**Course Registration System**

UCS2265 – Fundamentals and Practice of Software Development

A PROJECT REPORT

Submitted By

PRIYA VERMA 3122235001102

NAVEEN AGARWAL 3122235001089

MONISHKUMAR BALAJI 3122235001085

A blue and white logo

Description automatically generated

Department of Computer Science and Engineering

Sri Sivasubramaniya Nadar College of Engineering

(An Autonomous Institution, Affiliated to Anna University)

Kalavakkam – 603110

June 2024

**Sri Sivasubramaniya Nadar College of Engineering**

**(An Autonomous Institution, Affiliated to Anna University)**

**BONAFIDE CERTIFICATE**

Certified that this project report titled “**Course Registration System**” is the bonafide work of “PRIYA VERMA (3122235001102), NAVEEN AGARWAL(3122235001089) and MONISHKUMAR BALAJI (3122235001085)” who carried out the project work in the UCS2265 – Fundamentals and Practice of Software Development during the academic year 2023-24.

Internal Examiner External Examiner

Date:

**TABLE OF CONTENTS**

Sl. No Content Page Number

1. Problem Statement 4
2. Extended exploration of problem statement 4
3. Analysis using Data Flow Diagrams 6
4. Detailed Design 10
5. Description of each module 16
6. Implementation 17
7. Validation through detailed test cases 19
8. Limitations of the solution provided. 35
9. Observations from the societal, legal ,

Environmental and ethical Perspectives 36

1. Learning Outcomes 37
2. References 37

Abstract

1. Problem Statement:

Develop a software system for course registration in a university where each student should log into the system to register for the chosen courses in each semester. Once the student logs in, a list of courses that are offered in that semester should be displayed along with the information about faculty members who are teaching each section of the course. Assume that each core course will have 3 sections and each elective course will have two sections. Students can choose the courses in which they would like to register. Each student should register for a minimum of 4 and a maximum of 6 courses per semester . At the end of the registration process, the list of successfully allotted courses along with the name of the teacher for each course should be displayed. In addition, the list of waitlisted courses along with waitlist number should also be displayed.

1. Extended exploration of problem statement:

Constraints:

• A student should be able to register in a particular course only if he/she has already completed all prerequisites for that course.

• Each course can be offered by multiple faculty members

• Every section will have a fixed strength. If the section fills up, further registration should not be allowed.

• A waitlist(10% of the class strength) will be maintained in case the allotment is not possible in the first round.

• If a student wants to add or drop any course after initial registration, it should be allowed only within the first two weeks after the semester starts.

• Students from the waitlist can be allotted their choice subsequent to the processing of drop request.

Inputs:

• Student-id

• Number of courses the student wishes to register for

• For each course

o Course Code

o Course Name

o Name of the Faculty Member

Outputs:

A list where each entry comprises the following:

• Course Code

• Course Name

• Name of the Faculty Member

• Allotment Status(yes/No)

• Waitlist Number if Allotment Status is “No”

Executive Summary:

The university course registration system aims to provide students with a streamlined process for selecting and registering for courses each semester. The system facilitates students in logging in, viewing available courses, selecting their preferred options, and receiving confirmation of their course allotments. The course registration system aims to enhance the student experience by providing a user-friendly interface, ensuring fair course allotment, and accommodating necessary adjustments within predefined constraints. By automating the registration process, the system optimizes administrative efficiency and facilitates effective academic planning for both students and faculty.

Various Stages of Seat Allotment:

* Login of students into couse registration framework using

unique user-id and password.

* Showcase of all the available courses for the selected

semester.

* Choicefilling of preferred courses by the student.
* Allotment of courses to the student.

1. Analysis using Data Flow Diagrams:

Level 0:

A diagram of a company

Description automatically generated

This system aims to register courses for students in an university ,where the student will be able to select for his required courses and this system will be able to allot seat or put them in waitlist based on availability.

Level 1:

A diagram of a company

Description automatically generated

This project has four main modules login where the user’s entry into the system is checked he/she is allowed in , followed by view courses module where the user will be able to view all the courses available for his particular semester and department, this module is followed by course fiilling module, in this module the student enters the courses he/she wishes to register for. Finally the course allotment module where the seats are allocated based on the student’s preferences.

Level 2:

For Login:

A diagram of a login and password

Description automatically generated

User enters login credentials (username and password).System verifies the credentials.If the credentials are valid, the user is authenticated and granted access to the system.If the credentials are invalid, the user is prompted to re-enter them or create a new account

For View Courses:

A diagram of a computer

Description automatically generated

After successful login, the system asks to enter the current semester details and displays the respective available courses.Students can browse through different courses offered for registration.

For Choice Filling:

A diagram of a company

Description automatically generated

Once users have selected their desired courses, they submit their choices to the system.The system checks for any conflicts or constraints such as time overlap or prerequisites.If there are no conflicts, the courses are added to the user's registration list.

For Seat Allotment:

A diagram of a company

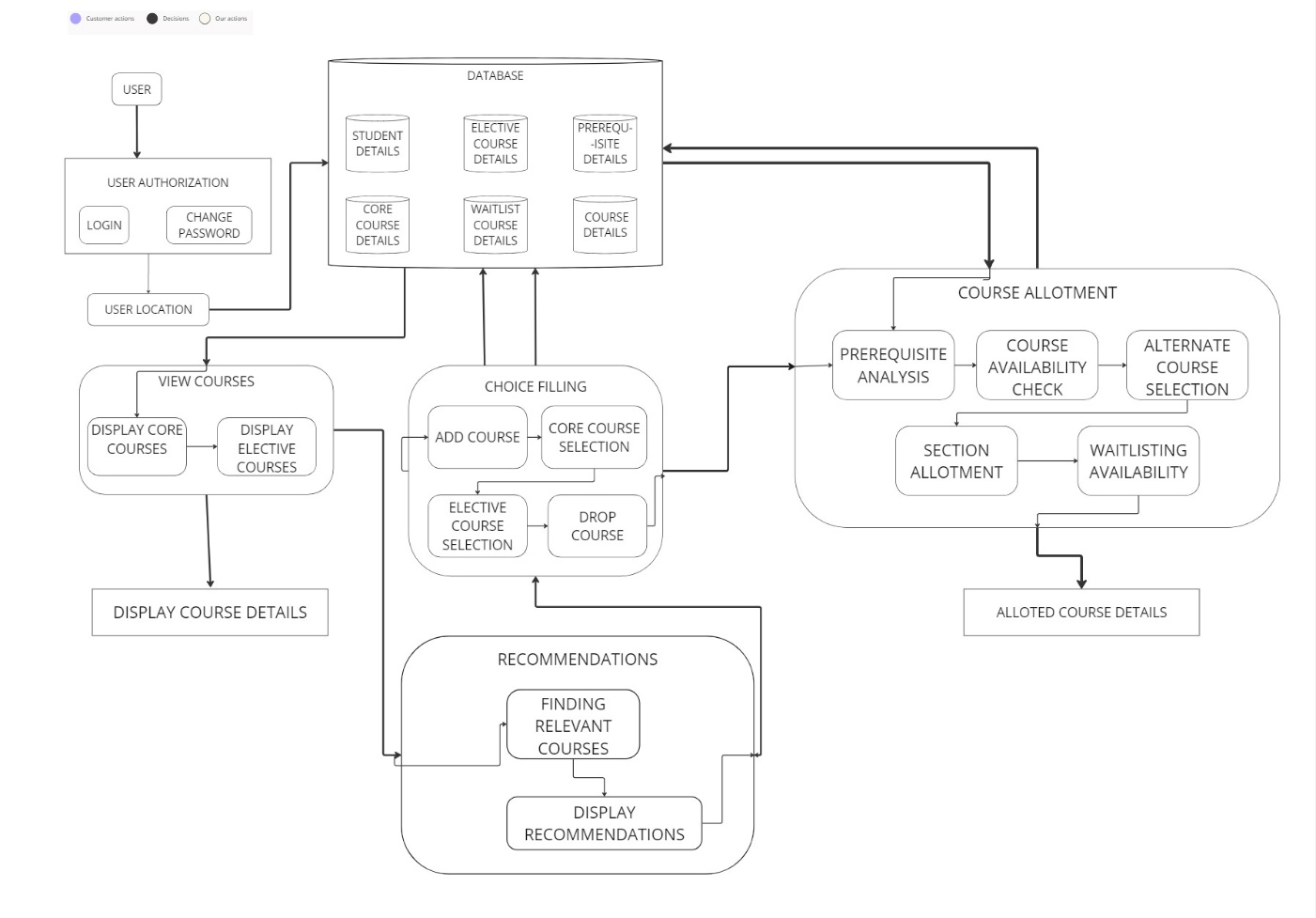
Description automatically generated

Depending on the registration process, course allotment may involve different steps,

* If it's a first-come-first-serve basis, users are allotted courses based on availability and the order in which they registered.
* If there are limited seats for a course, the system allows the students to be in waiting list. A waitlist(10% of the class strength) will be maintained .

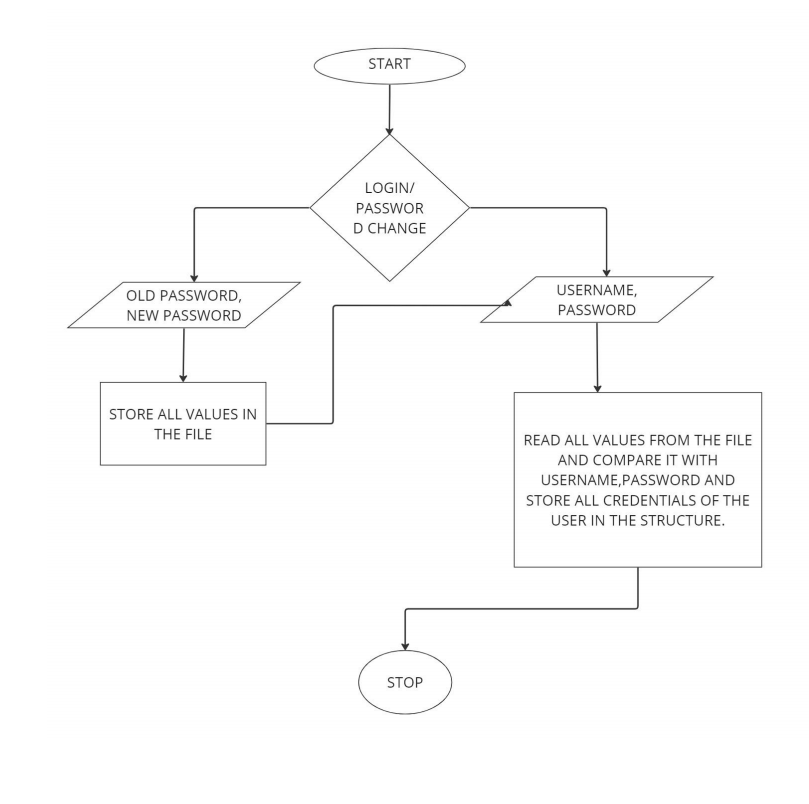
1. Detailed Design:

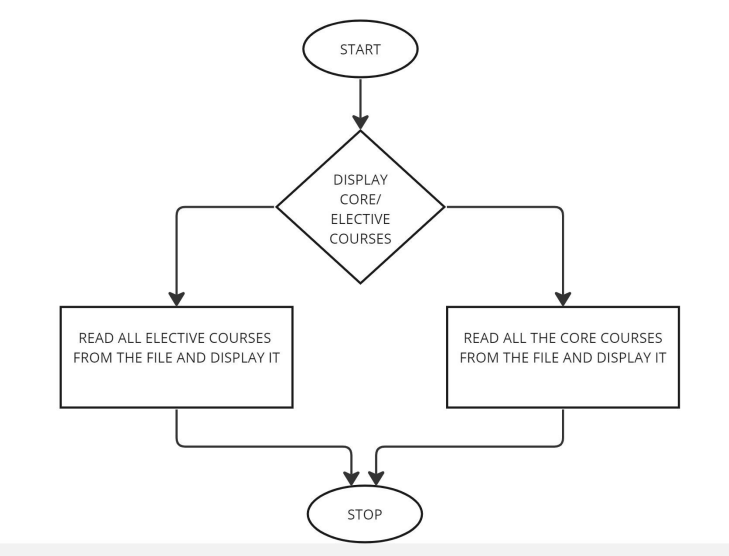
Architecture Diagram:



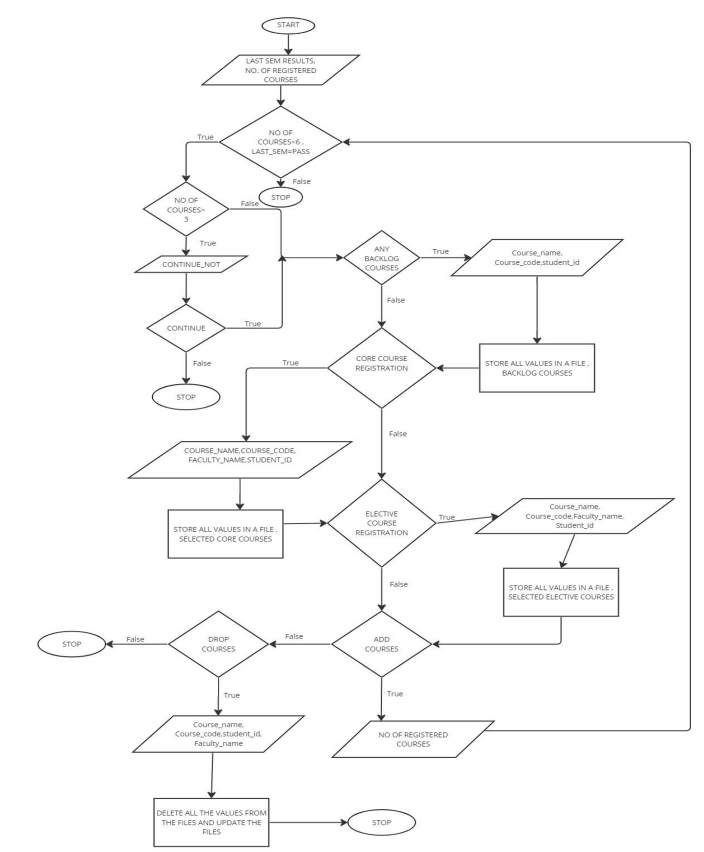
Flow Charts:

For login:



View Courses:

Choice Filling :



Seat Allotment:

A diagram of a flowchart

Description automatically generated

1. Description for Each Module:

Login:

The login module is used for validating the student’s username and password, this module has all the usernames and that password in a file, when a user enters their username and password,this module checks whether the details match with each other and allows the user into the system.

View Courses:

This module takes semester number and department as inputs and displays all the courses available for that particular semester and department . This module stores all the couses name, course id and faculty names in a structure and displays it on request.

Choice Filling:

After seeing all the courses that are available , the students enters all the courses that he or she wishes for. The students can also drop the courses that are not required though they will be allowed to do so only within 2 weeks from the start of the semester.Students can register for both core and elective courses.

Seat Allotment:

Based on the courses and faculty names(both core and elective) selected by students in the previous module,seats are allocated for those students , if the seats are not available the students are put into waiting list.

6. Implementation:

a). Explanation of how the data is organized and the Rationale behind the selection of a particular language construct.

Data is stored in the text as well as binary files. In text files, the data records are separated by comma and can be read by user as it is in human readable format. While in binary file, user can not see the data records format but can be read using fread and can be written by fwrite. As for C programming, writing data and reading records from the files, binary files are much easy to handle and use.

* Arrays : We have used arrays for storing similar data types items in consecutive address. This helped in storing the data records.
* Arrays of Structures: We made use of arrays of Structure for storing records of Student , Courses details and the selected course details.

List of Arrays of Structures used in this system-

* + - Student details
    - Core Courses details
    - Elective Courses details
    - Core selected courses
    - Elective selected courses
    - Backlog Courses
    - Final Allotment details
    - Waitlist courses details
* Structures : We have used structures in reading the file record by record and then doing the changes according to the need . Then storing the records back to the file.
* Files: For storing the record details like Student details, Courses selected and final allotment details. It makes easy for complex program and big data.

List of files used in this system are-

* + - “students.txt”
    - “Core\_details.bin”
    - “Elective\_details.bin”
    - “Student\_details.bin”
    - “Waitlist.bin”
    - “Allotment.bin”
    - “core\_course.txt”
    - “elective\_courses.txt”
    - “core\_selected.bin”
    - “elective\_selected.bin”
    - “backlog\_courses.bin”
    - “ConfirmedWaitlist.bin”

b) Explanation of any other libraries or APIs that have used: Not used.

c) User Interface design: Not any particular GUI but we tried using some of the better display method while showing output. Like printing == few places to separate from others and highlights the contents. Else using the new line to make more clear and avoid the ambiguity in the output.

d) Platform Used for Code Development : Visual Studio Code (VS Code)

7. Validation through Detailed Tests cases for various scenarios (input, expected output, Actual Output)

For login :

Input –

* Username and
* Password

Expected Output-

* Successful Login.

Actual Output-

* Successful Login,
* Username not found,
* Wrong password.

For View Courses:

Input-

* Semester,
* Department,
* faculty name.

Expected Output-

* List of Courses with course code,
* course name , and
* few faculties name.

Actual Output-

* List of Courses with course code,
* course name , and
* few faculties name.

Choice Filling:

Input –

* Any backlog course if so how many and backlog course code, course name and faculty name,
* total number of courses to register,
* number of core courses to register,
* Course Code, Course name, Faculty name.
* Any extra courses to register.
* Any drop courses to do.

Expected Output-

* Courses added successfully

Actual Output-

* Courses added successfully

Course Allotment:

Input-

* If course not available, want to go for waitlist or alternate course.
* If course for preferred faculty not available , want to go for alternate or waitlist.

Expected Output-

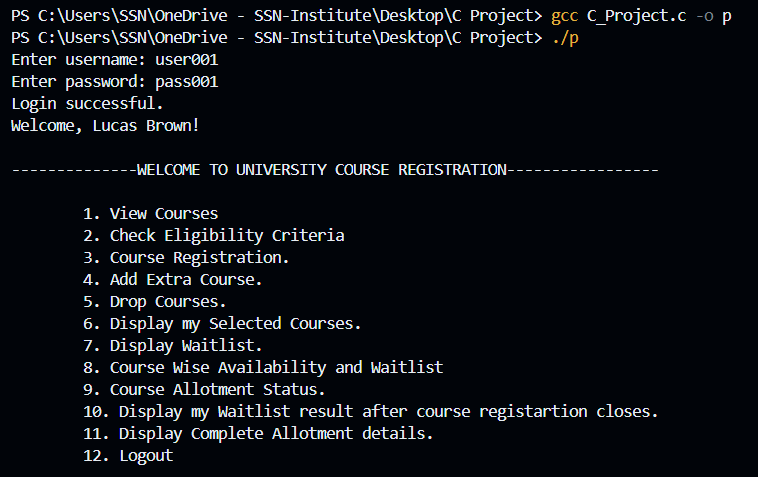
* Courses are available for preferred faculty and allotted successfully.
* Courses are not available for preferred faculty but allotted with alternate faculty.
* Courses added for waitlist.

Actual Output-

* Courses are available for preferred faculty and allotted successfully.
* Courses are not available for preferred faculty but allotted with alternate faculty.
* Courses added for waitlist

Screenshots of all possible test cases with input and output:

Login Successfully.



Checking eligibility criteria whether the course is still open based on the count of days between today and semester start date.

A computer screen shot of a course registration

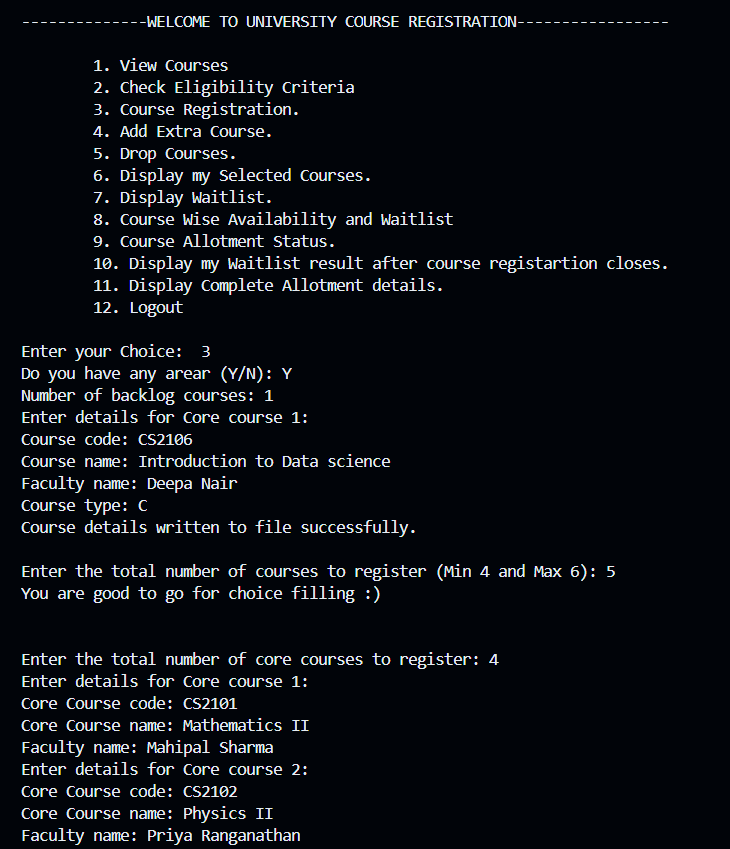
Description automatically generated

A computer screen with white text

Description automatically generated

Doing course registration , in which first asking for any backlog courses the student have. If so then adding the backlog course in a file and then proceeding for the course registration.

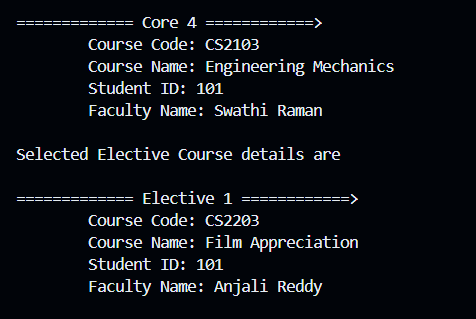
Entering the number of backlog courses and then adding the course details. Then entering the total number of courses to register and the number of core courses and elective courses. For each courses the user enter Course Code, Course Name, and Faculty name that the user wants. Finally the course details are written successfully to the file.



A computer screen shot of a course

Description automatically generated

Displaying my selected courses that is 4 core courses and 1 elective courses with full details about the courses.



Adding the extra courses that is one more elective course the user added to the file. So, for now the total number of selected courses are 6.

A computer screen shot of a program

Description automatically generated

A screenshot of a computer

Description automatically generated

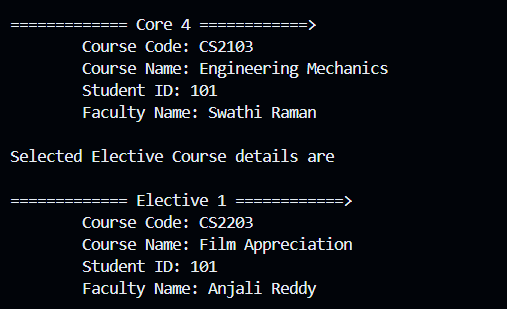
A computer screen shot of a course

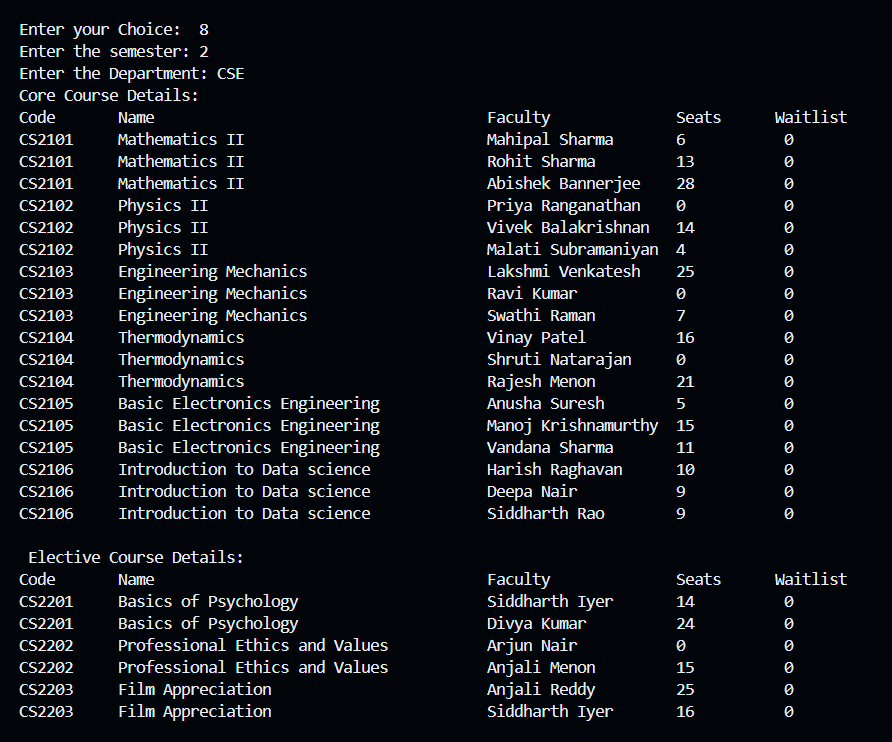
Description automatically generated

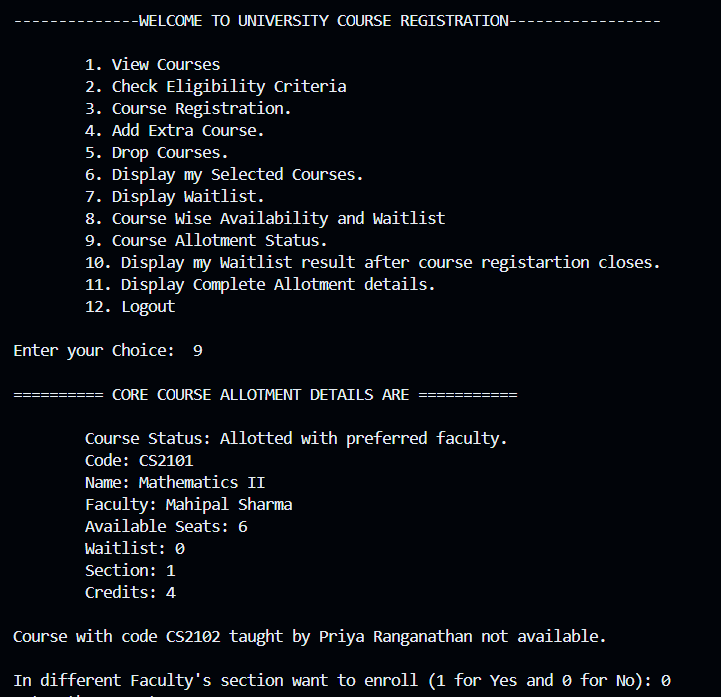
Deleting the course from the file . Now the selected courses are 5 i.e. 4 Core courses and 1 Elective Courses.

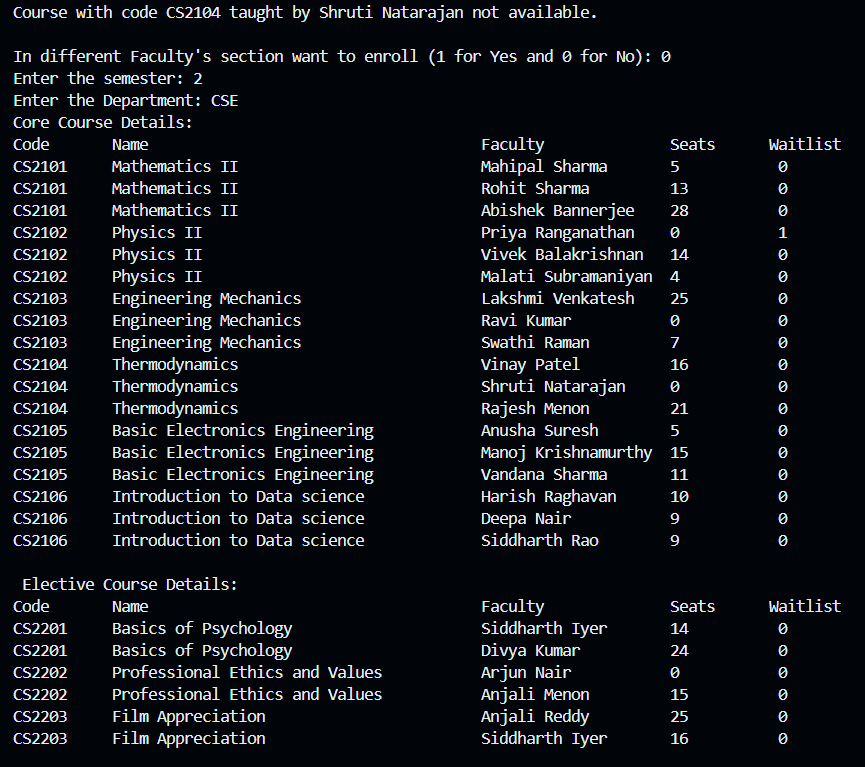
A screenshot of a computer program

Description automatically generated



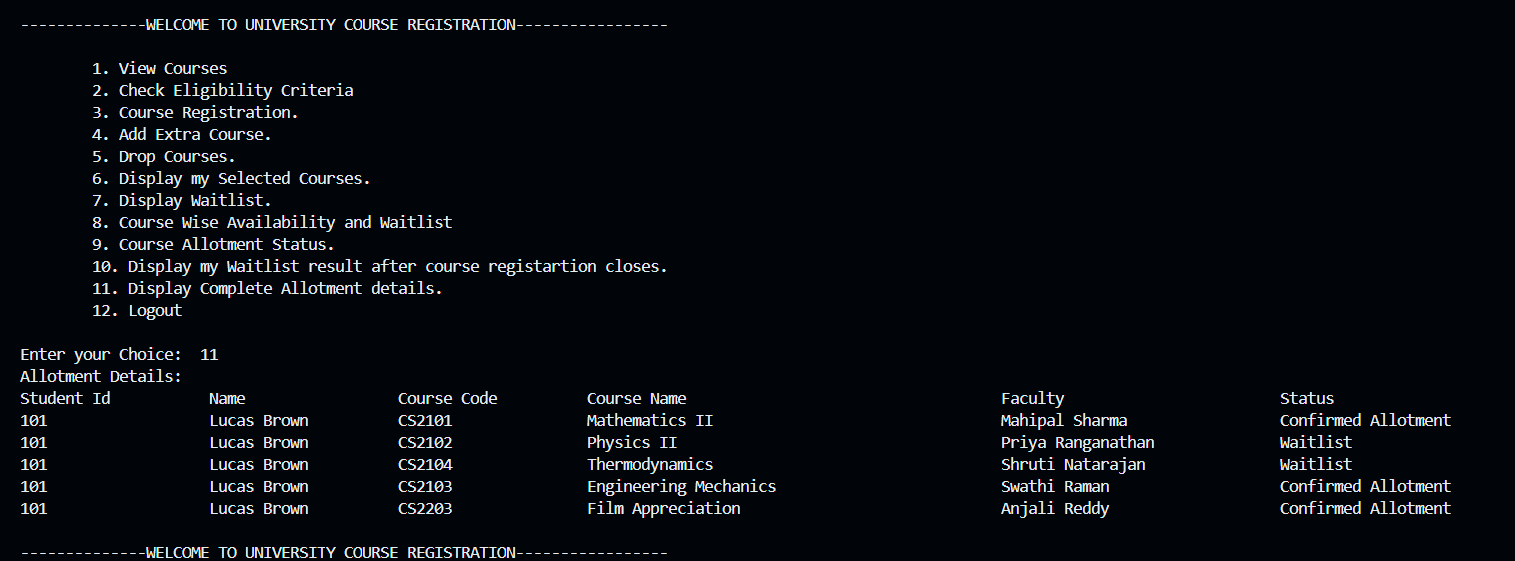
Displaying the course wise availability based on the semester and the department with the waitlist. 

This the course allotment process . First core course is available with the preferred faculty, so it is allotted successfully to the student. While the second course with CS2102 is not available with preferred faculty so asking for alternate faculty or not. The student enters no as the choice . Then asks for alternate course or waitlist. Here the student with id 101 put the course for waitlist. Similarly next core course also put for waitlist. And the rest 3 courses are allotted successfully to the student with preferred faculty and is displayed with all the details of section, credits, waitlist and available seats.A screenshot of a computer

Description automatically generatedA computer screen shot of a course

Description automatically generated

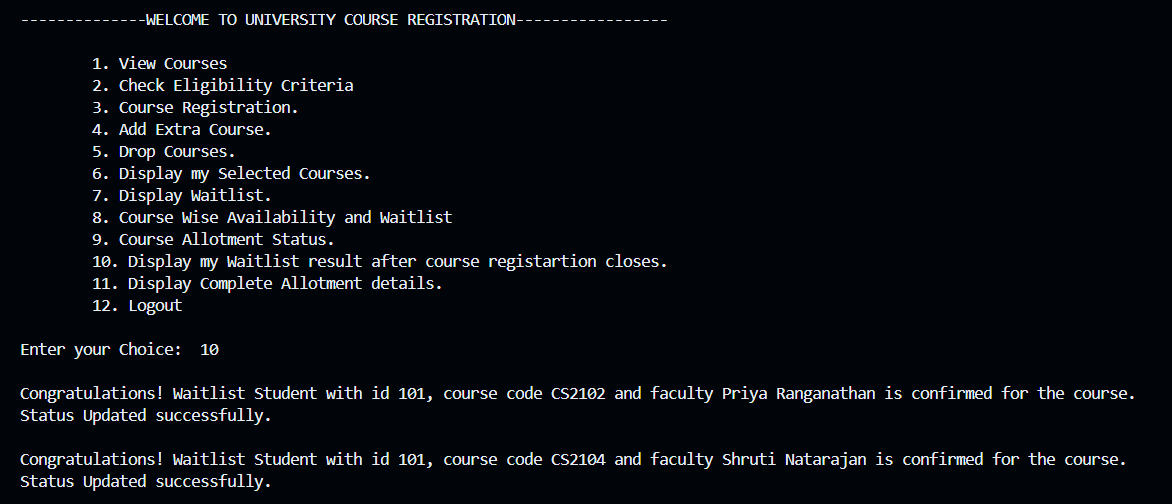
The choice 11 shows all the details of allotment done so far with the status whether the student with id 101 and name Lucas Brown is allotted respective course or is in the waitlist.



Here the choice 7 shows the waitlist details of the students. The student with id 101 has backlog and the last semester CGPA is also displayed with course code and faculty name.A screen shot of a computer

Description automatically generated

The choice 10 do the allotment of waitlist courses after the registration closes. The maximum waitlist that can be put is 3 .Here both the courses are confirmed the allotment .



This is the final display of allotment details of the users.

The student logs out.

A computer screen shot of a black screen

Description automatically generated

8.Limitations of the solution provided.

* + Lack of graphic user interface(GUI) might be difficult for the user to use the system.
  + Manually typing the courses required rather than just clicking might be inconvenient for the users

9. Observations from the Societal, Legal, Environmental and Ethical perspectives:

* The system is accessible to all users, including those with disabilities. Consider usability for elderly users as well.
* The system is same to all user groups, including students from different cultural backgrounds and varying levels of technological literacy.
* This system promotes electronic communication and documentation to reduce paper usage for registrations and related administrative tasks.
* This system ensures fairness in course allocation and registration processes to avoid favoritism or discrimination based on any criteria.

10.Learning Outcomes:

* We were able to understand how teams work on different projects and carried out this project successfully.
* We learned how to logically approach a problem and come up with the optimal solution.
* We learned how to approach a project from the scratch and slowly complete the project eventually.
* We learned how identify different modules involved in the project.
* We were able to learn how to do desired operations and functionalities in C language .

11.References:

* <https://www.geeksforgeeks.org/c-programming-language/>
* <https://www.codementor.io/c-programming-experts>
* <https://www.w3schools.com/c/>