**DECLARATION:**

We jointly declare that the project report on **“COLLEGE MANAGEMENT SYSTEM”** is the result of original work done by us and best of our knowledge, similar work has not been submitted to **“ANNA UNIVERSITY** **CHENNAI”** for the requirement of Degree of BE. This project report is submitted on the partial fulfilment of the requirement of the award of Degree of BE.

**ABSTRACT:**

The College Management System encompasses a multifaceted framework that addresses key components such as fees structure, library management, student profiles, and mark sheets. This comprehensive system offers an integrated solution to efficiently handle financial transactions related to fees, organize and manage library resources, maintain detailed student profiles, and accurately record and generate academic mark sheets. Through a user-friendly interface, this system streamlines administrative tasks, providing a centralized platform for seamless management of diverse aspects crucial to the effective functioning of a college.

**INTRODUCTION:**

**1. \*\*Fees Structure Management: \*\***

The Fees Structure Management module facilitates efficient handling of financial transactions within the college. It automates the process of fee collection, tracks payments, generates receipts, and maintains a transparent record of financial transactions. This ensures accuracy, reduces manual errors, and contributes to a more organized and accountable financial structure.

**2. \*\*Library System: \*\***

The Library System module serves as a digital hub for managing the college's library resources. It includes functionalities for cataloging books, tracking their availability, managing check-ins and check-outs, and providing a user-friendly interface for students and faculty to search and access the vast repository of educational materials. This promotes an efficient and organized approach to library administration.

**3. \*\*Student Profile: \*\***

The Student Profile module acts as a centralized repository for storing and managing comprehensive information about each student. It includes details such as personal information, academic history, attendance records, and any other relevant data. This module simplifies the process of accessing and updating student information, offering a holistic view of each student's academic journey.

**4. \*\*Marksheet Generation: \*\***

The Marksheet Generation module automates the creation and distribution of academic marksheets. It pulls data from the academic records, calculates grades, and generates standardized marksheets for each student. This not only saves time but also ensures accuracy and consistency in the documentation of student performance, contributing to a more efficient assessment and reporting system.

**PROGRAM:**

import java.util.ArrayList;

import java.util.Scanner;

public class CollegeManagementSystem {

static ArrayList<Student> students = new ArrayList<>();

static ArrayList<FeeReport> feeReports = new ArrayList<>();

static ArrayList<LibrarySystem> librarySystems = new ArrayList<>();

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

while (true) {

System.out.println("1. Add Student");

System.out.println("2. View Student Profile");

System.out.println("3. Enter Fee Details");

System.out.println("4. View Fee Report");

System.out.println("5. Library System");

System.out.println("6. View Marksheet");

System.out.println("7. Exit");

System.out.print("Enter your choice: ");

int choice = scanner.nextInt();

scanner.nextLine(); // Consume the newline character

switch (choice) {

case 1:

addStudent(scanner);

break;

case 2:

viewStudentProfile(scanner);

break;

case 3:

enterFeeDetails(scanner);

break;

case 4:

viewFeeReport(scanner);

break;

case 5:

librarySystem(scanner);

break;

case 6:

viewMarksheet(scanner);

break;

case 7:

System.out.println("Exiting program. Goodbye!");

System.exit(0);

default:

System.out.println("Invalid choice. Please enter a valid option.");

}

}

}

private static void addStudent(Scanner scanner) {

System.out.print("Enter student name: ");

String name = scanner.nextLine();

System.out.print("Enter roll number: ");

int rollNumber = scanner.nextInt();

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

double marks = scanner.nextDouble();

students.add(new Student(name, rollNumber, marks));

System.out.println("Student added successfully!");

}

private static void viewStudentProfile(Scanner scanner) {

System.out.print("Enter roll number: ");

int rollNumber = scanner.nextInt();

for (Student student : students) {

if (student.rollNumber == rollNumber) {

System.out.println("Student Profile:");

System.out.println("Name: " + student.name);

System.out.println("Roll Number: " + student.rollNumber);

System.out.println("Marks: " + student.marks);

return;

}

}

System.out.println("Student not found!");

}

private static void enterFeeDetails(Scanner scanner) {

System.out.print("Enter roll number: ");

int rollNumber = scanner.nextInt();

System.out.print("Enter fees paid: ");

double feesPaid = scanner.nextDouble();

feeReports.add(new FeeReport(rollNumber, feesPaid));

System.out.println("Fee details entered successfully!");

}

private static void viewFeeReport(Scanner scanner) {

System.out.print("Enter roll number: ");

int rollNumber = scanner.nextInt();

for (FeeReport feeReport : feeReports) {

if (feeReport.rollNumber == rollNumber) {

System.out.println("Fee Report:");

System.out.println("Roll Number: " + feeReport.rollNumber);

System.out.println("Fees Paid: " + feeReport.feesPaid);

return;

}

}

System.out.println("Fee report not found!");

}

private static void librarySystem(Scanner scanner) {

System.out.print("Enter roll number: ");

int rollNumber = scanner.nextInt();

LibrarySystem librarySystem = findLibrarySystem(rollNumber);

System.out.println("1. Borrow Book");

System.out.println("2. Return Book");

System.out.println("3. View Books Borrowed");

System.out.print("Enter your choice: ");

int choice = scanner.nextInt();

scanner.nextLine(); // Consume the newline character

switch (choice) {

case 1:

System.out.print("Enter book name to borrow: ");

String bookToBorrow = scanner.nextLine();

librarySystem.borrowBook(bookToBorrow);

System.out.println("Book borrowed successfully!");

break;

case 2:

System.out.print("Enter book name to return: ");

String bookToReturn = scanner.nextLine();

librarySystem.returnBook(bookToReturn);

System.out.println("Book returned successfully!");

break;

case 3:

System.out.println("Books Borrowed:");

for (String book : librarySystem.booksBorrowed) {

System.out.println(book);

}

break;

default:

System.out.println("Invalid choice. Please enter a valid option.");

}

}

private static LibrarySystem findLibrarySystem(int rollNumber) {

for (LibrarySystem librarySystem : librarySystems) {

if (librarySystem.rollNumber == rollNumber) {

return librarySystem;

}

}

LibrarySystem newLibrarySystem = new LibrarySystem(rollNumber);

librarySystems.add(newLibrarySystem);

return newLibrarySystem;

}

private static void viewMarksheet(Scanner scanner) {

System.out.print("Enter roll number: ");

int rollNumber = scanner.nextInt();

for (Student student : students) {

if (student.rollNumber == rollNumber) {

System.out.println("Marksheet:");

System.out.println("Roll Number: " + student.rollNumber);

System.out.println("Name: " + student.name);

System.out.println("Marks: " + student.marks);

return;

}

}

System.out.println("Marksheet not found!");

}

}

class Student {

String name;

int rollNumber;

double marks;

public Student(String name, int rollNumber, double marks) {

this.name = name;

this.rollNumber = rollNumber;

this.marks = marks;

}

}

class FeeReport {

int rollNumber;

double feesPaid;

public FeeReport(int rollNumber, double feesPaid) {

this.rollNumber = rollNumber;

this.feesPaid = feesPaid;

}

}

class LibrarySystem {

int rollNumber;

ArrayList<String> booksBorrowed;

public LibrarySystem(int rollNumber) {

this.rollNumber = rollNumber;

this.booksBorrowed = new ArrayList<>();

}

public void borrowBook(String book) {

booksBorrowed.add(book);

}

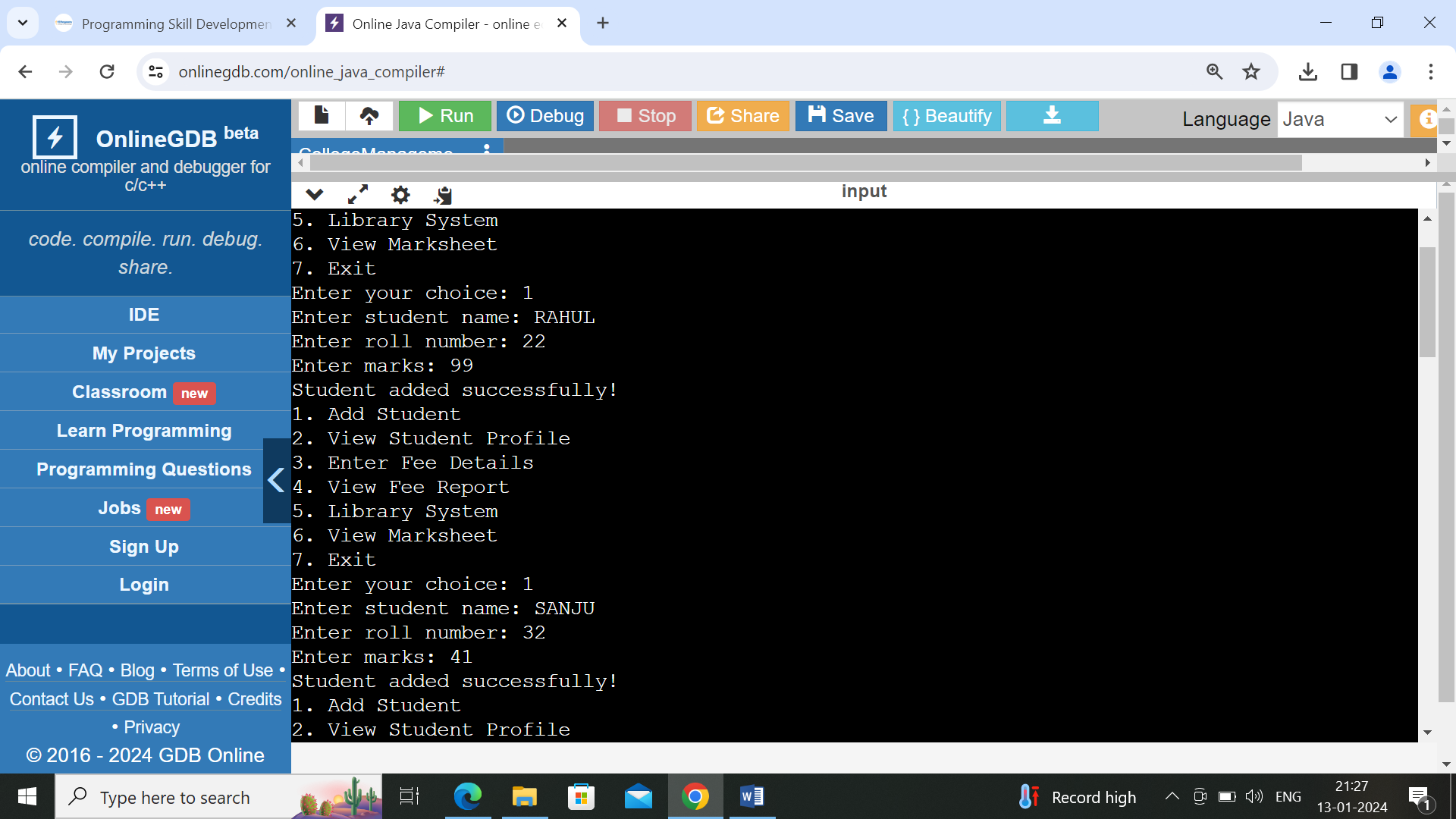
public void returnBook(String book) {

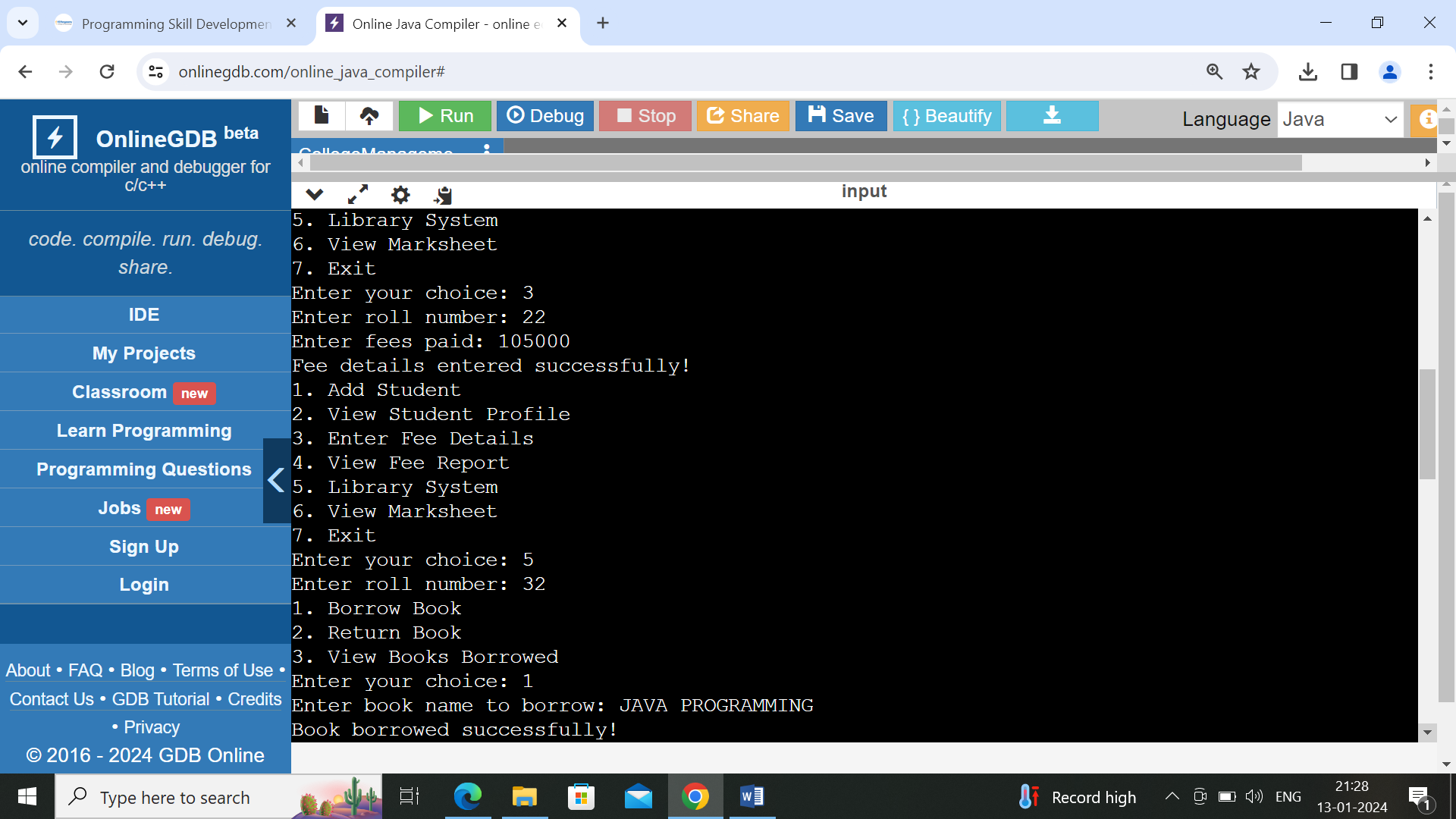
booksBorrowed.remove(book);

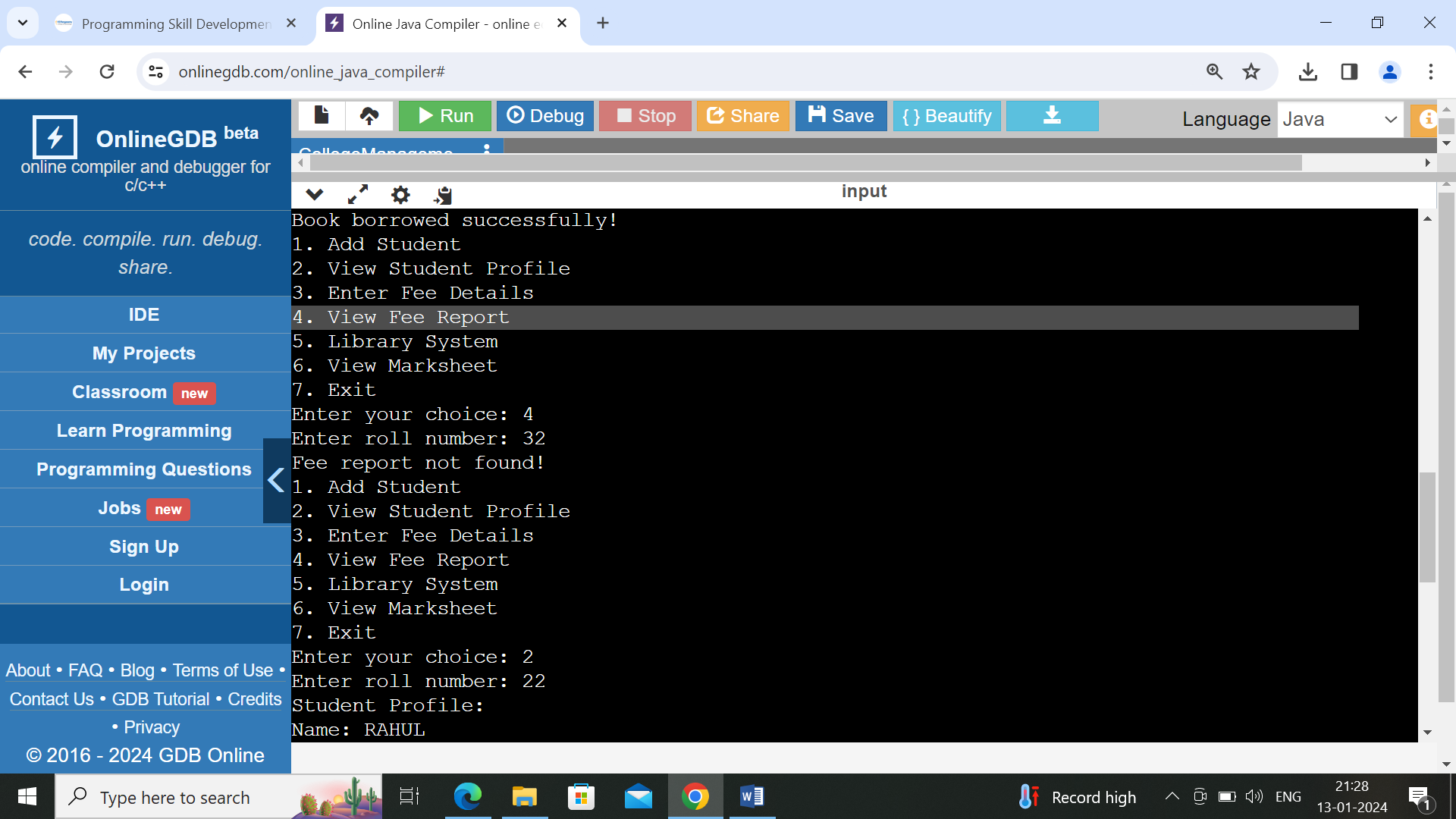
}

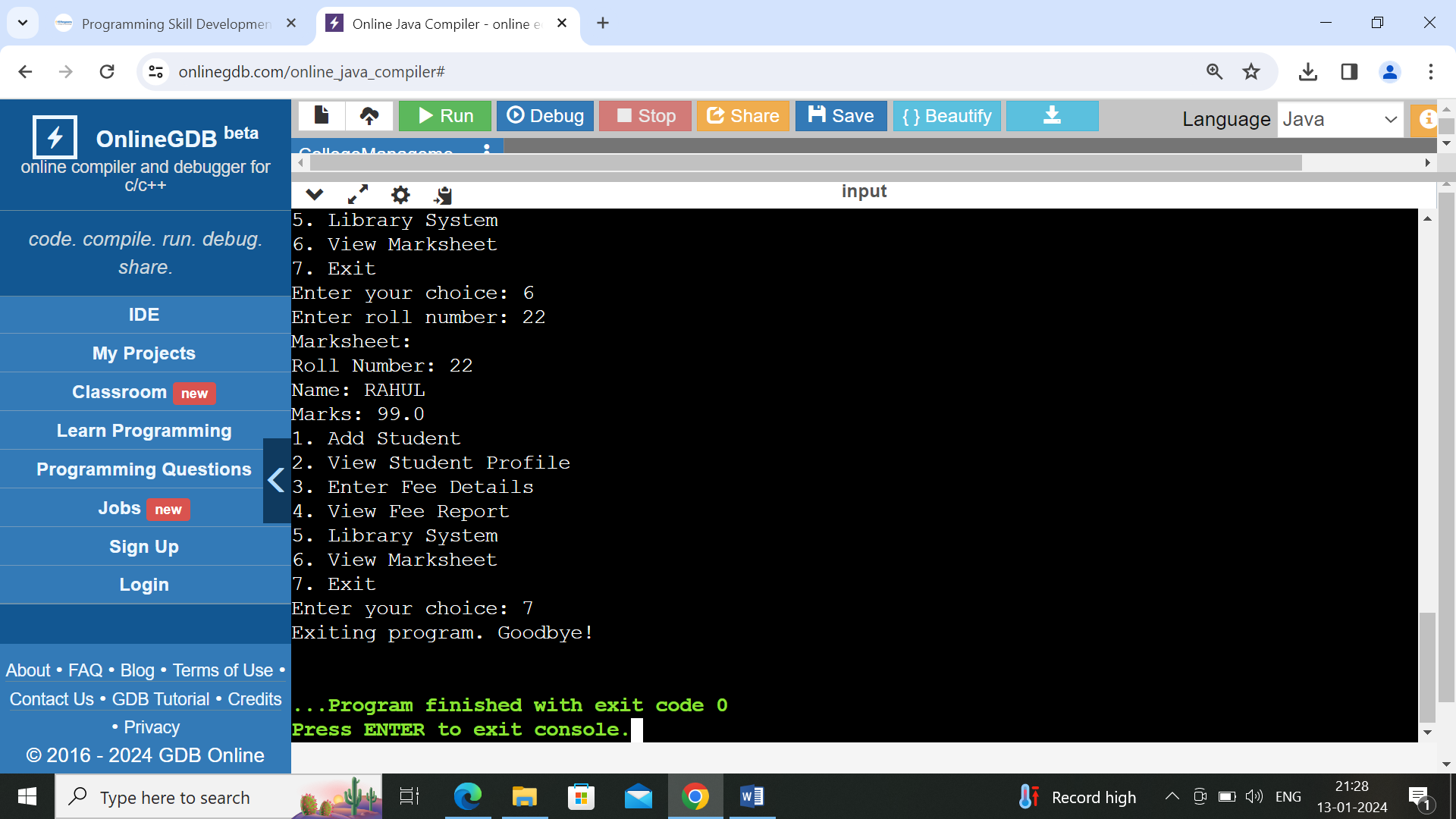
}

**OUTPUT:**









**CONCLUSION:**

The College Management System, encompassing modules for fees structure, library system, student profiles, and mark sheets, stands as a comprehensive solution to streamline and enhance administrative tasks within a college. The integration of a robust fees structure module facilitates efficient financial management, ensuring transparency and accuracy in fee-related transactions. The library system contributes to an organized and accessible repository of educational resources, promoting a conducive learning environment. The student profile module centralizes student information, fostering better communication and administration. Lastly, the mark sheet module automates the process of recording and managing academic performance, providing a reliable and efficient system for generating and maintaining student transcripts. Overall, the College Management System emerges as a valuable tool, promoting efficiency, accuracy, and systematic administration in various facets of academic management.