FSDM – JAVA SE 02 - PROJECT – 2022W

Student ID: C0871405

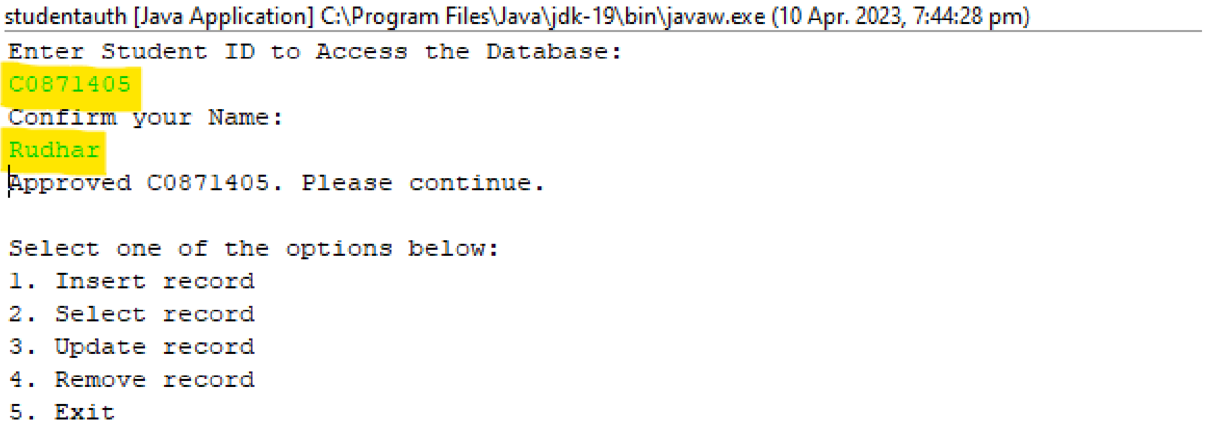
Student Name: Rudhar Sharma

Code:

```
package project;
import java.sql.*;
import java.util.Scanner;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.sql.Statement;
import java.sql.ResultSet;


public class studentauth {

    // credentials for database connection
    private static final String URL = "jdbc:mysql://localhost:3306/students";
    private static final String USERNAME = "root";
    private static final String PASSWORD = "SYSTEM123@";

    // database connection and statement objects
    private static Connection connection;
    private static Statement statement;
    private static ResultSet resultSet;

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        try {
            // establish database connection
            connection = DriverManager.getConnection(URL, USERNAME, PASSWORD);
            statement = connection.createStatement();

            // validate user credentials
            System.out.println("Enter Student ID to Access the Database: ");
            String studentID = scanner.nextLine();
            System.out.println("Confirm your Name: ");
            String name = scanner.nextLine();

            if (!validateUser(studentID, name)) {
                System.out.println("Credentials not valid. Please try again.");
                System.exit(0);
            }

            System.out.println("Approved " + studentID + ". Please continue.");

            // display menu
            while (true) {
                System.out.println("\nSelect one of the options below:\n1. Insert 
record\n2. Select record \n3. Update record\n4. Remove record\n5. Exit");
                int choice = scanner.nextInt();
                scanner.nextLine(); // consume newline character

                switch (choice) {
                    case 1:
                        insert_new_data(scanner);
                        break;
                    case 2:
                        view_record(scanner);
                        break;
                    case 3:
                        update_student_data(scanner);
                        break;
                    case 4:
                           remove_student_data(scanner);
                        break;
                    case 5:
                        System.out.println("Exiting the program.");
                        System.exit(0);
                    default:
                        System.out.println("Invalid choice.");
                }
            }
        } catch (SQLException e) {
            e.printStackTrace();
        } finally {
            try {
                if (resultSet != null) resultSet.close();
                if (statement != null) statement.close();
                if (connection != null) connection.close();
            } catch (SQLException e) {
                e.printStackTrace();
            }
        }
    }


    private static boolean validateUser(String studentID, String name) throws
SQLException {
        String sql = "SELECT COUNT(*) AS count FROM login_info WHERE stu_id='" +
studentID + "' AND stu_name='" + name + "'";
        resultSet = statement.executeQuery(sql);
        resultSet.next();
        int count = resultSet.getInt("count");
        return count > 0;
    }




    private static void insert_new_data(Scanner scanner) throws SQLException {
        System.out.println("Enter student ID:");
        String studentID = scanner.nextLine();
        System.out.println("Enter student name:");
        String studentName = scanner.nextLine();
        System.out.println("Enter date of birth (yyyy-mm-dd):");
        String dob = scanner.nextLine();
        System.out.println("Enter phone number:");
        String phone = scanner.nextLine();
        System.out.println("Enter guardian name:");
        String guardianName = scanner.nextLine();
        System.out.println("Enter course:");
        String course = scanner.nextLine();

        String sql = "INSERT INTO stu_info (stu_id, stu_name, dob, phone,
guardian_name, course) VALUES (?, ?, ?, ?, ?, ?)";
        PreparedStatement statement = connection.prepareStatement(sql);
        statement.setString(1, studentID);
        statement.setString(2, studentName);
        statement.setString(3, dob);
        statement.setString(4, phone);
        statement.setString(5, guardianName);
        statement.setString(6, course);

        int rowsInserted = statement.executeUpdate();
        if (rowsInserted > 0) {
            System.out.println("Record inserted successfully.");
        } else {
            System.out.println("Failed to insert record.");
        }
    }




    private static void view_record(Scanner scanner) throws SQLException {
        System.out.println("Input student ID:");
        String studentID = scanner.nextLine();

        String sql = "SELECT * FROM stu_info WHERE stu_id='" + studentID + "'";
        ResultSet result = statement.executeQuery(sql);

        if (result.next()) {
            System.out.println("Student ID: " + result.getString("stu_id"));
            System.out.println("Name: " + result.getString("stu_name"));
            System.out.println("DOB: " + result.getString("dob"));
            System.out.println("Mobile Number: " + result.getString("phone"));
            System.out.println("Guardian's Name: " +
result.getString("guardian_name"));
            System.out.println("Course Enrolled: " + result.getString("course"));
        } else {
            System.out.println("No record found for student ID " + studentID);
        }
    }



    private static void update_student_data(Scanner scanner) throws SQLException {
        System.out.println("Input student ID:");
        String studentID = scanner.nextLine();

        // Prepare the update statement
        String sql = "UPDATE stu_info SET stu_name=?, dob=?, phone=?,
guardian_name=?, course=? WHERE stu_id=?";
        PreparedStatement statement = connection.prepareStatement(sql);

        // Get the record to update
        sql = "SELECT * FROM stu_info WHERE stu_id=?";
        PreparedStatement selectStatement = connection.prepareStatement(sql);
        selectStatement.setString(1, studentID);
        ResultSet result = selectStatement.executeQuery();

        if (result.next()) {
            // Get new values from user
            System.out.println("Enter new student name (Skip - Enter):");
            String studentName = scanner.nextLine().trim();
            if (!studentName.isEmpty()) {
                statement.setString(1, studentName);
            } else {
                statement.setString(1, result.getString("stu_name"));
            }

            System.out.println("Enter new Birth Date (Skip - Enter):");
            String dob = scanner.nextLine().trim();
            if (!dob.isEmpty()) {
                statement.setString(2, dob);
            } else {
                statement.setString(2, result.getString("dob"));
            }

            System.out.println("Enter new phone number (Skip - Enter):");
            String phone = scanner.nextLine().trim();
            if (!phone.isEmpty()) {
                statement.setString(3, phone);
            } else {
                statement.setString(3, result.getString("phone"));
            }

            System.out.println("Enter guardian name (Skip - Enter):");
            String guardianName = scanner.nextLine().trim();
            if (!guardianName.isEmpty()) {
                statement.setString(4, guardianName);
            } else {
                statement.setString(4, result.getString("guardian_name"));
            }
            System.out.println("Enter new course (Skip - Enter):");
            String course = scanner.nextLine().trim();
            if (!course.isEmpty()) {
                statement.setString(5, course);
            } else {
                statement.setString(5, result.getString("course"));
            }
            statement.setString(6, studentID);
            statement.executeUpdate();
            System.out.println("Record Updated.");

        } else {
            System.out.print("Record not found");
        }
    }



    public static void remove_student_data(Scanner scanner) throws SQLException {
        System.out.println("Enter student ID:");
        String studentID = scanner.nextLine();

        // Prepare the delete statement
        String sql = "DELETE FROM stu_info WHERE stu_id=?";
        PreparedStatement statement = connection.prepareStatement(sql);
        statement.setString(1, studentID);

        // Execute the delete statement
        int rowsDeleted = statement.executeUpdate();

        if (rowsDeleted > 0) {
            System.out.println("Record deleted successfully.");
        } else {
            System.out.println("Record not found.");
        }
    }
}
```

Output:

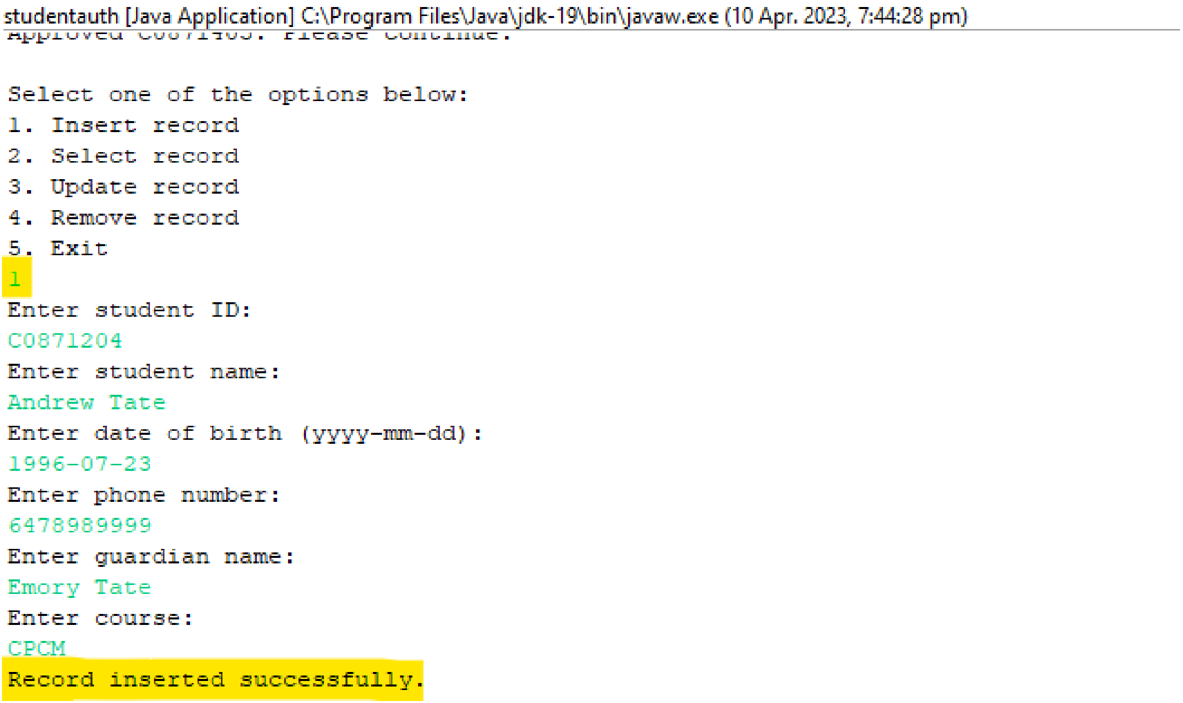
1. Accessing the database



studentauth [Java Application] C:\Program Files\Java\jdk-19\bin\javaw.exe (10 Apr. 2023, 7:44:28 pm)

```
Enter Student ID to Access the Database:
C0871405
Confirm your Name:
Rudhar
Approved C0871405. Please continue.

Select one of the options below:
1. Insert record
2. Select record
3. Update record
4. Remove record
5. Exit
```

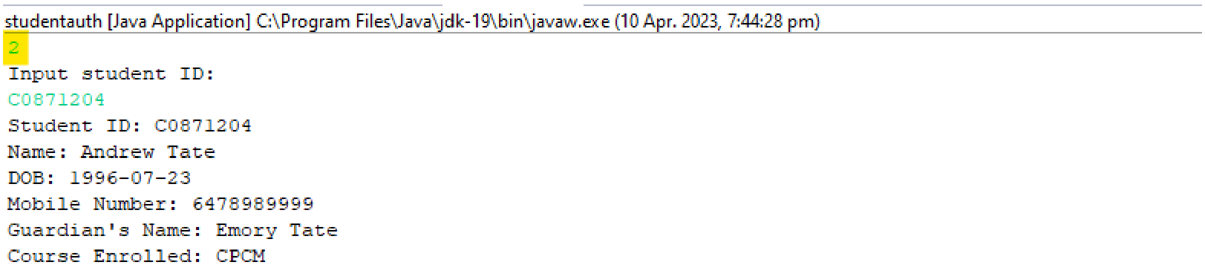
2. Inserting a record



studentauth [Java Application] C:\Program Files\Java\jdk-19\bin\javaw.exe (10 Apr. 2023, 7:44:28 pm)

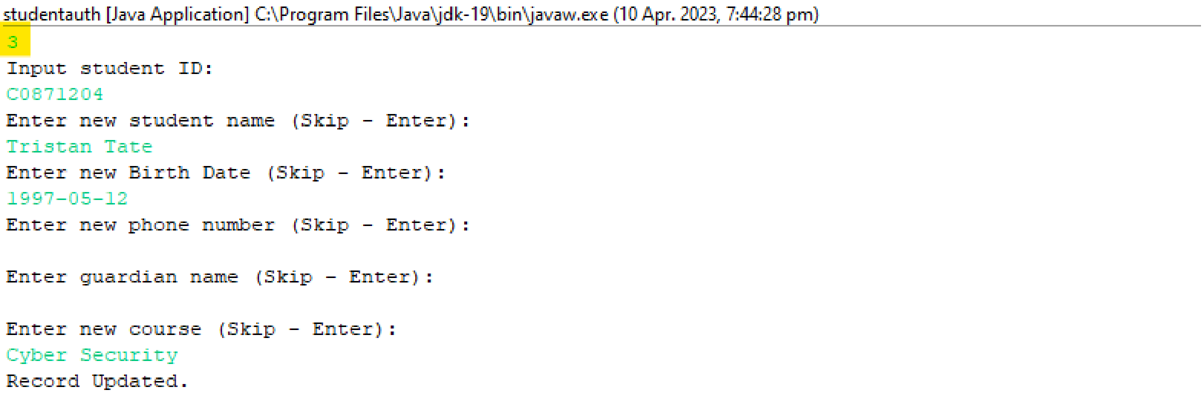
```
Approved C0871405. Please continue.

Select one of the options below:
1. Insert record
2. Select record
3. Update record
4. Remove record
5. Exit
1
Enter student ID:
C0871204
Enter student name:
Andrew Tate
Enter date of birth (yyyy-mm-dd):
1996-07-23
Enter phone number:
6478989999
Enter guardian name:
Emory Tate
Enter course:
CPCM
Record inserted successfully.
```

3. Viewing a Record



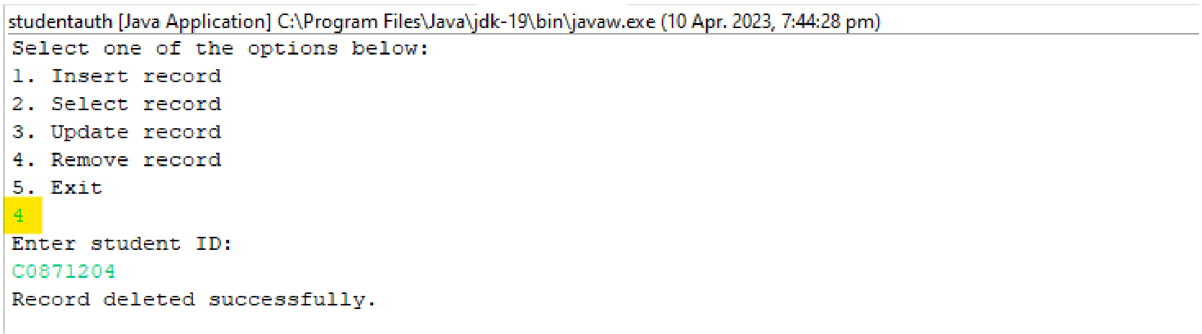
```
studentauth [Java Application] C:\Program Files\Java\jdk-19\bin\javaw.exe (10 Apr. 2023, 7:44:28 pm)
2
Input student ID:
C0871204
Student ID: C0871204
Name: Andrew Tate
DOB: 1996-07-23
Mobile Number: 6478989999
Guardian's Name: Emory Tate
Course Enrolled: CPCM
```

4. Updating the Record



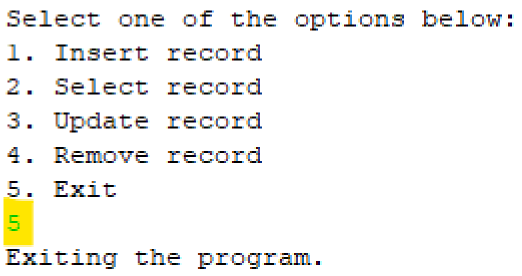
```
studentauth [Java Application] C:\Program Files\Java\jdk-19\bin\javaw.exe (10 Apr. 2023, 7:44:28 pm)
3
Input student ID:
C0871204
Enter new student name (Skip - Enter):
Tristan Tate
Enter new Birth Date (Skip - Enter):
1997-05-12
Enter new phone number (Skip - Enter):

Enter guardian name (Skip - Enter):

Enter new course (Skip - Enter):
Cyber Security
Record Updated.
```

5. Removing a Record



```
studentauth [Java Application] C:\Program Files\Java\jdk-19\bin\javaw.exe (10 Apr. 2023, 7:44:28 pm)
Select one of the options below:
1. Insert record
2. Select record
3. Update record
4. Remove record
5. Exit
4
Enter student ID:
C0871204
Record deleted successfully.
```

6. Exiting the database



```
Select one of the options below:
1. Insert record
2. Select record
3. Update record
4. Remove record
5. Exit
5
Exiting the program.
```