**Student Information Management System**

Members:

Gofredo, John Erick

Guial Gelli

Compano, Mark John Loyd

Macalinao, Charlos Dhanniel

Badron, Colah

**Pseudo Code**

**Variables.java**

package Declaration;

import java.sql.\*;

**This code block is for declaration of variables that will inherit into another class**

public class Variables {

protected String username,password;

protected String query ;

protected ResultSet result = null ;

protected Connection conn = null;

protected PreparedStatement pstmnt = null ;

protected String dbname = "jdbc:derby://localhost:1527/StudentInformation";

protected int choice , suBchoice , id;

protected String firstName , middleName , lastName , gender , course , section , age , colName , colVal , syear ;

}

**Connections.java**

package Database;

import java.sql.\*;

import Declaration.Variables;

**// This code block is for testing of connection to derby database**

public class Connections extends Variables {

public void connecting(){

try{

dbname = "jdbc:derby://localhost:1527/StudentInformation";

Connection conn = DriverManager.getConnection(dbname);

System.out.println("Connected");

System.out.println();

}

catch(Exception a){

a.printStackTrace();

System.out.println("Connection Failed");

}

}

}

**Add.java**

package HttpMethod;

import Declaration.Variables;

import java.util.Scanner;

import java.sql.\*;

**This code block inserts student data into a Derby database. The variables used are inherited from the Variables class. Additionally, java.util.Scanner is imported to read user input and store it in the corresponding fields.**

public class Add extends Variables {

Scanner sc = new Scanner(System.in);

public void addStudent() {

try {

System.out.print("Enter First Name: ");

firstName = sc.nextLine();

System.out.print("Enter Middle Name: ");

middleName = sc.nextLine();

System.out.print("Enter Last Name: ");

lastName = sc.nextLine();

System.out.print("Enter Age: ");

age = sc.nextLine();

System.out.print("Enter Gender: ");

gender = sc.nextLine();

System.out.print("Enter Course: ");

course = sc.nextLine();

System.out.print("Enter Year: ");

syear = sc.nextLine();

System.out.print("Enter Section: ");

section = sc.nextLine();

conn = DriverManager.getConnection(dbname);

query = "INSERT INTO STUDENTS (FIRST\_NAME, MIDDLE\_NAME, LAST\_NAME, Age, GENDER , COURSE, STUDENT\_YEAR,SECTION) VALUES (?, ?, ?, ?, ?, ?,?,?)";

pstmnt = conn.prepareStatement(query);

pstmnt.setString(1, firstName);

pstmnt.setString(2, middleName);

pstmnt.setString(3, lastName);

pstmnt.setString(4, gender);

pstmnt.setString(5, age);

pstmnt.setString(6, course);

pstmnt.setString(7, syear);

pstmnt.setString(8, section);

System.out.println();

int rowsAffected = pstmnt.executeUpdate();

if (rowsAffected > 0) {

System.out.println("Inserted Data.");

} else {

System.out.println("Insert failed.");

}

} catch(SQLException e) {

// Handle any SQL errors

e.printStackTrace();

} catch(Exception e) {

// Handle other errors

e.printStackTrace();

} finally {

// Close the resources in finally block to ensure they are always closed

try {

if (result != null) {

result.close();

}

if (pstmnt != null) {

pstmnt.close();

}

if (conn != null) {

conn.close();

}

} catch(SQLException e) {

e.printStackTrace();

}

}

}

}

**Display.java**

package HttpMethod;

import java.sql.\*;

import Declaration.Variables;

**This code block is for reading data from the database and displaying it. The variables used are inherited from the Variables class. It executes a query to retrieve all records from the STUDENTS table and prints each student's details using a while loop.**

public class Display extends Variables {

public void students(){

try {

conn = DriverManager.getConnection(dbname);

query = "SELECT \* FROM STUDENTS";

pstmnt = conn.prepareStatement(query);

result = pstmnt.executeQuery();

System.out.println("Students List");

System.out.println();

while (result.next()) {

id = result.getInt("Student\_ID");

firstName = result.getString("First\_Name");

middleName = result.getString("Middle\_Name");

lastName = result.getString("Last\_Name");

age = result.getString("AGE");

gender = result.getString("Gender");

course = result.getString("Course");

syear = result.getString("Student\_Year");

section = result.getString("Section");

System.out.println("ID: " + id +

" First Name: " + firstName +

" Middle Name: " + middleName +

" Last Name: " + lastName +

" Age: " + age +

" Gender: " + gender +

" Course: " + course +

" Year: " + syear +

" Section: " + section +

" "

);

}

// Close the resources

result.close();

pstmnt.close();

conn.close();

} catch(SQLException e) {

// Handle any SQL errors

e.printStackTrace();

} catch(Exception e) {

// Handle other errors

e.printStackTrace();

}

}

}

**Updata.java**

package HttpMethod;

import Declaration.Variables;

import java.sql.\*;

import java.util.Scanner;

**This code block is for updating student information in the database. It prompts the user to enter the student ID to update. Then, it retrieves the student details based on the ID, allows the user to choose which column to update, and executes the update query. The variables used are inherited from the Variables class. It interacts with the user using Scanner for input and updates the database accordingly.**

public class Update extends Variables {

Scanner sc = new Scanner(System.in);

public void executeUpdate(int id){

try{

conn = DriverManager.getConnection(dbname);

query = "SELECT \* FROM STUDENTS WHERE Student\_ID = ?";

pstmnt = conn.prepareStatement(query);

pstmnt.setInt(1, id);

result = pstmnt.executeQuery();

while (result.next()) {

System.out.println("STUDENT INFORMATION");

id = result.getInt("Student\_ID");

firstName = result.getString("First\_Name");

middleName = result.getString("Middle\_Name");

lastName = result.getString("Last\_Name");

age = result.getString("AGE");

gender = result.getString("Gender");

course = result.getString("Course");

syear = result.getString("Student\_Year");

section = result.getString("Section");

System.out.println("ID: " + id +

" Firstname: " + firstName +

" Middlename: " + middleName +

" Lastname: " + lastName +

" Age: " + age +

" Gender: " + gender +

" Course: " + course +

" Year: " + syear +

" Section: " + section +

" "

);

}

System.out.println();

System.out.println("Choose column that you want to update:");

System.out.println("[1] Firstname [2] Middlename [3] Lastname [4] Age [5] Gender [6] Course [7] year [8] Section");

System.out.print("Enter Your Choice: ");

suBchoice = sc.nextInt();

switch(suBchoice){

case 1:

colName = "First\_Name";

break;

case 2:

colName = "Middle\_Name";

break;

case 3:

colName = "Last\_Name";

break;

case 4:

colName = "Age";

break;

case 5:

colName = "Gender";

break;

case 6:

colName = "Course";

break;

case 7:

colName = "Student\_Year";

break;

case 8:

colName = "Section";

break;

default:

System.out.println("Invalid Input");

}

sc.nextLine(); // Consume newline character

System.out.print("Enter the value for " + colName + " : ");

colVal = sc.nextLine();

System.out.println();

try{

query = "UPDATE STUDENTS SET " + colName + " = ? WHERE STUDENT\_ID = ?";

pstmnt = conn.prepareStatement(query);

pstmnt.setObject(1, colVal);

pstmnt.setInt(2, id);

System.out.println();

int rowsAffected = pstmnt.executeUpdate();

if (rowsAffected > 0) {

System.out.println("Successfully updated column " + colName + " where Student ID = " + id + "");

System.out.println();

System.out.print("Do you want to update more column for Student ID = " + id + "? [y]yes [n]no? ");

String con = sc.nextLine();

if(con.equals("y")){

executeUpdate(id);

}else if(con.equals("n")){

}else{

System.out.print("Invalid Input");

}

} else {

System.out.println("updating failder");

}

} catch(SQLException e) {

// Handle any SQL errors

e.printStackTrace();

} catch(Exception e) {

// Handle other errors

e.printStackTrace();

}

result.close();

pstmnt.close();

conn.close();

} catch(SQLException e) {

// Handle any SQL errors

e.printStackTrace();

} catch(Exception e) {

// Handle other errors

e.printStackTrace();

}

}

public void updateStudent(){

System.out.println("Enter Student ID that you want to update");

System.out.print("Enter ID: ");

id = sc.nextInt();

System.out.println();

executeUpdate(id);

}

}

**Delete.java**

**This code block is for deleting student information from the database. It prompts the user to enter the student ID that they want to delete. Then, it confirms the deletion with the user. If confirmed, it executes the deletion query for the specified student ID. The variables used are inherited from the Variables class. It interacts with the user using Scanner for input and deletes the data from the database accordingly.**

package HttpMethod;

import Declaration.Variables;

import java.sql.\*;

import java.util.Scanner;

public class Delete extends Variables {

Scanner sc = new Scanner(System.in);

public void deleteStudent(){

System.out.println("Enter Student ID that you want to delete");

System.out.print("Enter ID: ");

id = sc.nextInt();

sc.nextLine(); // Consume newline character

System.out.print("Are you sure? you want to delete the Student ID: " + id + " [y]yes [n]No ? ");

String confirm = sc.nextLine();

if(confirm.equals("y")){

executeDelete(id);

}else if(confirm.equals("n")){

deleteStudent();

}else{

System.out.println("Invalid input");

deleteStudent();

}

}

public void executeDelete(int id){

try{

conn = DriverManager.getConnection(dbname);

query = "DELETE FROM STUDENTS WHERE STUDENT\_ID = ? ";

pstmnt = conn.prepareStatement(query);

pstmnt.setInt(1, id);

System.out.println();

int rowsAffected = pstmnt.executeUpdate();

if (rowsAffected > 0) {

System.out.println("Deleted " + rowsAffected + " row(s) deleted.");

} else {

System.out.println("No rows deleted. The student with ID " + id + " might not exist.");

}

}catch(SQLException e) {

// Handle any SQL errors

e.printStackTrace();

} catch(Exception e) {

// Handle other errors

e.printStackTrace();

} finally {

try {

if (result != null) {

result.close();

}

if (pstmnt != null) {

pstmnt.close();

}

if (conn != null) {

conn.close();

}

} catch(SQLException e) {

e.printStackTrace();

}

}

}

**}**

**Home.java**

**This code block represents the home page of the Student Management System.**

**It provides options for the user to interact with student data.**

**The user can choose to:**

**- Display student list**

**- Add new student**

**- Update existing student information**

**- Delete student**

**- Logout**

**It utilizes classes like Display, Add, Delete, Update, and Logout for various functionalities.**

**User input is taken using Scanner.**

package Pages;

import HttpMethod.Display;

import HttpMethod.Add;

import UserEntry.Logout;

import HttpMethod.Delete;

import HttpMethod.Update;

import Declaration.Variables;

import java.util.Scanner;

public class Home extends Variables {

Scanner sc = new Scanner(System.in);

Logout logout = new Logout();

Display display = new Display();

Add add = new Add();

Delete del = new Delete();

Update update = new Update();

public void chooseOption(){

System.out.println("[1]Show Students List");

System.out.println("[2]LogOut");

System.out.println();

System.out.print("Enter Your Choice: ");

choice = sc.nextInt();

System.out.println();

switch(choice){

case 1:

display.students();

SubchooseOption();

break;

case 2:

logout.Logout();

break;

default:

System.out.println("Invalid Input");

}

}

public void SubchooseOption(){

System.out.println();

System.out.println("[1]ADD");

System.out.println("[2]SHOW");

System.out.println("[3]UPDATE");

System.out.println("[4]DELETE");

System.out.println("[5]LogOut");

System.out.println();

System.out.print("Enter Your Choice: ");

suBchoice = sc.nextInt();

switch(suBchoice){

case 1:

add.addStudent();

SubchooseOption();

break;

case 2:

display.students();

SubchooseOption();

break;

case 3:

update.updateStudent();

SubchooseOption();

break;

case 4:

del.deleteStudent();

SubchooseOption();

break;

case 5:

logout.Logout();

break;

default:

System.out.println("Invalid Input");

}

}

public Home(){

System.out.println("Welcome to Student Management System");

System.out.println();

chooseOption();

}

public void menu(){

System.out.println();

System.out.println("[1]Show Students List");

System.out.println("[2]LogOut");

}

}

**Login.java**

**This code block is for user login functionality.**

**It prompts the user to enter their username and password.**

**Then, it checks the entered credentials against the database.**

**If the credentials are correct, it displays "LOGIN SUCCESS" and allows access to the Home page.**

**If the credentials are incorrect, it displays "LOGIN FAILED" and terminates the program.**

**It utilizes the Variables class for database connection and the Home class for navigating to the home page.**

**User input is taken using Scanner.**

package UserEntry;

import Declaration.Variables;

import Pages.Home;

import java.sql.\*;

import java.util.Scanner;

public class Login extends Variables {

public void UserCredential(){

Scanner sc = new Scanner(System.in);

System.out.print("Username: ");

username = sc.nextLine();

System.out.print("Password: ");

password = sc.nextLine();

System.out.println();

CorrectCredentials(username, password);

}

public void CorrectCredentials(String username, String password ){

try {

conn = DriverManager.getConnection(dbname);

query = "SELECT \* FROM ACCOUNT WHERE username = ? AND password = ?";

pstmnt = conn.prepareStatement(query);

pstmnt.setString(1, username);

pstmnt.setString(2, password);

result = pstmnt.executeQuery();

if (result.next()) {

System.out.println("LOGIN SUCCESS");

System.out.println();

Home home = new Home();

} else {

System.out.println("LOGIN FAILED");

System.exit(0);

}

// Close the resources

result.close();

pstmnt.close();

conn.close();

} catch(SQLException e) {

// Handle any SQL errors

e.printStackTrace();

} catch(Exception e) {

// Handle other errors

e.printStackTrace();

}

}

}

**Logout.java**

**This code block is for user logout functionality.**

**It simply prints "Successfully Logout" message and terminates the program.**

package UserEntry;

public class Logout{

public void Logout(){

System.out.println("Successfully Logout");

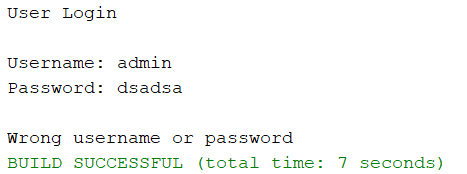
System.exit(0);

}

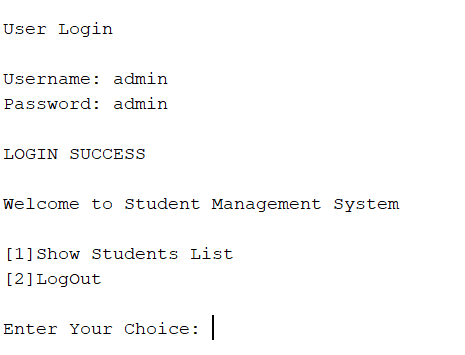
}

**OUTPUT**

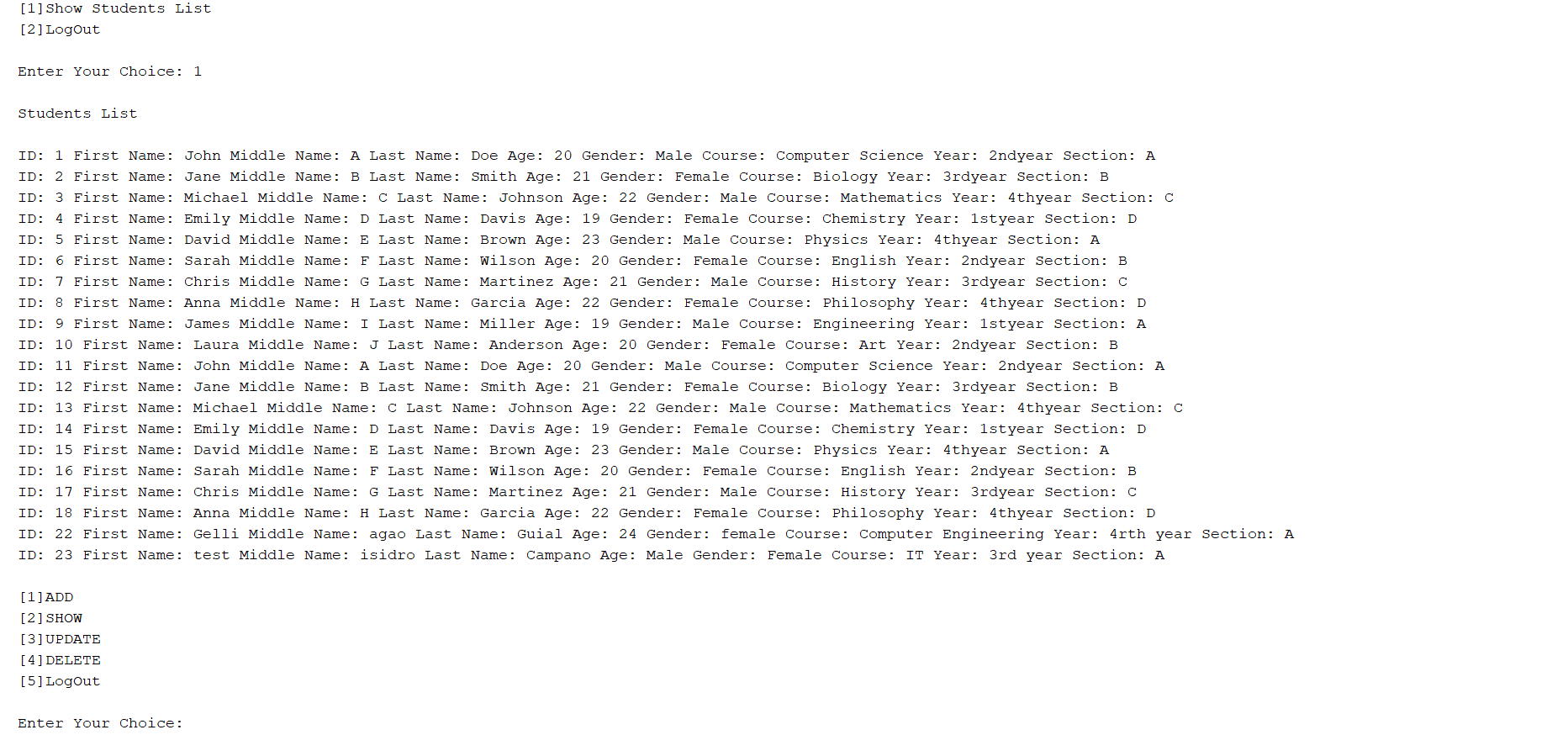
Here's the output of entering the wrong credentials for user login

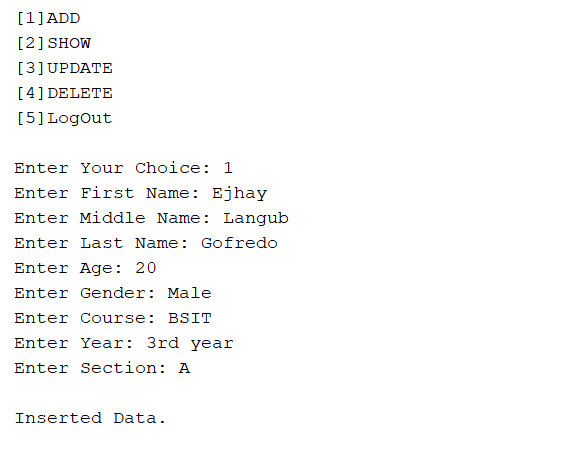
****

After entering the credentials, it will go directly to the home page and show option 1 for 'Show Student' and option 2 for 'Logout



After entering 1, it will display the list of students, and then it will show the 5 options: 1 for adding a student, 2 to show the list of students, 3 to update student information, 4 to delete a student, and 5 to logout

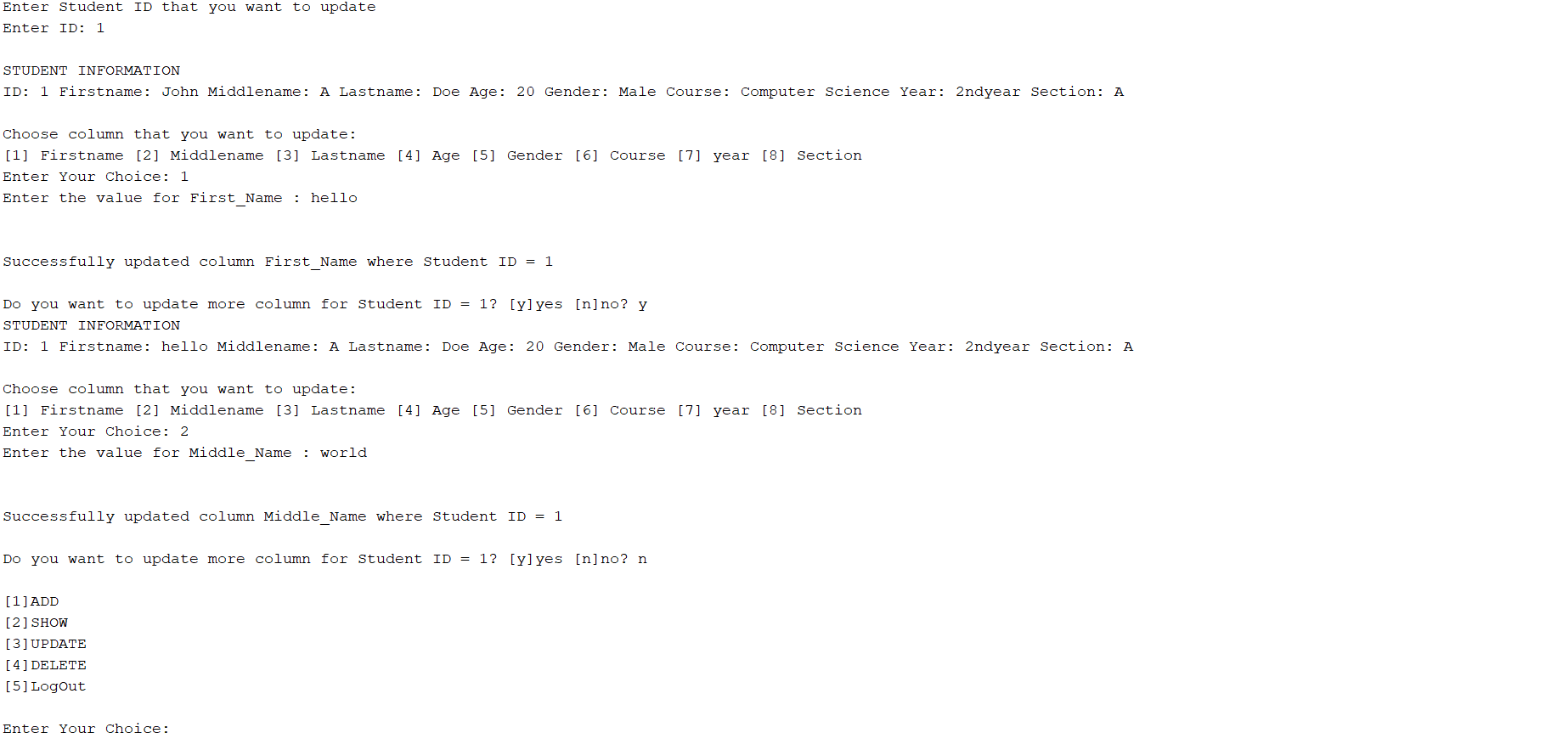


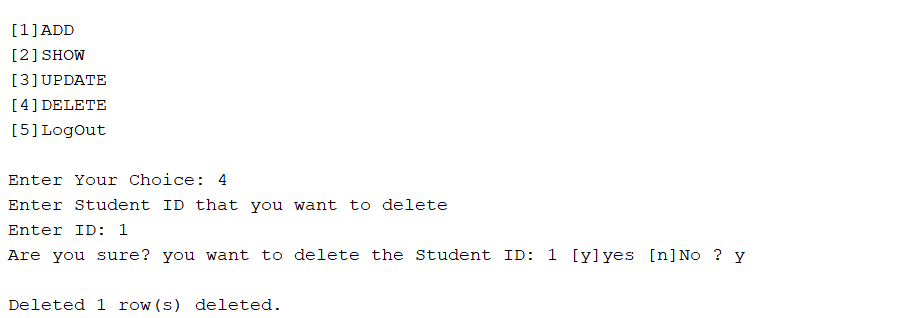
I choose option 1 to add a student, and it will display the fields where you need to input your information. Once you're done, it will message that the data has been inserted

As you can see, the ID 24 is the latest student that I have added



Choose option 3 for update, and the program will ask for the student ID that you want to update the information for. It will then show the student information based on the entered Student ID. After that, the program will ask for the column name that you want to update: [1] first name, [2] middle name, [3] last name, [4] age, [5] gender, [6] course, [7] year, [8] section. Once you choose the column, the program will ask for the new value for that column. After updating, it will send the message 'Successfully updated column name where Student ID = id'. Then, it will ask again if you want to update more columns: [y] for yes, [n] for no. If you choose 'yes', it will ask for another column to update. If you choose 'no', it will end the update session and ask for another query or HTTP method



Choose option 4 for deleting student records. It will ask for the student ID, and after you input the ID, it will ask for confirmation (yes or no) if you want to continue. If you choose 'yes', it will delete the student, and the program will send the message 'Deleted 1 row(s)'

As you can see, I chose option 2 to show or view the student list. The ID number is gone because it has already been deleted from the database

The last option is for logout. Once you choose it, the program will send the message 'Successfully logged out,' and the program will end