@vitapstudent.ac.in	aruna123	9876543210	897
23BEE7004	2023450002	Ashwin Nair	ashwin.23bee7004@vitapstudent.ac.in	ashwin123	9876543211	987
23BEE7005	2023450003	Deepika Iyer	deepika.23bee7005@vitapstudent.ac.in	deepika123	9876543212	2141
23BEE7006	2023450004	Gopal Kumar	gopal.23bee7006@vitapstudent.ac.in	gopal123	9876543213	1844
23BEE7007	2023450005	Indira Patel	indira.23bee7007@vitapstudent.ac.in	indira123	9876543214	2697

3. Detail of individual student (New Implementation)

1. About the System. 2. Show Details of Registered Students. 3. Show Detail of a individual Student. 4. Number of Students at Campuses. 5. Admit a student. 6. Cancel the admission. 7. Show Past Students. 8. Update password of a student. 9. Exit						

>> 1						
Enter the College:						
1 for VIT-Vellore						
2 for VIT-AP						
Enter the branch:						
1 for CSE						
2 for ECE						
1						
VIT-Vellore - CSE Students						

Registration No	Application No	Name	Email	Password	Phone No	Rank
23BCE1000	202317001	Alice John	alice.23bce1000@vitstudent.ac.in	Alice123	9755155308	203
23BCE1001	202317002	Bob Melquen	bob.23bce1001@vitstudent.ac.in	Bob123	9594329373	325
23BCE1002	202317003	Charlie Johnson	charlie.23bce1002@vitstudent.ac.in	Charlie123	9706180971	215
23BCE1003	202317004	Aditya Kumar	aditya.23bce1003@vitstudent.ac.in	Aditya123	9876543210	418
23BCE1004	202317005	Neha Sharma	neha.23bce1004@vitstudent.ac.in	Neha123	9876543211	501
23BCE1005	202317006	Rahul Gupta	rahul.23bce1005@vitstudent.ac.in	Rahul123	9876543212	333
23BCE1006	202317007	Priya Singh	priya.23bce1006@vitstudent.ac.in	Priya123	9876543213	265
23BCE1007	202317008	Amit Patel	amit.23bce1007@vitstudent.ac.in	Amit123	9876543214	419

```

Enter the registration number:
23BCE1007

Here are the details:
-----

Registration No: 23BCE1007
College: VIT Vellore
Application No: 202317008
Rank: 419
Name: Amit Patel
Branch : Computer Science and Engineering.
Email: amit.23bce1007@vitstudent.ac.in
Password: Amit123
Phone No: 9876543214
-----

```

4. Number of students at campus.

```

-----
1. About the System.
2. Show Details of Registered Students.
3. Show Detail of a individual Student.
4. Number of Students at Campuses.
5. Admit a student.
6. Cancel the admission.
7. Show Past Students.
8. Update password of a student.
0. Exit
-----

```

```
>> 4
```

```

-----
Number of Students at Campuses
-----

```

```

1. VIT Vellore
2. VIT AP
0. Back
>> 2

```

```

Students at VIT-AP
Total Students: 14
Branch | Student Count
-----
CSE    | 7
ECE    | 7

```


5. Admitting a new student.

```
5. Admit a student.
6. Cancel the admission.
7. Show Past Students.
8. Update password of a student.
0. Exit

-----

>> 5

Admit a student
Application number: 2023584654

Rank: 369

Name of student: Satyajit Panda

Phone Number: 9876326313

You have two college options:
1. VIT Vellore
2. VIT AP
0. Go Back
Choose a college (1, 2, or 0 to go back): 1

VIT Vellore
You have two options:
1. CSE
2. ECE
0. Go Back
Choose a branch (1, 2, or 0 to go back): 1

Admitted to CSE in VIT Vellore
The student has been admitted.
```

After addition.

```
mysql> SELECT * FROM cstudents WHERE CollegeID = 1;
```

RegNo	CollegeID	ApplicationNo	Name	Email	Password	PhoneNo	rank
23BCE1000	1	202317001	Alice John	alice.23bce1000@vitstudent.ac.in	Alice123	9755155308	203
23BCE1001	1	202317002	Bob Melquer	bob.23bce1001@vitstudent.ac.in	Bob123	9594329273	325
23BCE1002	1	202317003	Charlie Johnson	charlie.23bce1002@vitstudent.ac.in	Charlie123	9706180971	215
23BCE1003	1	202317004	Aditya Kumar	aditya.23bce1003@vitstudent.ac.in	Aditya123	9876543210	418
23BCE1004	1	202317005	Neha Sharma	neha.23bce1004@vitstudent.ac.in	Neha123	9876543211	501
23BCE1005	1	202317006	Rahul Gupta	rahul.23bce1005@vitstudent.ac.in	Rahul123	9876543212	333
23BCE1006	1	202317007	Priya Singh	priya.23bce1006@vitstudent.ac.in	Priya123	9876543213	265
23BCE1007	1	202317008	Amit Patel	amit.23bce1007@vitstudent.ac.in	Amit123	9876543214	419
23BCE1008	1	2023584654	Satyajit Panda	satyajit.23bce1008@vitstudent.ac.in	satyajit123	9876326313	369

9 rows in set (0.00 sec)

```
mysql> |
```

6. Cancelling a student's registration (if the person is willing to withdraw his/her admission)

```

6. Cancel the admission.
7. Show Past Students.
8. Update password of a student.
0. Exit

>> 6

Note down the registration no.**
Enter the College :
1 for VIT-Vellore
2 for VIT-AP
2
Enter the branch :
1 for CSE
2 for ECE
2

VIT-AP - ECE Students
-----
Registration No      Application No      Name      Email      Password      Phone No      Rank
-----
23BEE7001            202345029          Saptarishi Patnaik      saptarishi.23bee7003@vitapstudent.ac.in      saptarishi123      9235467702      5890
23BEE7002            2023452121         Sankhojyoti Halder      sankhojyoti.23bee7004@vitapstudent.ac.in      sankhojyoti123      9183574618      5920
23BEE7003            2023450001         Aruna Menon      aruna.23bee7003@vitapstudent.ac.in      aruna123      9876543210      897
23BEE7004            2023450002         Ashwin Nair      ashwin.23bee7004@vitapstudent.ac.in      ashwin123      9876543211      987
23BEE7005            2023450003         Deepika Iyer      deepika.23bee7005@vitapstudent.ac.in      deepika123      9876543212      2141
23BEE7006            2023450004         Gopal Kumar      gopal.23bee7006@vitapstudent.ac.in      gopal123      9876543213      1844
23BEE7007            2023450005         Indira Patel      indira.23bee7007@vitapstudent.ac.in      indira123      9876543214      2697

Check the Registration Number of which is to be cancelled.
Give the Registration No. >> 23BEE7007

The admission has been cancelled.

```

After withdrawal

```
mysql> SELECT * FROM ecestudents WHERE CollegeID = 2;
```

RegNo	CollegeID	ApplicationNo	Name	Email	Password	PhoneNo	rank
23BEE7001	2	202345829	Saptarishi Patnaik	saptarishi.23bee7003@vitapstudent.ac.in	saptarishi123	9235467702	5890
23BEE7002	2	2023452121	Sankhojyoti Halder	sankhojyoti.23bee7004@vitapstudent.ac.in	sankhojyoti123	9183574618	5920
23BEE7003	2	2023450001	Aruna Menon	aruna.23bee7003@vitapstudent.ac.in	aruna123	9876543210	897
23BEE7004	2	2023450002	Ashwin Nair	ashwin.23bee7004@vitapstudent.ac.in	ashwin123	9876543211	987
23BEE7005	2	2023450003	Deepika Iyer	deepika.23bee7005@vitapstudent.ac.in	deepika123	9876543212	2141
23BEE7006	2	2023450004	Gopal Kumar	gopal.23bee7006@vitapstudent.ac.in	gopal123	9876543213	1844

6 rows in set (0.00 sec)

The withdrawals will be transferred to another table : cancelled_students.

7. To check past student/withdrawn students records

```
-----
1. About the System.
2. Show Details of Registered Students.
3. Show Detail of a individual Student.
4. Number of Students at Campuses.
5. Admit a student.
6. Cancel the admission.
7. Show Past Students.
8. Update password of a student.
9. Exit

>> 2

Table: Cancelled_Students
-----
RegNo      CollegeID  ApplicationNo  Name      Email      Password      PhoneNo      Rank
-----
23BCE1003   1         20232489479   Ariyan Rout  ariyan.23bce1003@vitstudent.ac.in  ariyanrout123  9368708715  0
23BCE1004   1         2023214059   Veer Pratap Singh  veer.23bce1004@vitstudent.ac.in  veer123  9394213438  1234
23BEE7003   2         2023595684   Shreesh Sanyal  shreesh.23bee7003@vitapstudent.ac.in  shreesh123  9394213438  4493
23BEE7007   2         2023450005   Indira Patel  indira.23bee7007@vitapstudent.ac.in  indira123  9876543214  2697
-----
```

8. To Update password (like provided in vtop)

```
8. Update password of a student.
9. Exit

>> 8

Note down the registration no. and password**
Enter the College :
1 for VIT-Vellore
2 for VIT-AP

Enter the branch :
1 for CSE
2 for ECE

VIT-Vellore - CSE Students
-----
Registration No      Application No      Name      Email      Password      Phone No      Rank
-----
23BCE1000            202317001          Alice John  alice.23bce1000@vitstudent.ac.in  Alice123  9755155388  203
23BCE1001            202317002          Bob Melquer  bob.23bce1001@vitstudent.ac.in  Bob123  9594329393  325
23BCE1002            202317003          Charlie Johnson  charlie.23bce1002@vitstudent.ac.in  Charlie123  9786188971  215
23BCE1003            202317004          Aditya Kumar  aditya.23bce1003@vitstudent.ac.in  Aditya123  9876543210  418
23BCE1004            202317005          Neha Sharma  neha.23bce1004@vitstudent.ac.in  Neha123  9876543211  501
23BCE1005            202317006          Rahul Gupta  rahul.23bce1005@vitstudent.ac.in  Rahul123  9876543212  333
23BCE1006            202317007          Priya Singh  priya.23bce1006@vitstudent.ac.in  Priya123  9876543213  265
23BCE1007            202317008          Amit Patel  amit.23bce1007@vitstudent.ac.in  Amit123  9876543214  419
23BCE1008            2023584654          Satyajit Panda  satyajit.23bce1008@vitstudent.ac.in  satyajit123  9876326313  369
-----
```

Let me change the password of Satyajit Panda

I will note his Registration No

```
Enter the registration no : 23BCE1008

Enter the old password :
satyajit123

Enter the new password :
satya1234567

The Login password has been updated.

-----

1. About the System.
2. Show Details of Registered Students.
3. Show Detail of a individual Student.
4. Number of Students at Campuses.
5. Admit a student.
6. Cancel the admission.
7. Show Past Students.
8. Update password of a student.
9. Exit

>> 0
```

Password updation:

```
mysql> SELECT * FROM csestudents WHERE CollegeID = 1;
```

RegNo	CollegeID	ApplicationNo	Name	Email	Password	PhoneNo	rank
23BCE1000	1	202317001	Alice John	alice.23bce1000@vitstudent.ac.in	Alice123	9755155308	203
23BCE1001	1	202317002	Bob Melquer	bob.23bce1001@vitstudent.ac.in	Bob123	9594329373	325
23BCE1002	1	202317003	Charlie Johnson	charlie.23bce1002@vitstudent.ac.in	Charlie123	9706180971	215
23BCE1003	1	202317004	Aditya Kumar	aditya.23bce1003@vitstudent.ac.in	Aditya123	9876543210	418
23BCE1004	1	202317005	Neha Sharma	neha.23bce1004@vitstudent.ac.in	Neha123	9876543211	501
23BCE1005	1	202317006	Rahul Gupta	rahul.23bce1005@vitstudent.ac.in	Rahul123	9876543212	333
23BCE1006	1	202317007	Priya Singh	priya.23bce1006@vitstudent.ac.in	Priya123	9876543213	265
23BCE1007	1	202317008	Amit Patel	amit.23bce1007@vitstudent.ac.in	Amit123	9876543214	419
23BCE1008	1	2023584654	Satyajit Panda	satyajit.23bce1008@vitstudent.ac.in	satya1234567	9876326313	369

9 rows in set (0.00 sec)

-----X-----X-----

After the feedback, further implementation:

1. Added a Individual Student Detail In Menu
To get name, registration number, rank and all other important details regarding the student.
2. Added Phone Numbers and Rank Associated with them.

References Used:

- Geeks for Geeks – JDBC Tutorial:
<https://www.geeksforgeeks.org/jdbc-tutorial/>
- Geeks For Geeks – mySQL
<https://www.geeksforgeeks.org/mysql-tutorial/>
- Javatpoint – OOPs
<https://www.javatpoint.com/java-oops-concepts>
- Java main concepts
Youtube, Class Notes
<https://math.hws.edu/javanotes/index.html>

Conclusion:

In conclusion, the "Email and Registration Number Administration System" project provides a user-friendly solution for managing student registrations and email addresses at our institution. With its efficient interface, robust security measures, and flexibility to adapt to our institution's needs, the system ensures smooth administrative processes.

Satyajit Panda