Assignment: Database Connectivity using JDBC

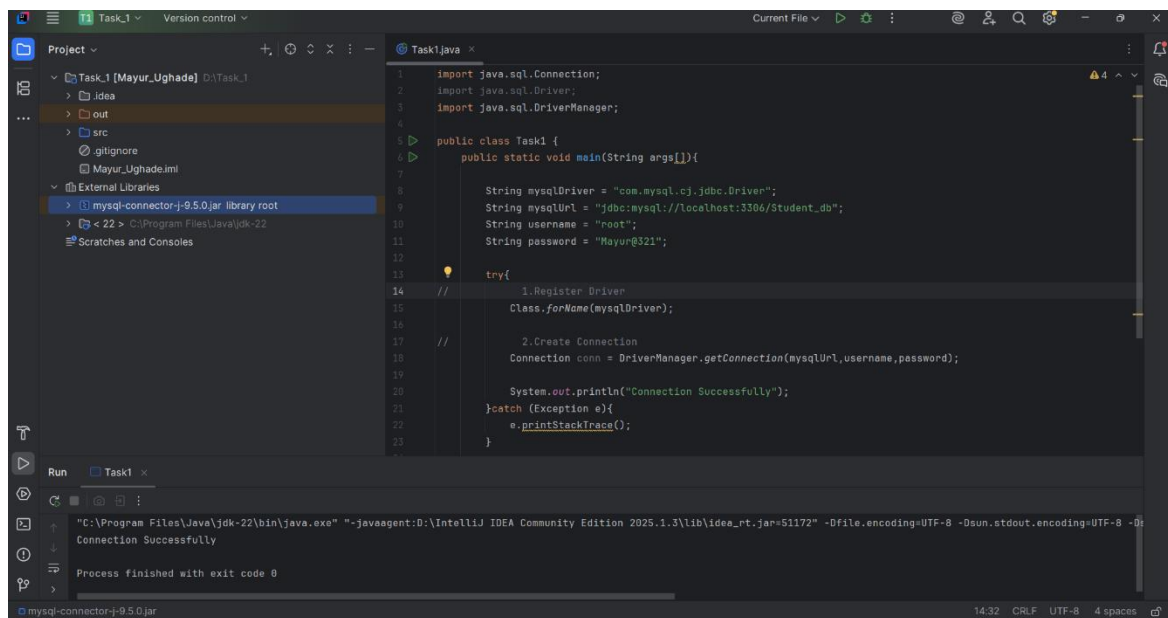**Name: Mayur Ughade**

Roll No: 21

Task 1: **Establishing Database Connection**

File Name: Task1.java

1. **Description**
   The objective of this task is to establish a connection between the Java application and the MySQL database Student_db. It utilizes the DriverManager class and the MySQL JDBC driver.

2. **Source Code with output**

3. **Execution Logic**

   - **Driver Loading:** Loaded `com.mysql.cj.jdbc.Driver` to communicate with the MySQL server.
   - **Connection:** Established connection using URL `jdbc:mysql://localhost:3306/Student_db`.
   - **Authentication:** Used username `root` and the specified password.
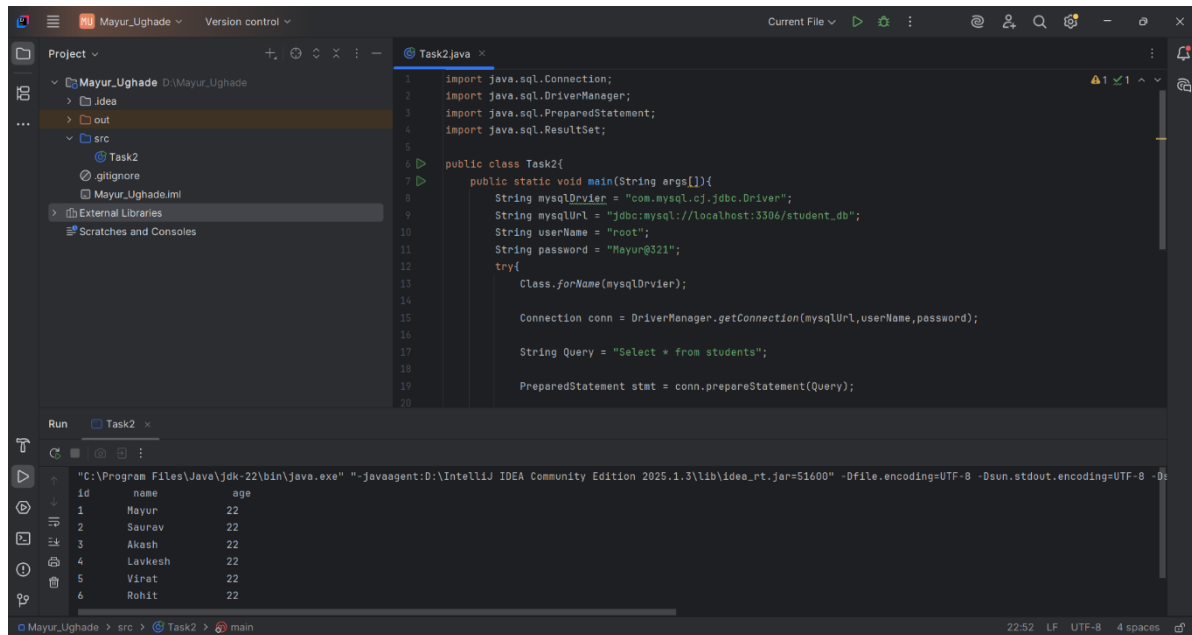
4. **Verification**

   Connection Successfully

File Name: Task2.java

Description

This task focuses on fetching all records from the students table and displaying them in a formatted tabular manner in the Java console.

1. Source Code with output

2. Execution Logic

- **Query:** Executed `Select * from students`.

- **Loop:** Iterated through the ResultSet to print ID, Name, and Age.

- **Formatting:** Used tabs (`\t`) to align columns.

3. MySQL Verification

select * from student;

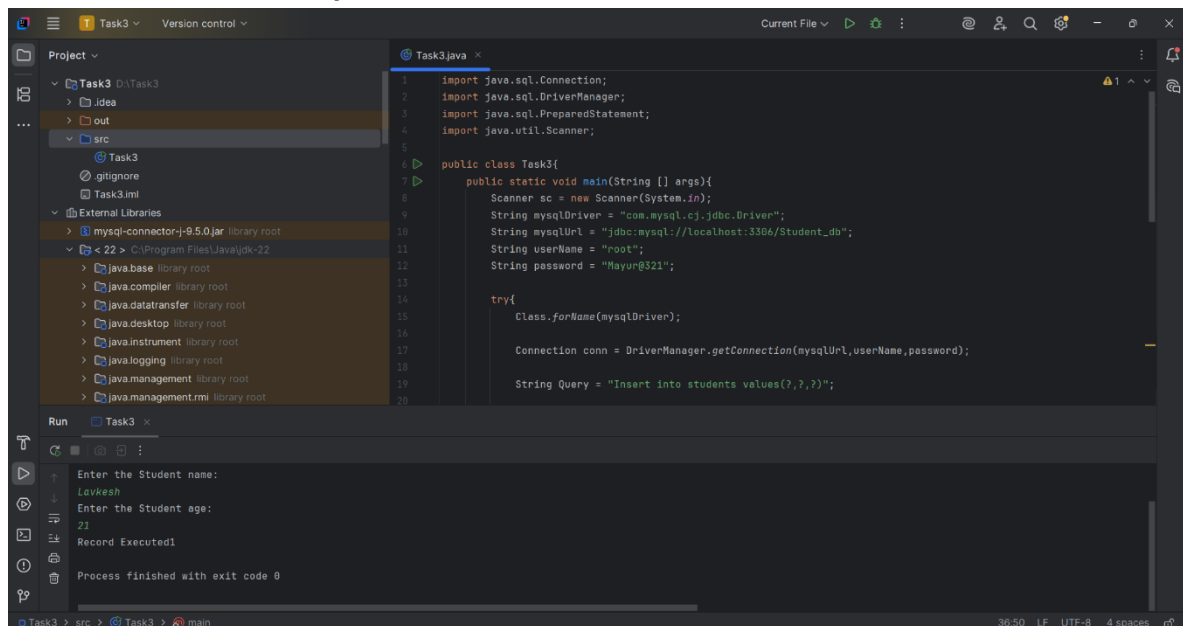| id	name	age
1	Mayur	22
2	Saurav	22
3	Akash	22
4	Lavkesh	22

Task 3: Inserting Data into Database

File Name: Task3.java

1. Description

This task demonstrates how to accept user input (Name and Age) from the console and insert a new record into the student table using a PreparedStatement.

2. Source Code with output

3. Execution Logic

- **Input:** Scanner captured Name (`Lavkesh`) and Age (`21`).

- **Query:** Used `Insert into students values(?,?,?)`.

- **Security:** PreparedStatement used to handle inputs safely.

4. MySQL Verification

Enter the Student name: Lavkesh Enter the Student age: 21 Record Executed:1
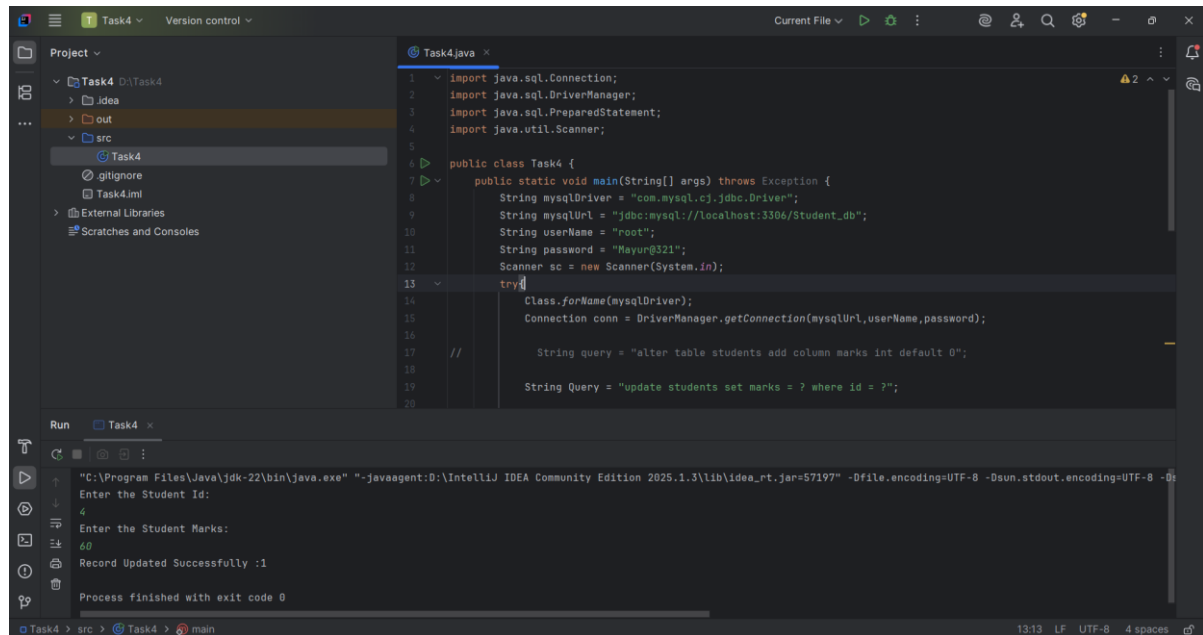
Task 4: Updating Existing Records

File Name: Task4.java

Description

This task performs an update operation. It first alters the table to add a marks column, then takes a Student ID and new Marks from the user to update the record.

1. Source Code with output

2. Execution Logic

Alter: `alter table students add column marks int default 0.`

Update: `update students set marks = ? where id = ?.`

Input: Updated Student ID 4 with Marks 60.

3. MySQL Verification

BEFORE UPDATION

select * from student;

| id | name | age | marks |

+----+--------+------+ ------- +

| 1 | Saurav | 21 | 0 |

| 2 | Mayur | 20 | 0 |

| 3 | Nikhil | 21 | 0 |

| 4 | Anand | 5 | 0 |

AFTER UPDATION

select * from student;

| id | name | age | marks |

+----+--------+------+ ------- +

| 1 | Saurav | 21 | G0 |

| 2 | Mayur | 20 | 0 |

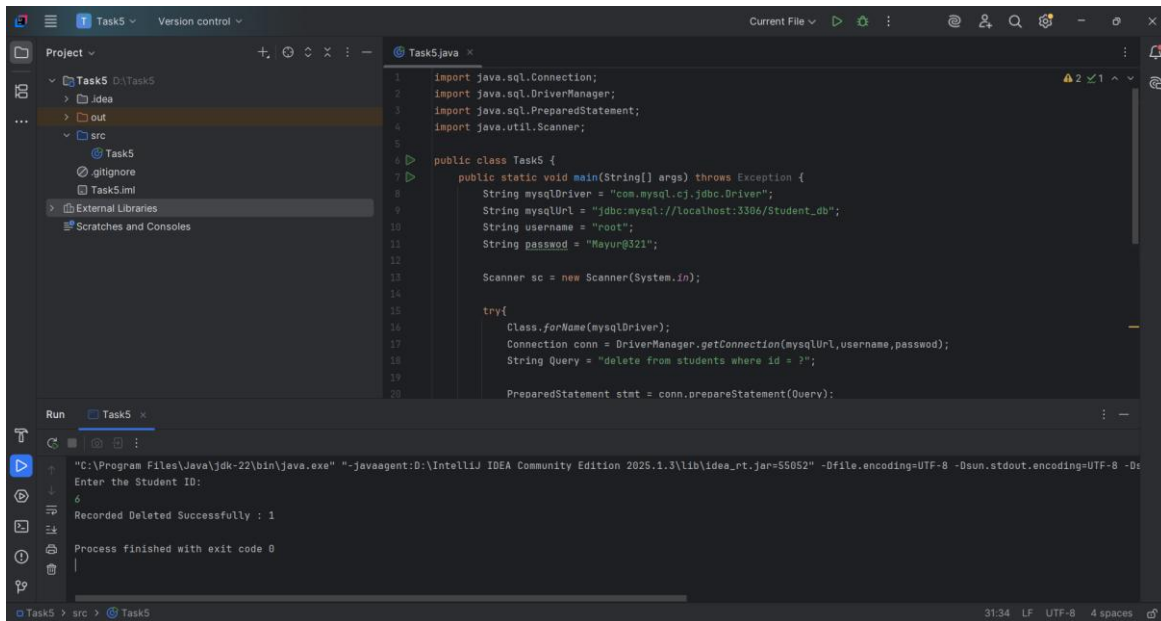| 3 | Nikhil | 21 | 0 |

| 4 | Anand | 5 | 0 |

| 5 | Akash | 21 | 0 |

Task 5: Deleting Records from Database

File Name: Task5.java

1. Description

This task handles the removal of data. It prompts the user for a Student ID and deletes that specific record from the student table.

2. Source Code with output

3. Execution Logic

- **Input:** The user provided Student ID 6.

- **Query:** delete from students where id = ?.

- **Result:** The record for Student ID 6 was permanently removed.

4. MySQL Verification

select * from student;

| id | name | age | marks |

+----+--------+------+------- +

| 1 | Saurav | 21 | G0 |

| 2 | Mayur | 20 | 0 |

| 3 | Nikhil | 21 | 0 |

| 4 | Anand | 5 | 0 |