**STUDENT**

**RECORD**

**MANAGEMENT**

**SYSTEM**

Group Member:

Muhammad Saad 2112118

Fawad Masood Khan 2112111

Wajahat Hussain 2112134

Course: Data Structures and Algorithm

Instructor: Sir Talha Javaid

Table of Contents

[INTRODUCTION: 2](#_Toc123562093)

[SCOPE: 2](#_Toc123562094)

[FUNCTIONALITIES: 2](#_Toc123562095)

[IMPLEMENTATION OF THE LOGIC: 3](#_Toc123562096)

[ERROR HANDLING 4](#_Toc123562097)

[HOW IT WORKS 4](#_Toc123562098)

[SCREENSHOTS: 5](#_Toc123562099)

[SOLVING PROBLEM STATEMENT: 14](#_Toc123562100)

[CONCLUSION 14](#_Toc123562101)

# 

# INTRODUCTION:

This program is an implementation of a Student Record Management System. It is designed to store and manage records of students enrolled in an institute. This program is a student record management system for the SZABIST University. It is a project to demonstrate the application of Linked List data structure. The system is designed to help manage student records, such as student name, course, roll number, and marks. The SRMS is a console-based application, is implemented using MySQL and Java language, that allows the user to create, delete, search, and view student records. The system will have a login, menu and a user interface with CRUD capabilities. The login will be used to validate the user who is using the system and to access the menu and the user interface. This program allows the user to create, search, delete, and view student records. The users must authenticate themselves by entering the correct username and password to access the system.

# SCOPE:

The scope of this program is limited to the student record management system of the SZABIST University. The user interface is designed to be simple and intuitive, and the data stored is kept secure and encrypted. The system is built using the Java programming language and MySQL database, and the linked list data structure is used to store and manipulate the data. This program will provide the user with an interface to manage student records. The program uses a linked list data structure to store the student records. It allows the user to enter student details such as name, course, marks etc. and store them in the linked list. The user will be able to create, delete, search and view student records. The scope of the Management System program is to provide a user-friendly interface for users to manage student records. The program should provide an easy and secure way for users to create, delete, search and view student records. The program should also provide an authentication system to protect the user’s data.

# FUNCTIONALITIES:

The program is able to carry out the following tasks:

The source code of the student management system is written in Java and is composed of three different packages. The “com.company” package contains the main program that acts as the UI for the user to interact with the program and contains the code related to database connectivity. The “Node” package keeps the model and structure of the student records stored in the database. The “Student” package contains the methods which are used to perform the student record management operations.

Create Record: The user will be able to create a new student record. The user will be required to enter the student’s name, roll number, course, and total marks.

Delete Record: The user will be able to delete a student record. The user will be required to enter the roll number of the student whose record is to be deleted.

Search Record: The user will be able to search for a student record. The user will be required to enter the roll number of the student whose record is to be searched.

View Records: The user will be able to view all the student records.

Authentication system: The program provides an authentication system to protect the user’s data. The user must enter the correct username and password to access the system.

# IMPLEMENTATION OF THE LOGIC:

The program was implemented in java and used SQL database as the backend of the system. The linked list data structure is used to store and manipulate the data. The user interface is designed for easy navigation. The program uses a try-catch block to handle any exceptions thrown by the user. The program also uses the dbConnection () method to connect to the MySQL database. The system is composed of three classes: Main, Node, and Student. The Main class is the main entry point of the program. It contains the main() method, which is responsible for prompting the user to enter their username and password. If the username and password are correct, the user is presented with the menu of options. The Node class is responsible for creating a node object for each student record. The Student class is responsible for managing the student records. It contains methods for creating, deleting, searching, and viewing student records. The program is written in Java and uses the Node class to store the student records. The Node class has four fields: roll, name, department, and marks. The program uses a linked list to store the student records. The linked list is implemented using the Node class. The program uses a linked list data structure to store the student records. The linked list is a data structure in which each node contains a student record. The user can create, delete, search, and view student records using the linked list.

The program also implements an authentication system to protect the user’s data. The user must enter the correct username and password to access the system. The program uses an if-else statement to check whether the user has entered the correct username and password. If the user has entered the correct username and password, the program will display a message stating that the authentication was successful. Otherwise, the program will display an error message and the user will be asked to enter the username and password again. The program has four main functions: dbconnection (), find (), insertRecord (), searchRecord (), and deleteRecord ().

The dbConnection () function deals with the SQL database connectivity, retrieving/inserting or deleting data from or to the database.

The find () function is used to check if the record exists before adding or deleting it.

The addRecord() function is used to add a new student record to the linked list. The function takes the student's roll number, name, department, and marks as parameters. The function first checks if the record already exists before adding it to the list. If the record exists, the function prints an error message. Otherwise, the record is added to the list.

The deleteRecord() function is used to delete a student record from the linked list. The function takes the student's roll number as a parameter. The function first checks if the record exists before deleting it from the list. If the record does not exist, the function prints an error message. Otherwise, the record is deleted from the list.

The searchRecord() function is used to search for a student record in the linked list. The function takes the student's roll number as a parameter. The function first checks if the record exists before displaying it. If the record does not exist, the function prints an error message. Otherwise, the record is displayed.

The showRecord() function is used to view all the student records in the linked list. The function displays all the student records in the list.

# ERROR HANDLING

The program handles errors in the following ways:

Already Exist: The program provides the user with error handling to ensure the accuracy of the data. The user is given an error message if a student record already exists or if the record does not exist.

Authentication error: If the user has entered the wrong username and password, the program will display an error message and will ask the user to enter the correct username and password.

Input validation: The program has error handling for all the functions. The check() function checks if a student record already exists in the linked list. If a record already exists, the program displays an error message. The insertRecord() function checks if a student record already exists in the linked list. If a record already exists, the program displays an error message and does not insert the record. The searchRecord() function checks if a student record exists in the linked list. If a record does not exist, the program displays an error message. The deleteRecord() function checks if a student record exists in the linked list. If a record does not exist, the program displays an error message and does not delete the record.

Exception Mismatch: Sometimes the input from user can throw an illegal character which will make the program crash so to avoid this input mismatch exception is handled.

Overall, the system is designed keeping in mind the security of the whole system as one of the main priorities so that malicious users can’t attack the system and cause damage.

# HOW IT WORKS

The program begins with the main method which is the entry point of the program. The main method will prompt the user to enter their username and password. If the username and password are correct, the authentication will be successful and the program will proceed to the next step.

The program will then display a menu to the user with the following options:

Create a new record

Delete a student record

Search a student record

View all student’s record

Exit

The user will then enter their choice. Depending on the user’s choice, the program will execute the corresponding method.

If the user selects 1, the program will prompt the user to enter the student’s name, roll number, course, and total marks. The program will then create a new node with the student’s information and add it to the linked list.

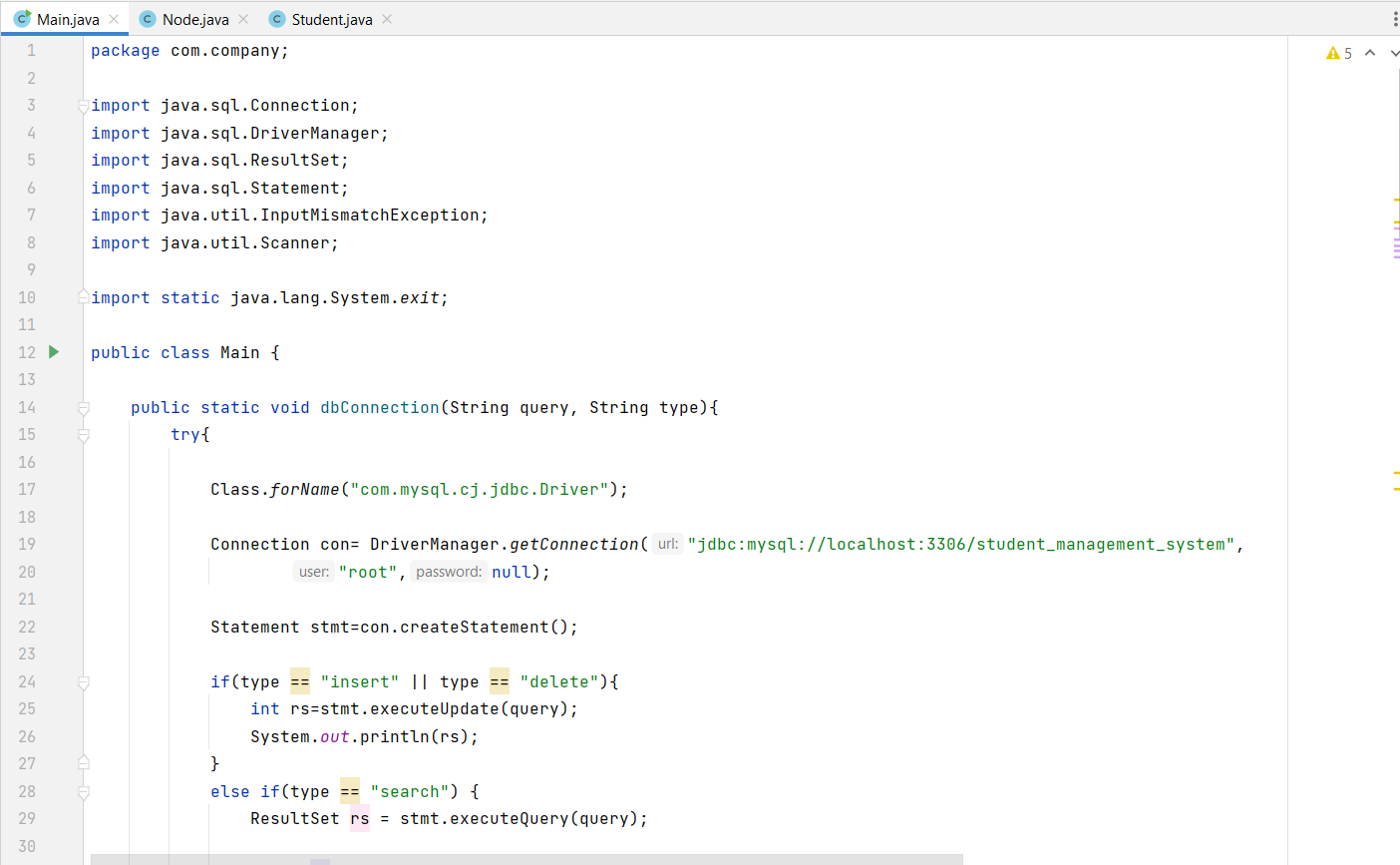
If the user selects 2, the program will prompt the user to enter the roll number of the student whose record is to be deleted. The program will then delete the corresponding node from the linked list.

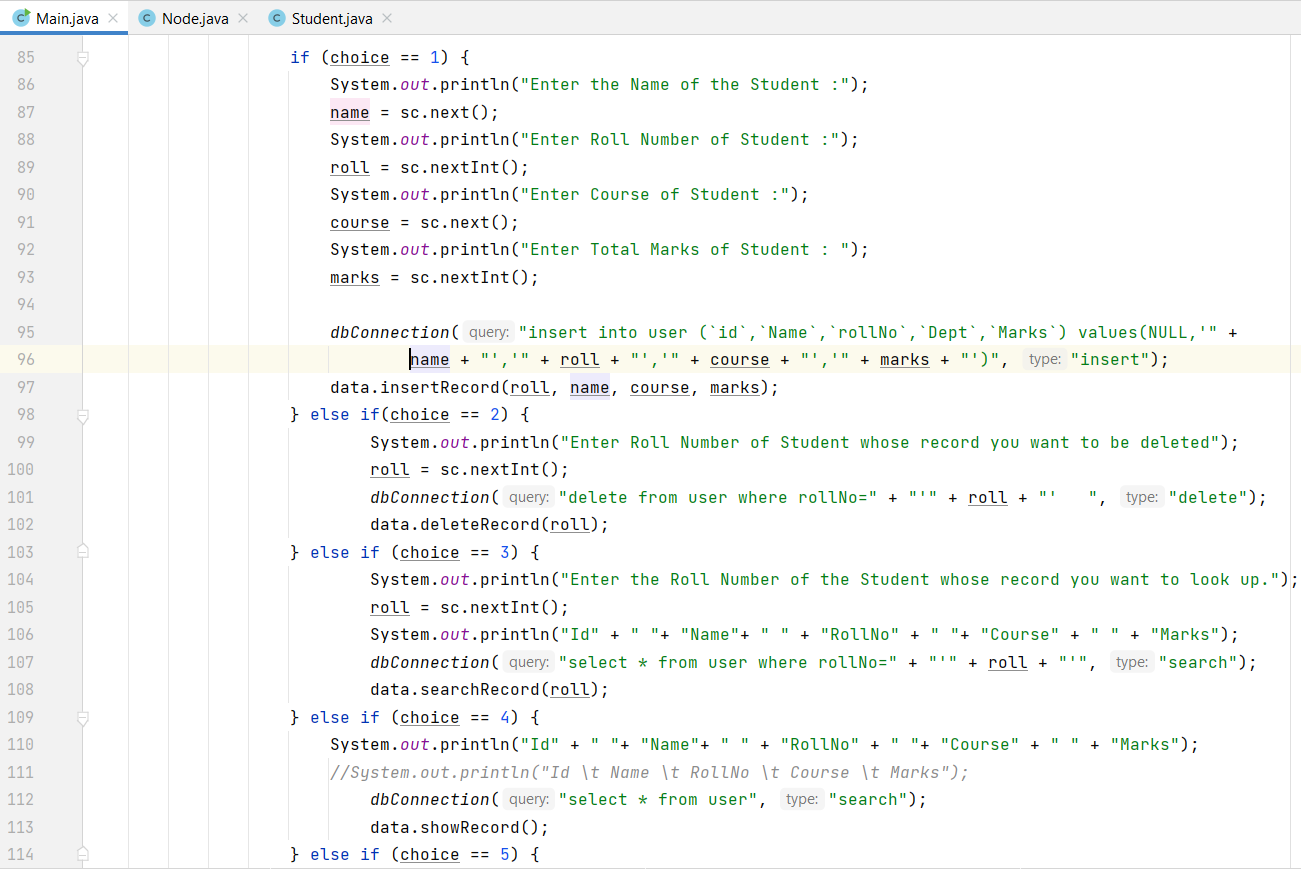
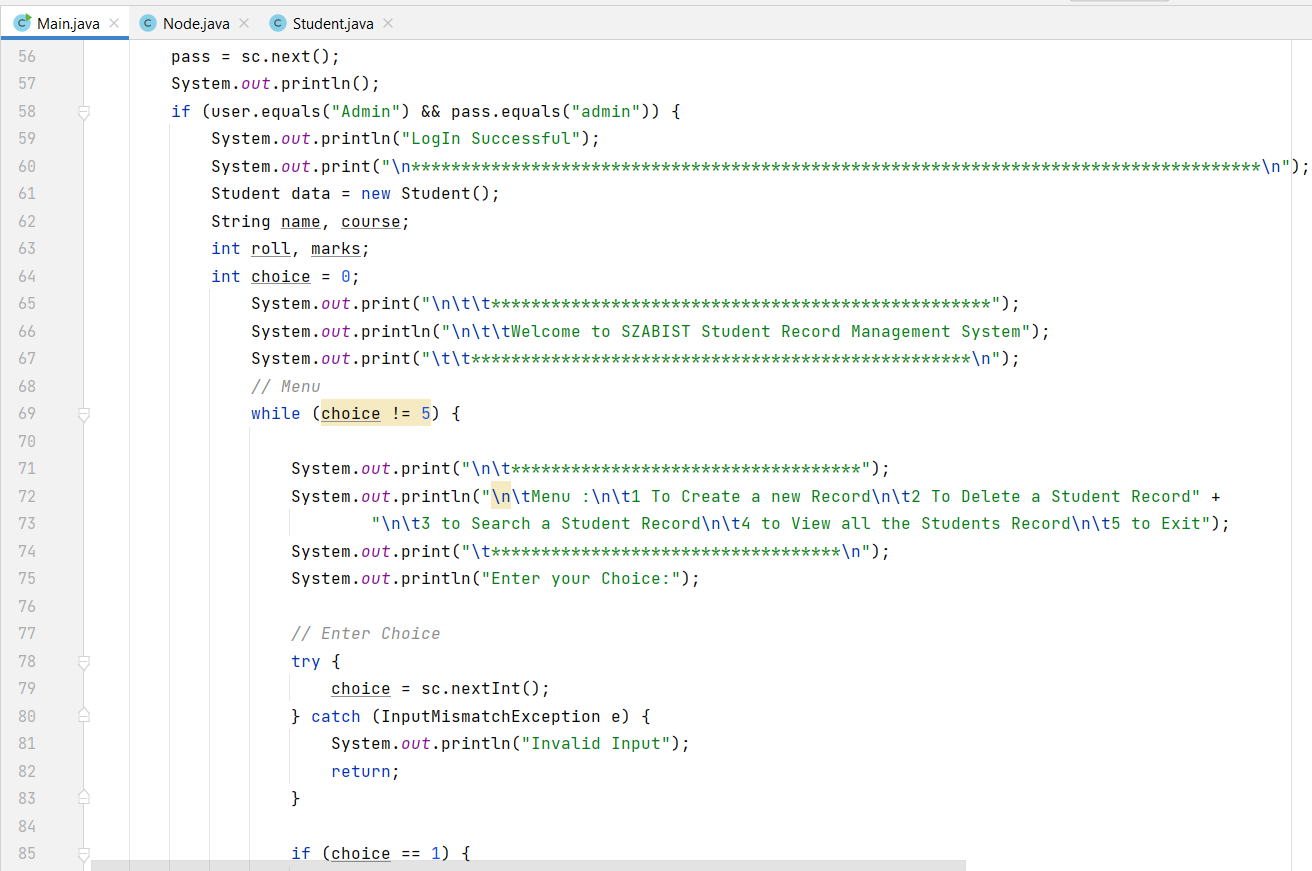
If the user selects 3, the program will prompt the user to enter the roll number of the student whose record is to be searched. The program will then search for the corresponding node in the linked list and display the student’s information.

If the user selects 4, the program will display all the student records stored in the linked list.

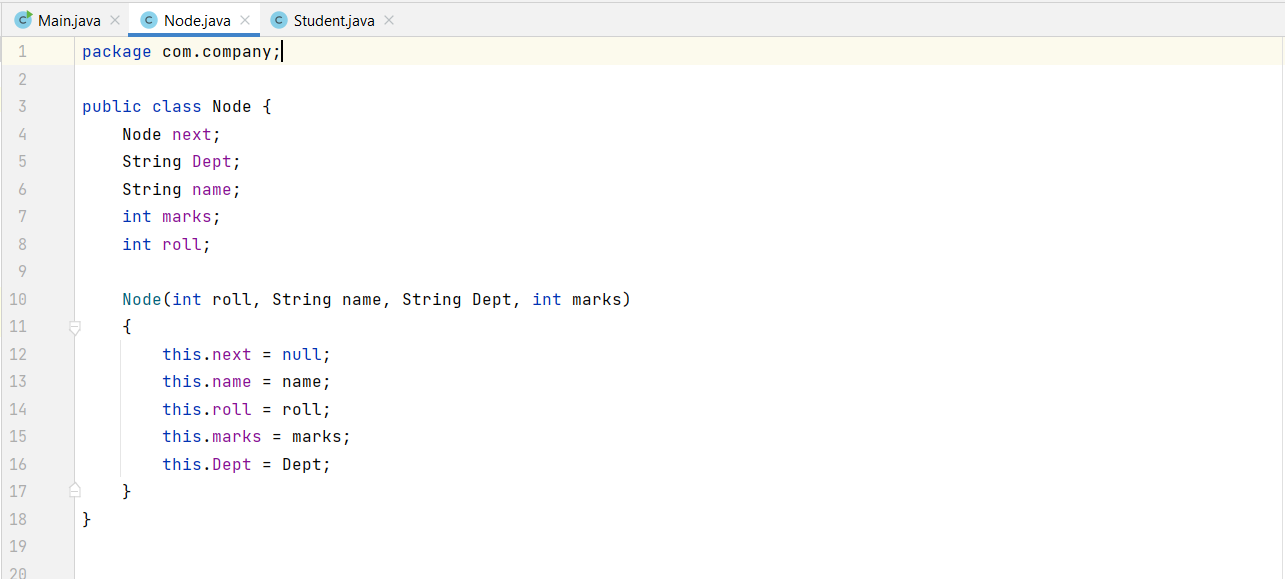
Finally, if the user selects 5, the program will exit.

# SCREENSHOTS:

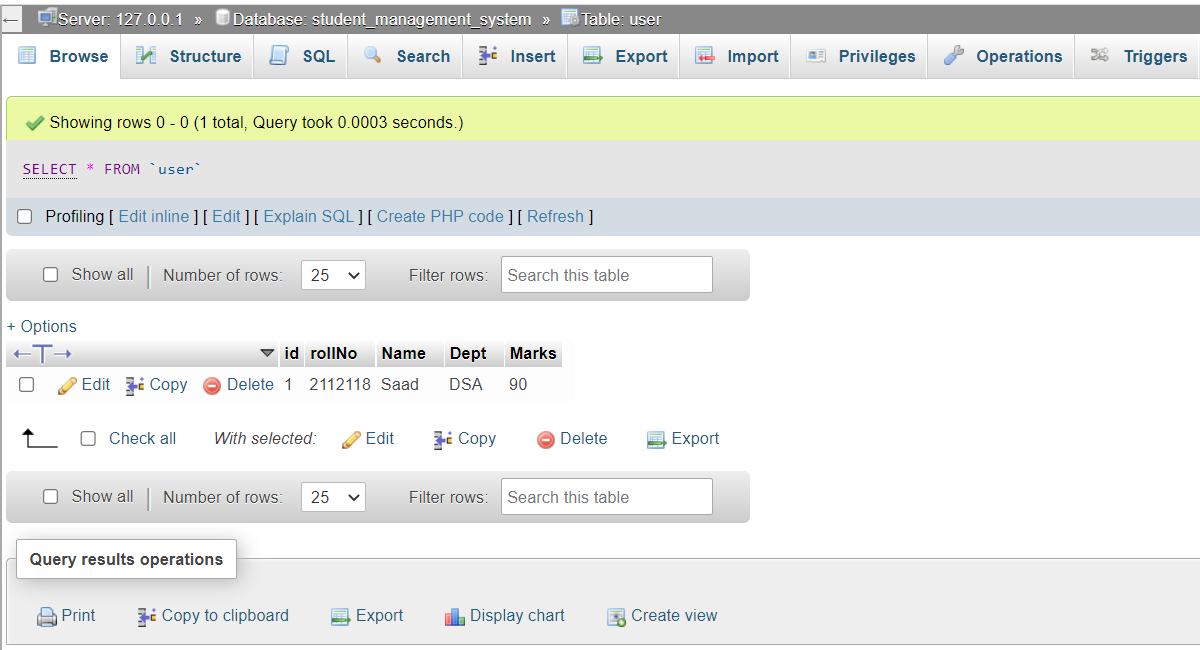
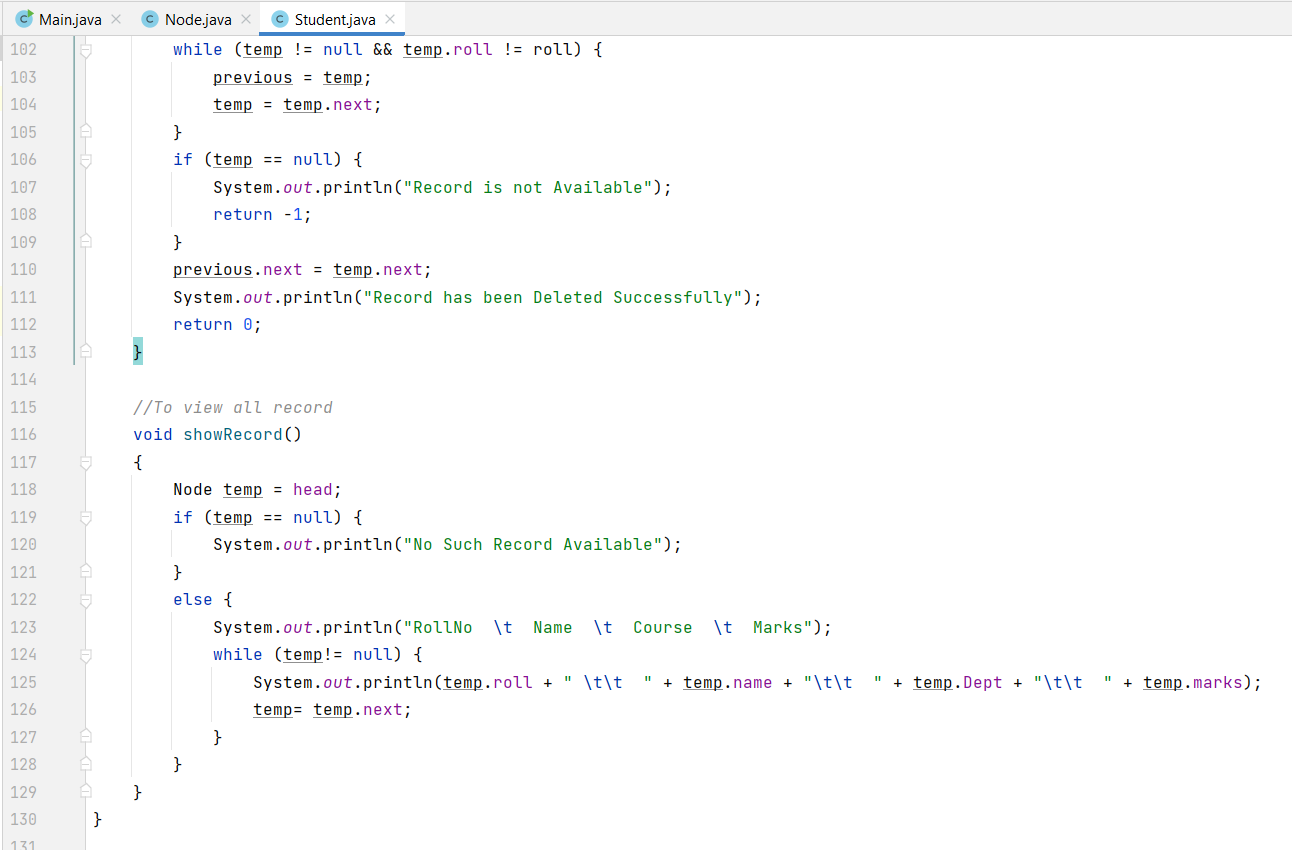
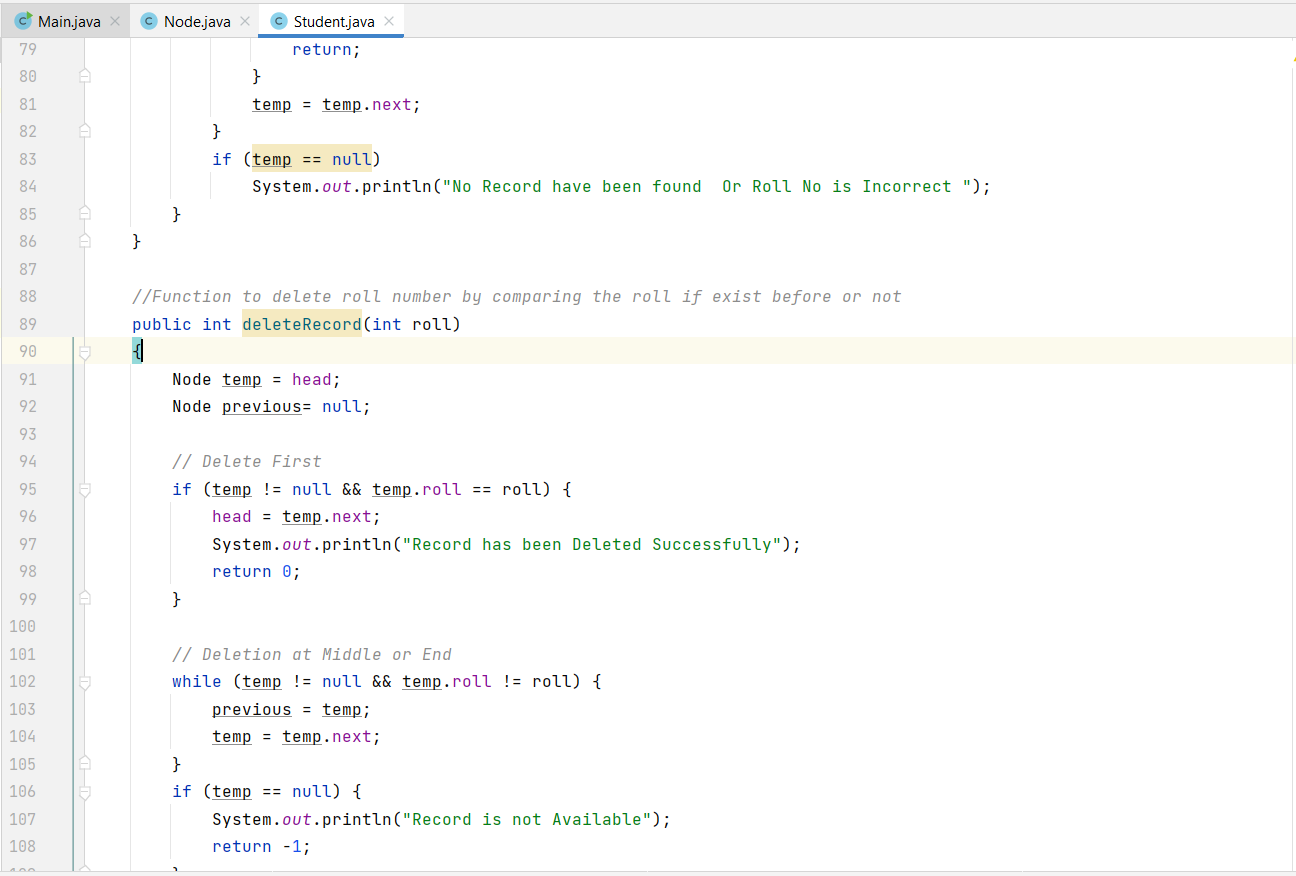
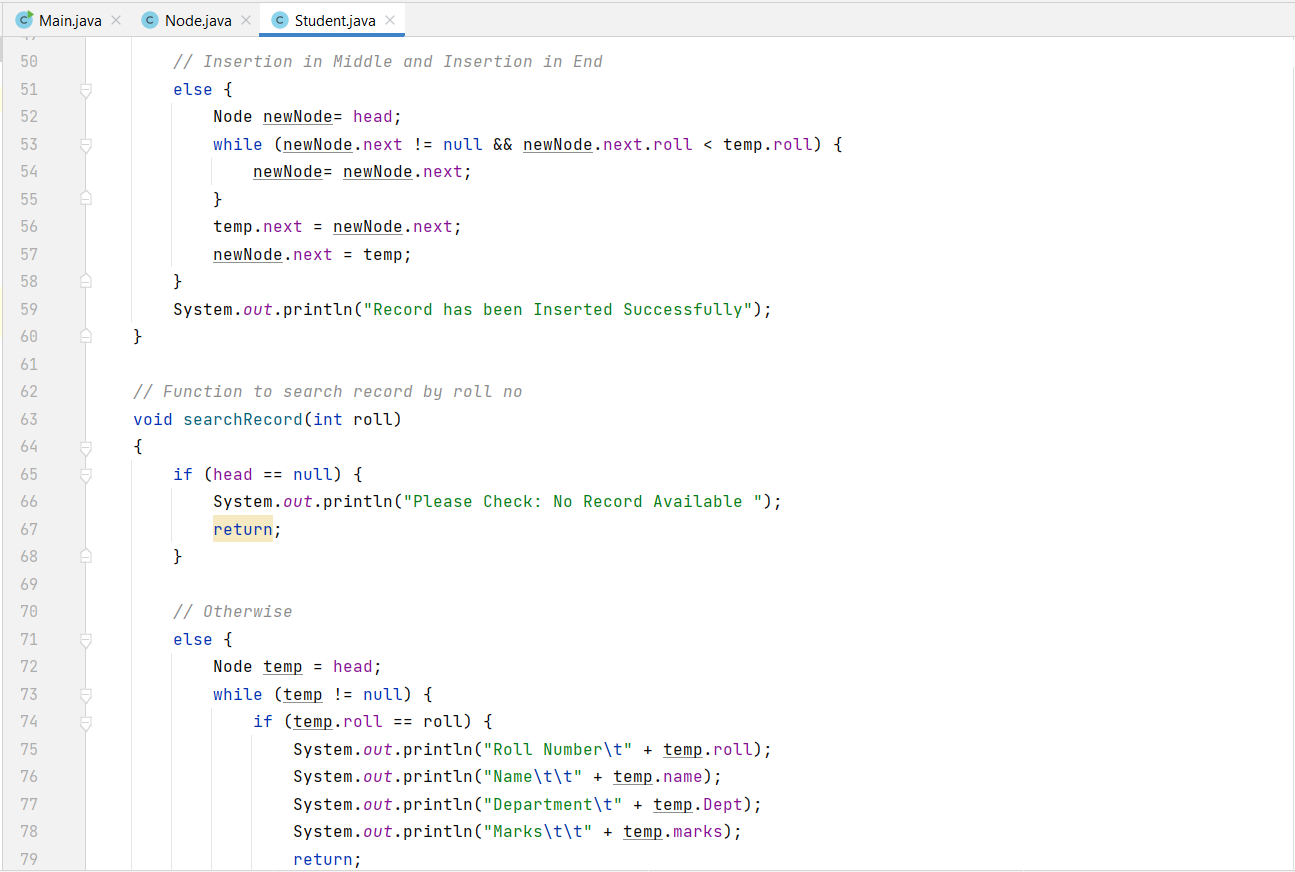
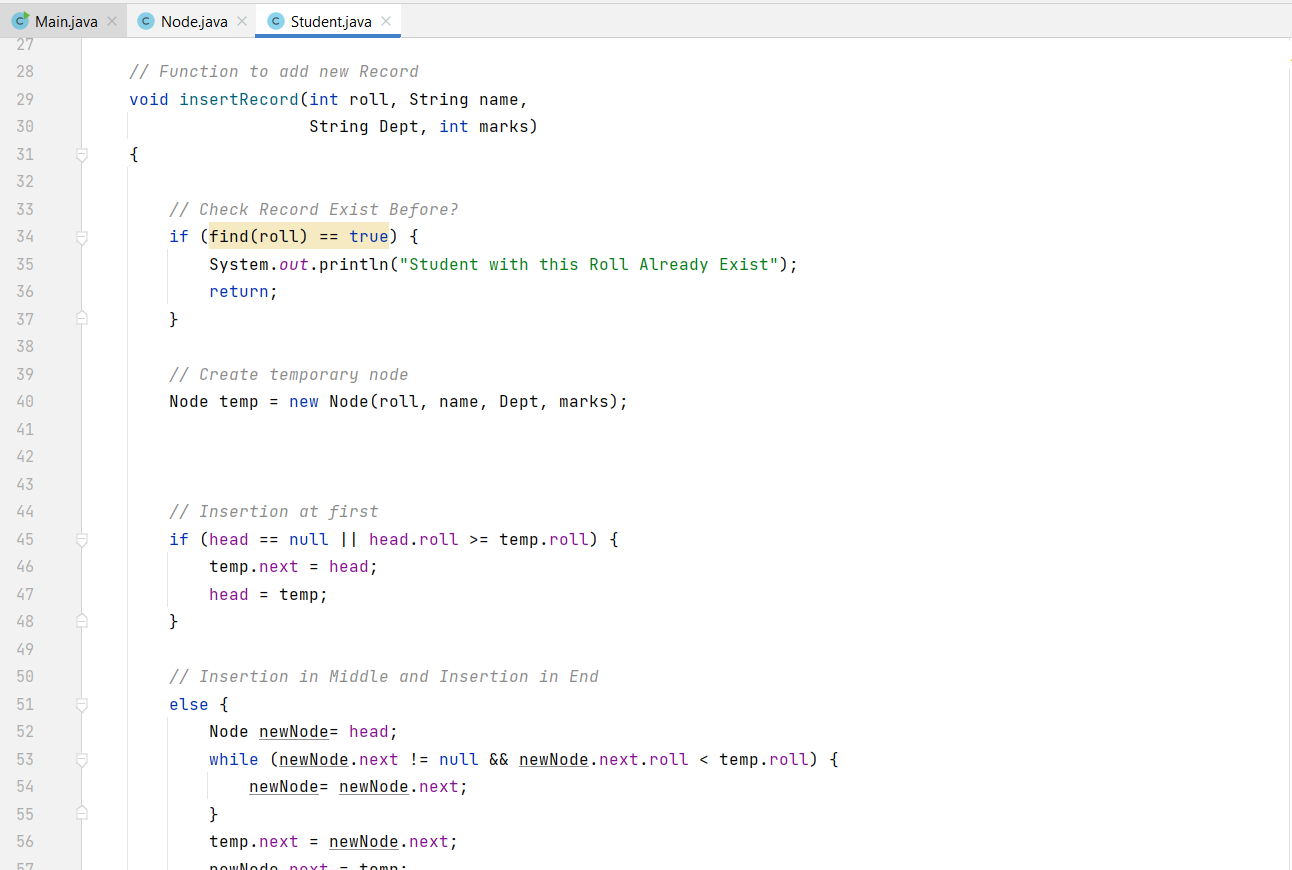


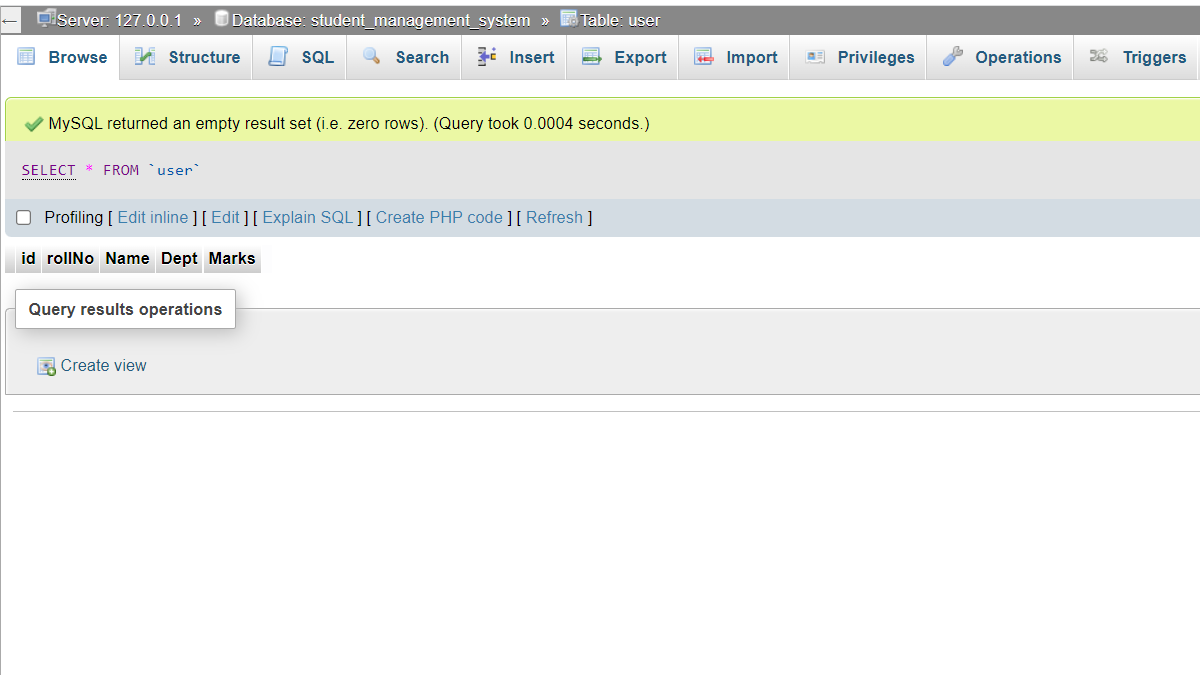
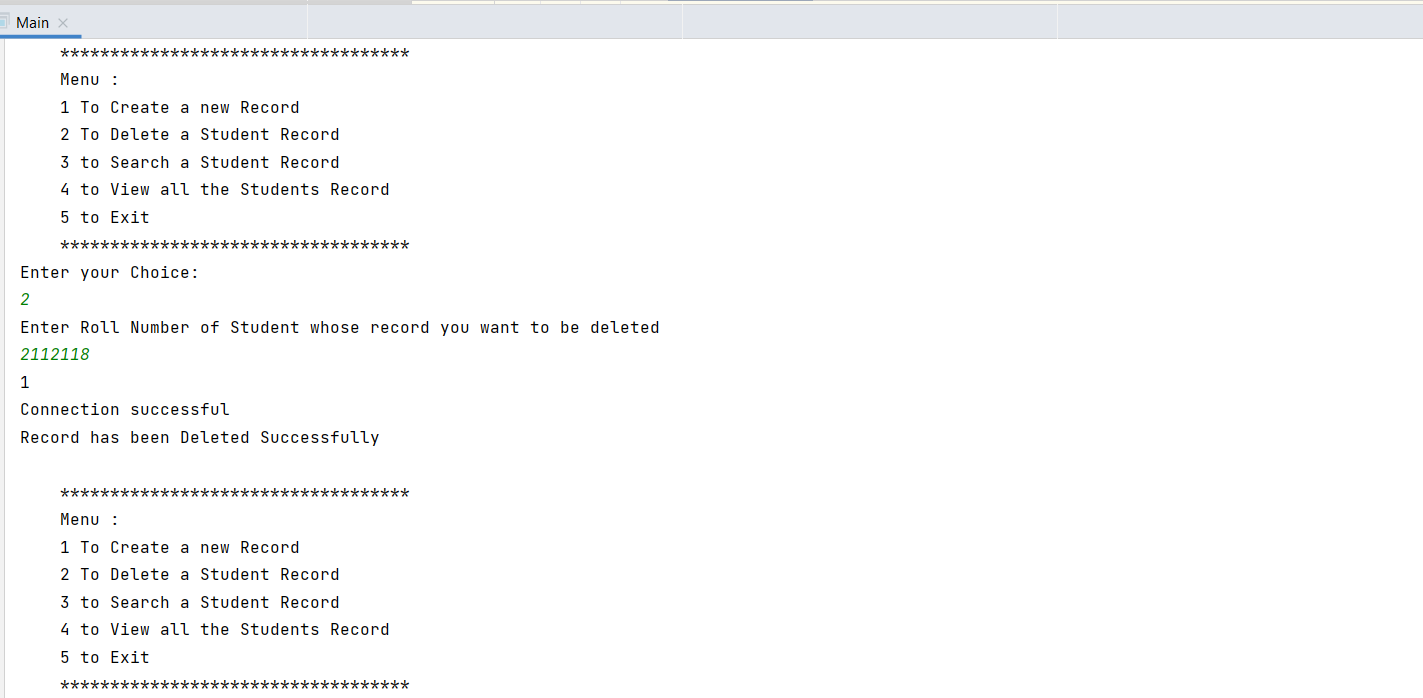
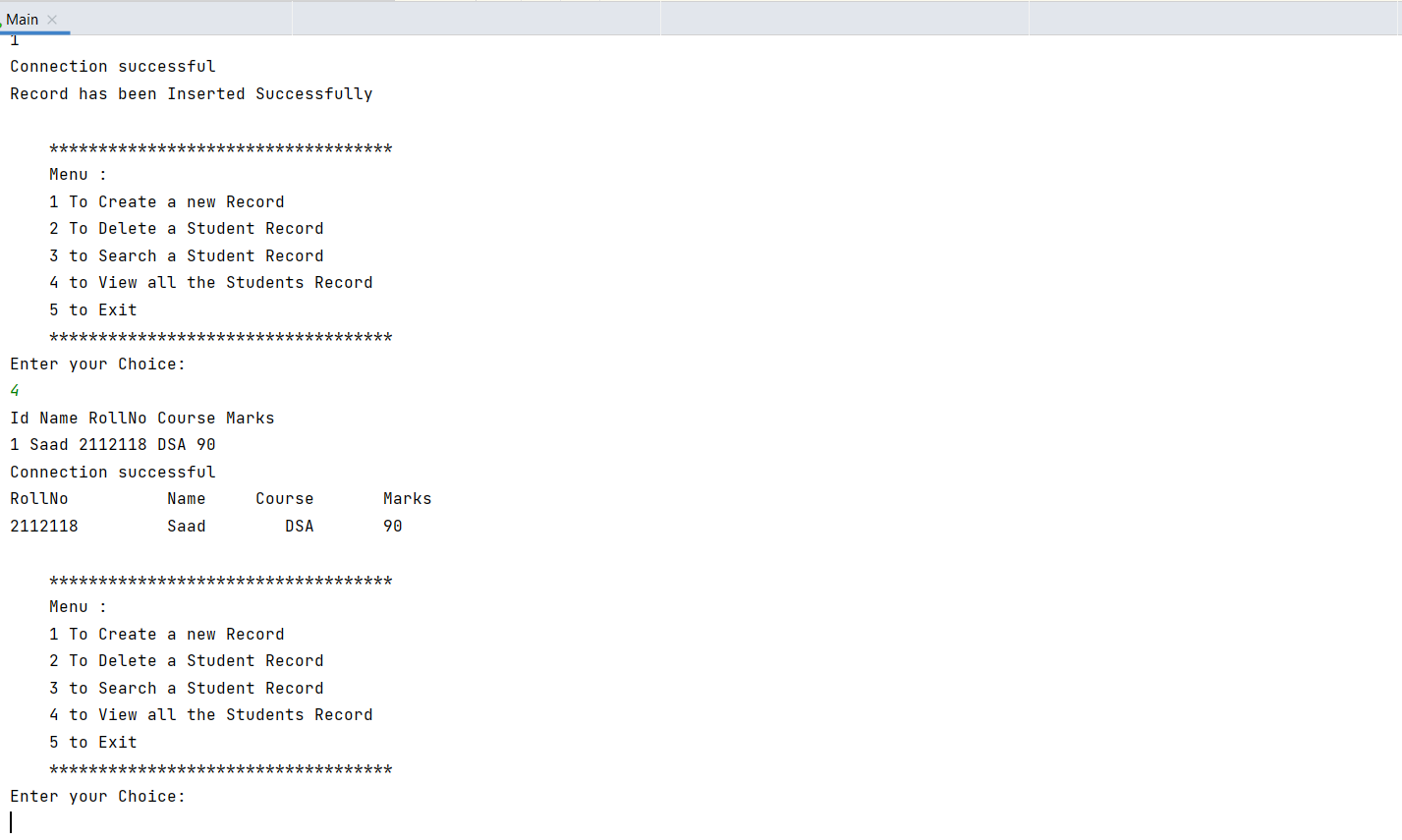
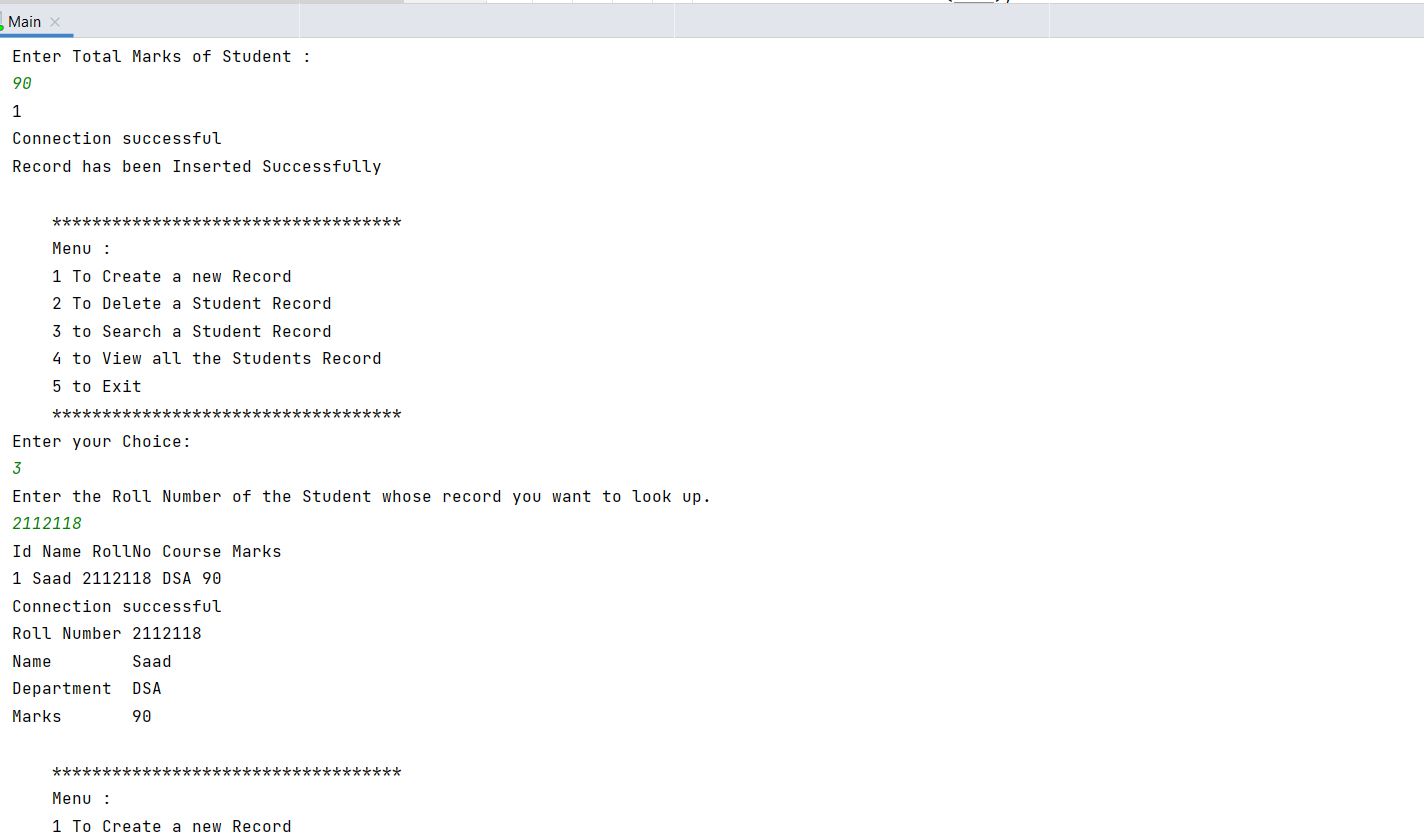














# SOLVING PROBLEM STATEMENT:

* RECORD SEARCHING: When there is no automated system in place, it is always challenging to search for records, especially if there are many of them.
* COST CONSUMING: As there is no computerized way to add each record, paper will be required, which will raise the cost of managing the library.
* SPACE CONSUMING: If no computerized system is put in place, the space for physical storage of files and data will likewise increase as the quantity of records increases.
* FILE BROKEN: When a computerized system is absent, files are frequently destroyed as a result of accidents, such as member’s accidently spilling water on them. Additionally, the data might be harmed by fires or floods, two examples of natural disasters.

# CONCLUSION

This program is an implementation of a Student Record Management System. It is designed to be user friendly and secure, and uses the Java programming language and MySQL database to store and manipulate the data to provide the user with an interface to manage the records of the students in an institute. The Student Record Management System is a program that demonstrates the application of the Linked List data structure. The program provides the user with an interface to manage student records. The program is able to create, delete, search, and view student records. The program also provides the user with error handling to ensure the accuracy of the data. The program also provides an authentication system to protect the user’s data. The program is implemented using Java programming language, MySQL and a linked list data structure. The program also handles errors by validating the user’s input and data. The program is a useful tool for managing student records.