

# **SOFTWARE SYSTEM FOR UNIVERSITY COURSE REGISTRATION**

UCS2265 – Fundamentals and Practice of Software Development

## **PROJECT REPORT**

Submitted By

Lakshman Vijay VI 3122235001072

Meghna Manimaran 3122235001079

Rahul Malaikani 3122235001103



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 “**Software system for university course registration**” is the bonafide work of “Meghna Manimaran 3122235001079, Rahul Malaikani 3122235001103 and Lakshman Vijay 3122235001072” 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: 21.06.2024

## TABLE OF CONTENTS

Content	Page Number
Abstract	
1. Problem Statement	5
2. Exploration of problem statement	6
3. Analysis using Data Flow Diagrams & Overall Architecture Diagram	7
4. Description of each module	10
5. Implementation	18
6. Validation through Detailed Test Cases for various scenarios	25
7. Limitations of the solution provided	34
8. Observations from the Societal, Legal, Environmental and Ethical perspectives	35
9. Learning Outcomes	36
10. References	37

## **ABSTRACT**

The "Course Registration system" is a C language program that facilitates the process of choice filling and seat allocation in educational institutions, by the Joint. The program aims to efficiently allocate seats to students based on their ranks and preferences for various courses.

The code utilizes csv files and data structures like arrays and structures to store student information, college details, and seat availability. It reads input data from external files, such as rank lists and seat availability records, and processes this data to allocate seats to eligible students.

The main components of the code include functions to read and display the preferred courses of students and functions to perform the seat allocation process based on their grade points and ranks, and functions to update seat availability after each allocation. The code also handles scenarios where students may decline allocated seats or change their course preference list before going on to the seat allotment process.

Additionally, it seeks to maintain fairness and transparency in the allocation process, adhering to the principles of merit-based admissions.

## 1. Problem Statement

Develop a software system for taking care of course registration in a university. This system should have the following features:

- 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 the waitlist number should also be displayed.

## 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 requests.

## Input

- Student-id
- Prerequisites
- For each course
  - Course Code
  - Course Name
  - Name of the Faculty Member

## Output

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”

## **2. Exploration of Problem Statement**

### **Ranking Score:**

During the first semester, all students will have common courses.

For registration of courses in the upcoming semesters, the criteria for ranking students could be done based on their prerequisite score, which is calculated using the cgpa and attendance of the student.

$$\text{Ranking score} = 80\% \text{ CGPA} + 20\% \text{ attendance}$$

### **Credits constraint**

The student is allowed to pick any number of courses from 4 to 6 while at the same time maintaining a total sum of credits from 10 to 15.

The core courses carry 4 credits while the electives carry 2 credits.

For example,

A person can pick one core course and 3 electives (4+2+2+2) or two core courses and 3 electives (4+4+2+2+2) or Any other combination of their choice.

Once these preferences are procured, seat allotment is done based on seat availability and merit. The allotted seat is displayed, and the user is asked to either accept or decline the seat. If accepted, the seat is confirmed for the candidate. If declined, the remaining seats are compiled and are used for the next round of counseling to ensure that all the seats are filled.

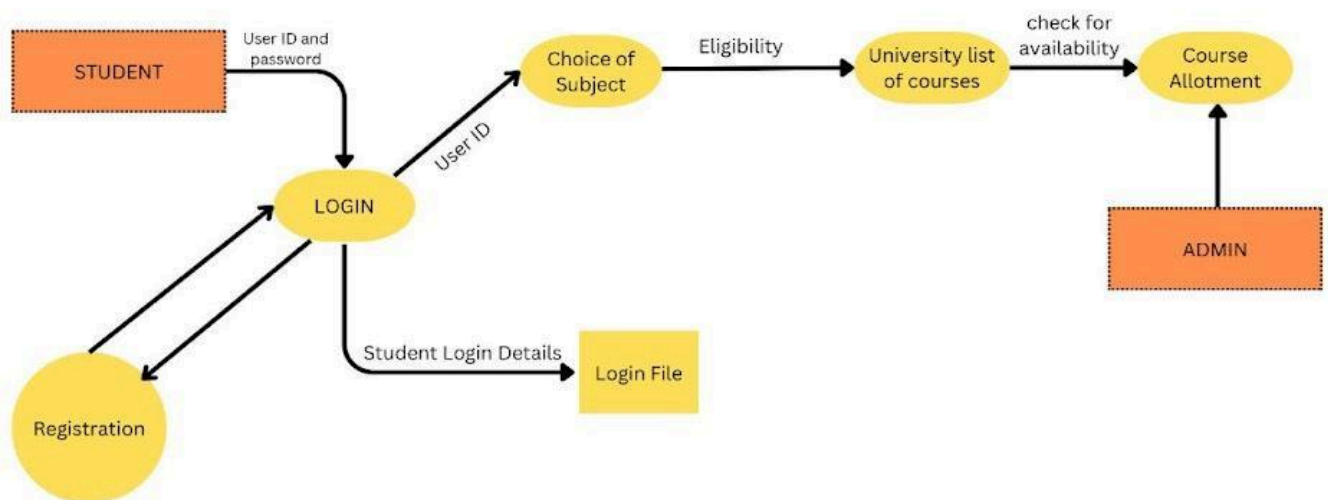
### 3. Analysis using Data Flow Diagrams & Overall Architecture Diagram

#### Data Flow Diagrams

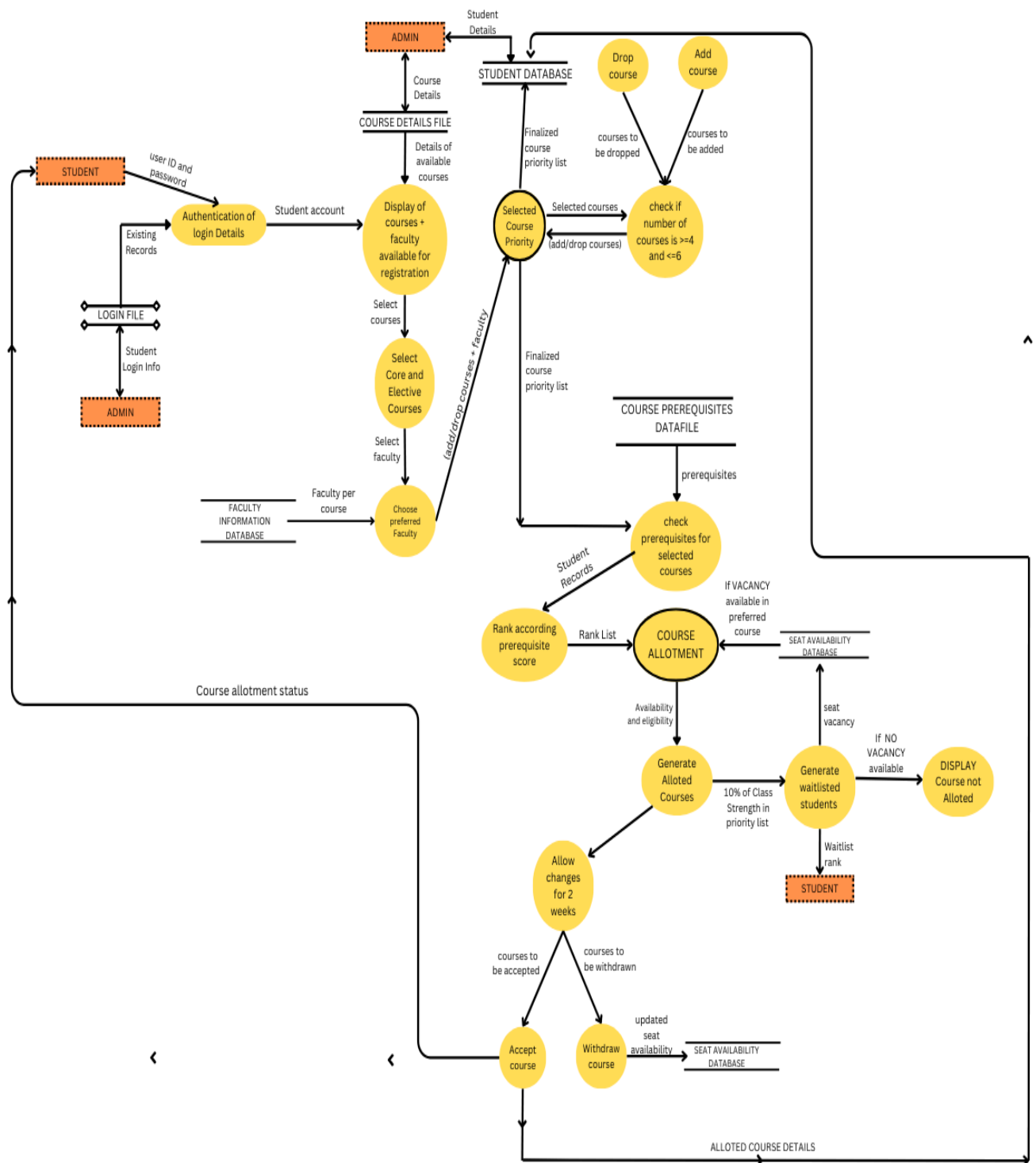
##### LEVEL 0:



##### LEVEL 1:



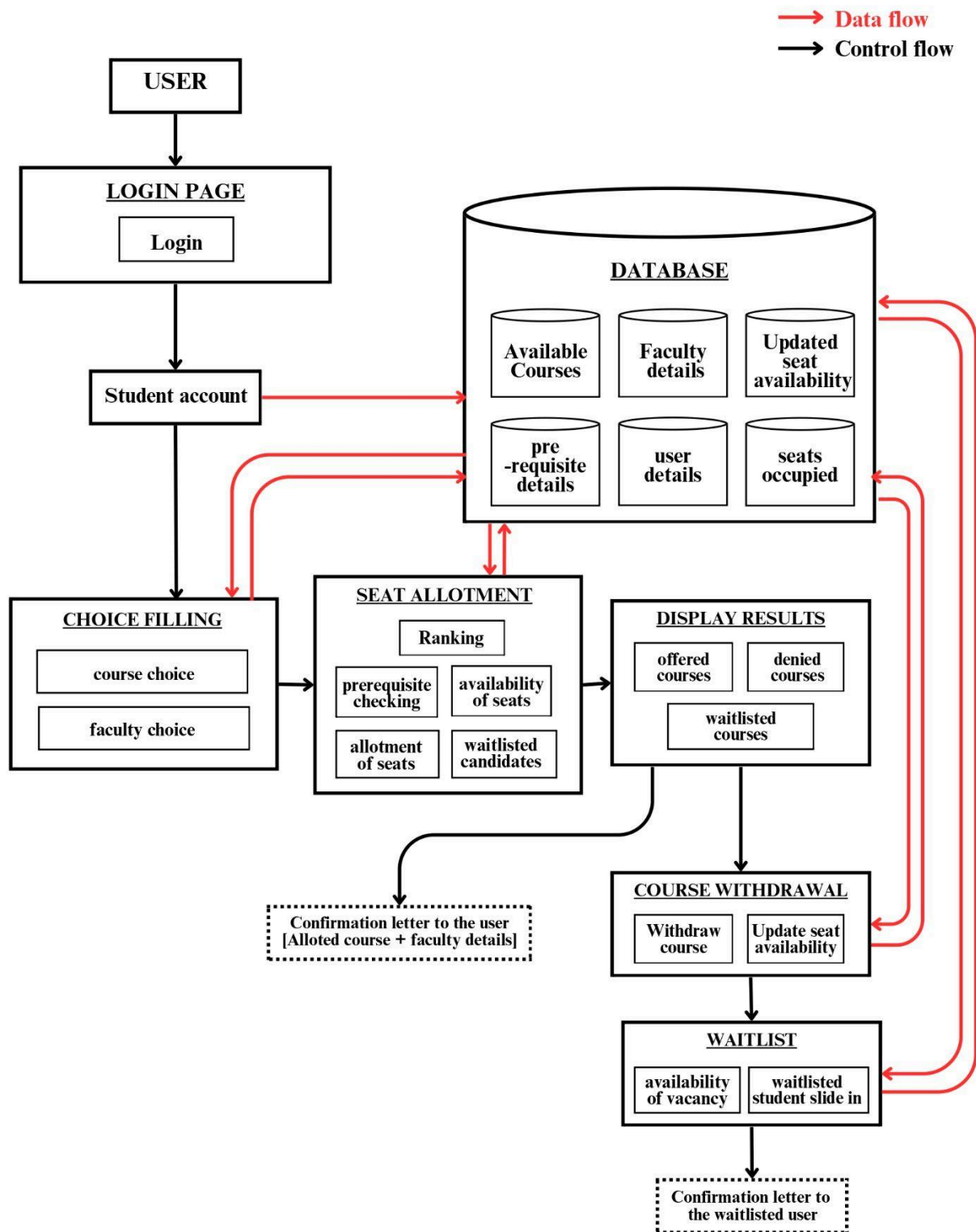
## LEVEL 2:





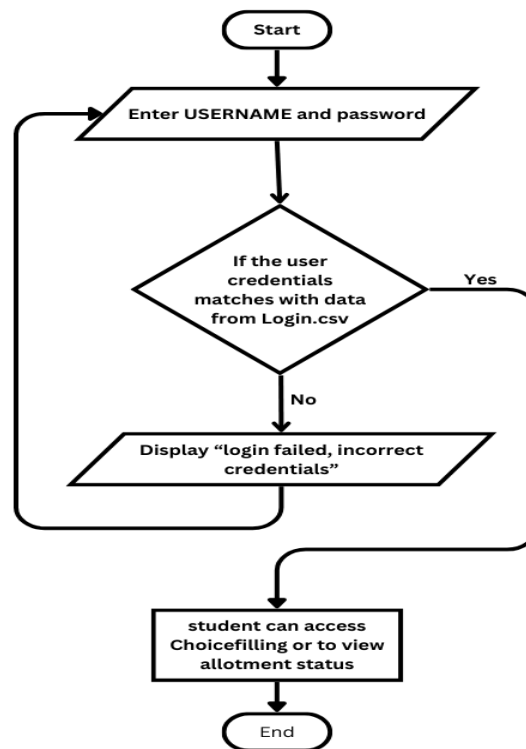
## Overall Architecture Diagram

### SYSTEM ARCHITECTURE



## 4. MODULES

### LOGIN MODULE



#### Explanation:

The program consists of functions to register and authenticate users using a CSV file for data storage.

This function authenticates a user by prompting for a username and password while opening "Login.csv" in read mode.

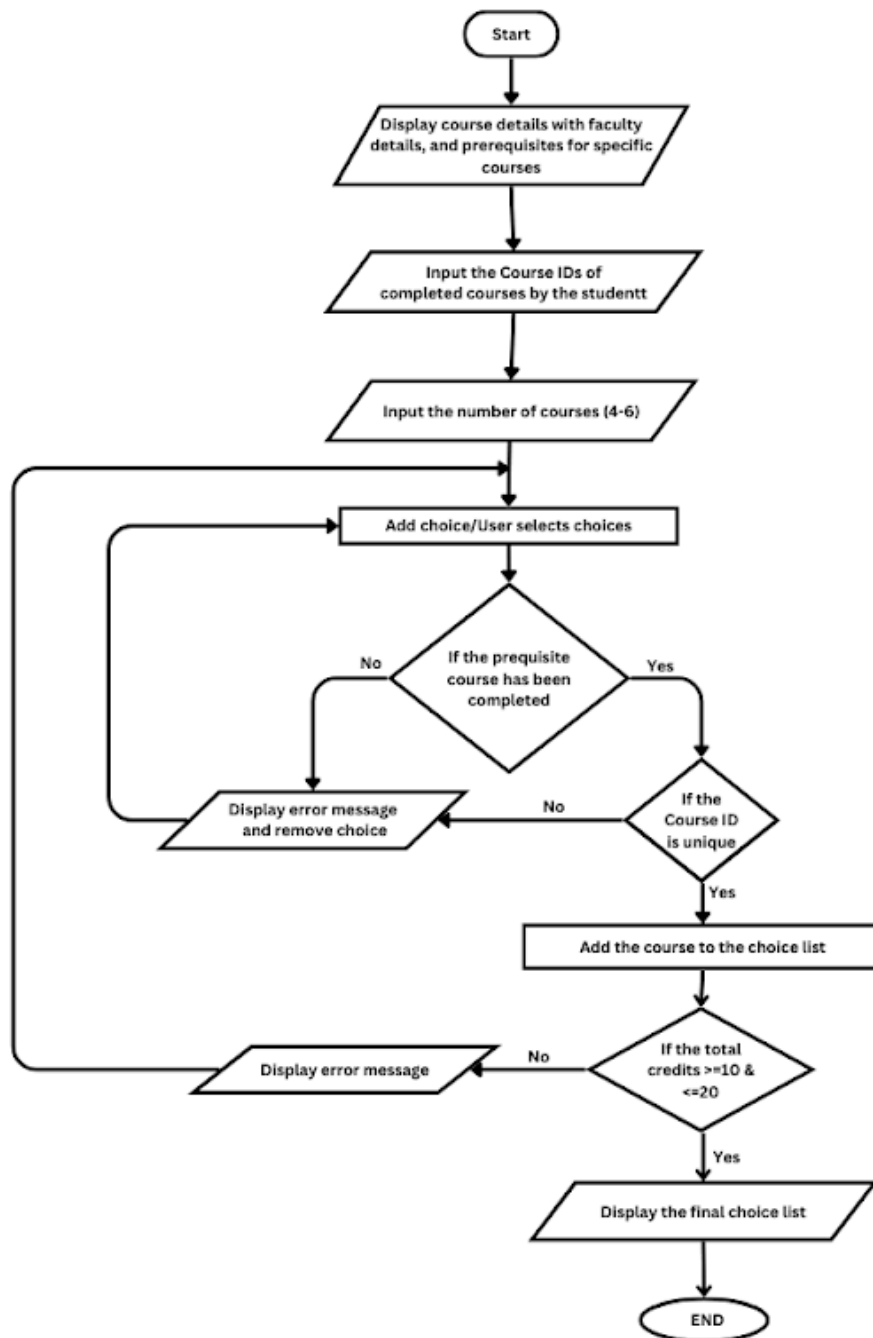
It reads each line, splitting it into username and password. It then compares the entered credentials with those in the file.

If a match is found, it returns 1 and prints "Login Successful!", otherwise returns 0 and "Login Failed!".

It repeatedly prompts the user for a choice and executes the corresponding function until the user chooses to exit.

This program ensures basic user authentication through a simple file-based approach.

## CHOICE FILLING MODULE



## Explanation:

The main function does the course selection process by reading course data from a CSV file, displaying available courses, and allowing the user to select courses based on prerequisites and credit limits.

It opens "Courses.csv" to read course details. It then reads and parses each line from the CSV, skipping the header.

Then all available courses are displayed using **display\_courses(struct Course \*courses, int num\_courses)**.

It then prompts the user to enter completed courses, number of courses and the ID of what they want in the upcoming semester.

**parse\_course(char \*line, struct Course \*course)**

This function parses a CSV line and stores the data in a Course struct. It uses **sscanf()** to extract fields from the line and assigns them to the corresponding struct members.

It iterates through the courses array and prints each course's details.

It validates the input, checks for duplicate selections, and verifies prerequisites using **check\_prerequisites()**.

**check\_prerequisites(struct Course course, char \*completedCourses)**

This function checks if a student has completed the prerequisites for a given course. It checks each prerequisite against the completedCourses string. It returns 1 if all prerequisites are met, otherwise 0.

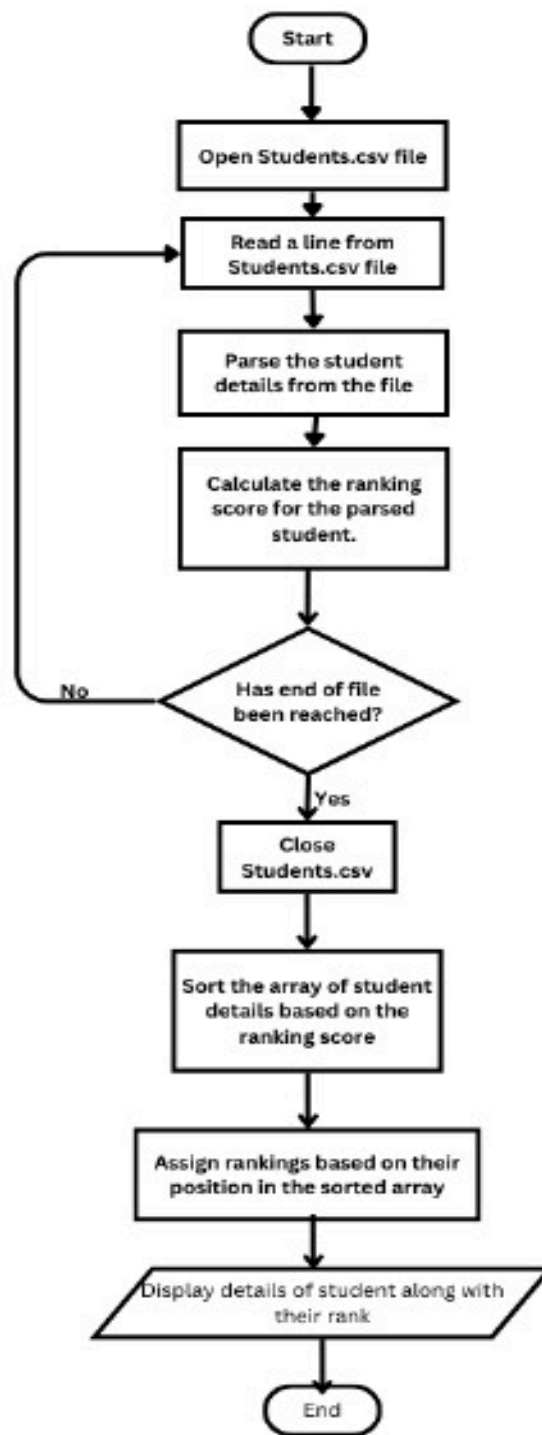
It displays selected courses and ensures that total credits fall within the specified range (10 to 20).

It asks for confirmation of the chosen courses.

It prompts the user to choose again if desired.

The program provides a robust framework for course selection, ensuring students meet prerequisites and adhere to credit limits, thereby facilitating a structured and error-free registration process.

## SEAT ALLOTMENT MODULE



## **Explanation:**

The main function coordinates the entire process of reading student and course data, calculating ranking scores, updating course choices, and allocating courses. It also generates the output files for course allotments and remaining seats. It opens and reads the "Students.csv" file, parses each line into StudentDetail structs and stores them in an array.

### **1. Reading and Parsing Students:**

- Opens "Students.csv".
- Skips the header line and parses each subsequent line into StudentDetail structs.

### **2. Calculating and Sorting Rankings:**

- Calculates ranking scores.
- Sorts students based on ranking scores using qsort().
- Assigns rankings.

### **3. Updating Course Choices:**

- Reads and updates choices from "choicelist.csv".

### **4. Reading and Parsing Courses:**

- Opens "seats.csv".
- Skips the header line and parses each subsequent line into Course structs.
- Closes the file.

### **5. Course Allocation:**

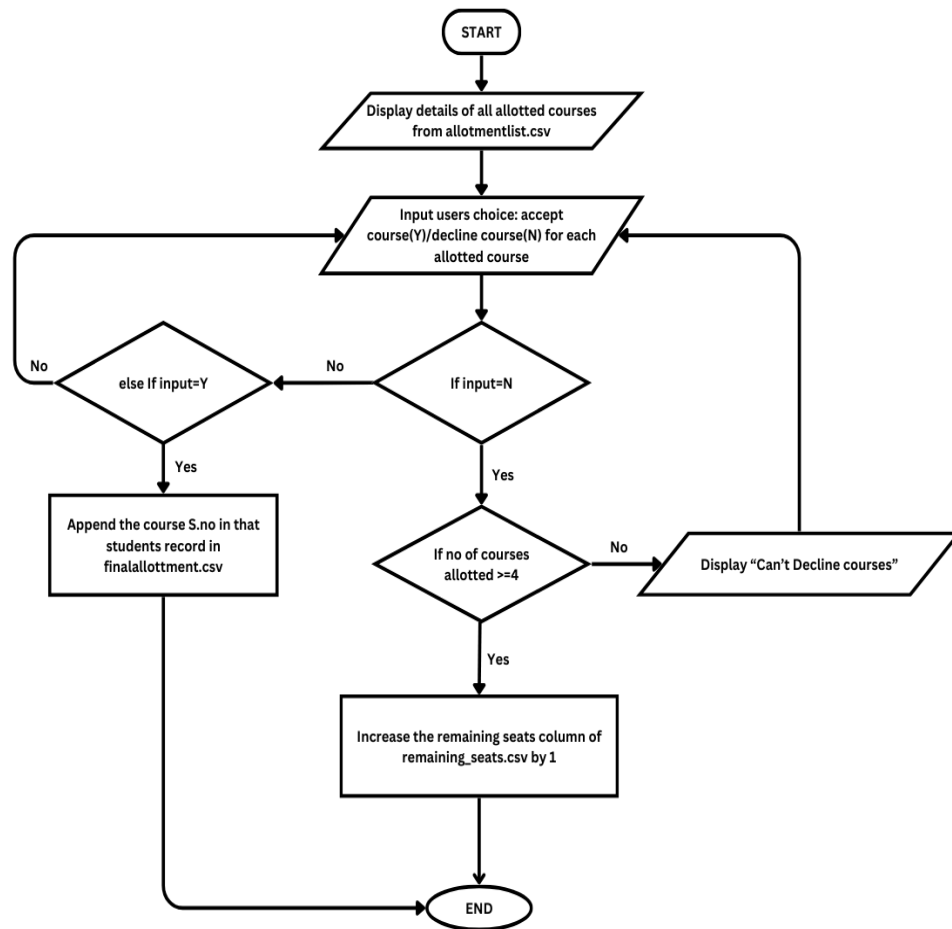
- Iterates through students in ranking order.
- Allocates courses based on choices and availability.
- Updates remaining seats.

### **6 Writing Output Files:**

- Writes the course allocation results to "allotmentlist.csv".
- Writes the remaining seats to "remainingseats.csv".

This program ensures a fair and systematic allocation of courses to students based on their preferences, ranking scores, and seat availability.

## WITHDRAWAL MODULE



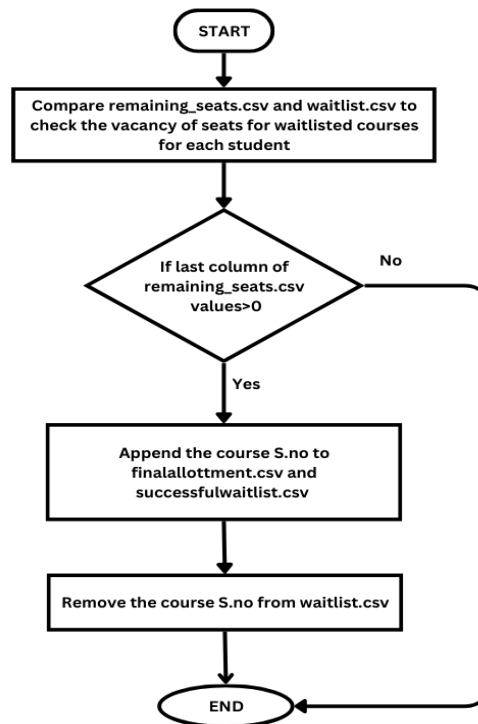
### Explanation:

#### Student Course Processing

- The main function processes each student's course withdrawals.
- It displays the student's allotted courses and asks the student whether they want to accept each course.(User interaction)
- Students can only decline a limited number of courses (minimum number of courses constraint which manages error handling).
- For each accepted or declined course, the updateSeats function updates the seat availability accordingly.
- After processing all students, the main function writes the updated course information back to seats.csv.

This approach ensures a systematic management of student course withdrawals while maintaining data integrity and adhering to constraints.

## WAITLISTING MODULE



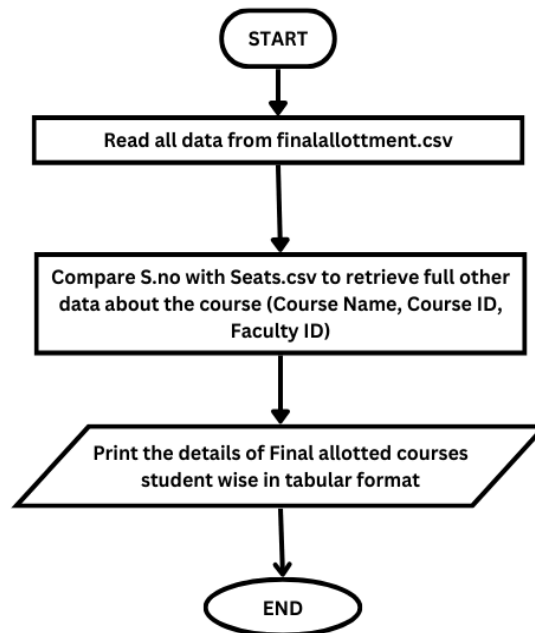
### Explanation:

- Reads course data from remainingseats.csv and populates the courses array using parse\_course. Uses sscanf to extract the student's name and the waitlisted course serial number (S.no) from the line, and stores them in the provided StudentDetails struct.
- Reads waitlisted students data from waitlisted.csv and populates the students array using parse\_waitlisted\_student.
- Iterates through each waitlisted student and tries to allocate the waitlisted course. For each waitlisted student, it checks if the desired course has remaining seats.
  - If seats are available, it allocates the course to the student and decreases the seatcount.
  - Prints a message indicating the allocation.
- After processing all students, the program writes the updated course information back to remainingseats.csv, ensuring that the new seat counts are saved.

This systematic approach ensures that waitlisted students are allocated courses based on seat availability and updates the course data to reflect these changes.



## PRINTFINAL MODULE



### Explanation:

- The process starts by reading courses from "Courses.csv", parsing each line, and storing the details in an array, returning the number of courses read or -1 if the file can't be opened.
- Next, it reads student allocations from "finalallottment.csv", storing names and allocated courses in an array, also returning -1 if there's a file error. To print the final course allotment, it iterates over students, printing their names and allocated course details by matching them from the courses array in a tabular format.
- The main function for final allotment calls these functions to read data and print details, returning 0 on success or 1 if errors occur. A similar function reads courses again, storing them in another array and returning the count or -1 on failure. Another function finds a course by serial number by searching the array and returns the course details if found, or NULL if not.
- To print course details, given a serial number and details, the information is formatted and displayed. For printing student waitlist details, the process prompts for a student's name, reads from "waitlist.csv" and "successfulwaitlist.csv", and prints the student's waitlisted and successfully allocated courses, indicating if no details are found.
- The main function for the waitlist reads courses and prints waitlist details for a specific student using the previously read course data.

## 5. Implementation

### 5.1. Data Organization

To organise, structure and arrange data effectively the storage and classification of data was done using csv files and structures.

### 5.2 Data Storage

The Login details and choice filling details are stored in csv files. Files were also used as databases for storing preferences of candidates and also storing the allotments of counselling. In the code itself, Arrays of structures were used to store the candidate details. The rationale behind picking of Array of structures is that if the student count is quite high it would be more efficient to access them through an index than to define a structure variable for all of them individually.

The “**Login.csv**” file contains all the candidate Names, their username and password.

	A	B	C
1	John Doe	john123	password123
2	Alice Smith	alice_student	alicepassword
3	Sam Johnson	sam_student	sam_securepass
4	Emma Taylor	emma_22	passwordEmma
5	Michael Lee	mike34	mike_pass123
<	Lisa Garcia	lisa_student	lisapassword
7	David Martinez	david_student	david_secure123
8	Sophia Robinson	ryan321	ryan_pass987
9	James Clark	james_student	jamespass456
10	Emily Brown	emily44	emily_pass

**Fig: Login.csv**

The “**Students.csv**” file stores S.no, student name, student ID, the course they have completed, their CGPA and attendance percentage.

	A	B	C	D	E	F
1						
2	1	John Doe	S001	CICS101	8.5	85
3	2	Alice Smith	S002	CICS101	7.8	90
4	3	Sam Johnson	S003	CPF102	7.2	80
5	4	Emily Brown	S004	CICS101	8	92
6	5	Michael Lee	S005	CDS103	7.5	88
7	6	Lisa Garcia	S006	CICS101	6.9	75
8	7	David Martinez	S007	CPF102	8.2	87
9	8	Sophia Robinson	S008	CDBMS104	7.7	82
10	9	James Clark	S009	COS105	8.6	95
11	10	Emma Taylor	S010	COS105	7	78

**Fig: Students.csv**

The “**seats.csv**” contains the S.no, Course name, Course ID, Faculty ID and number of available seats.

	A	B	C	D	E
1	<a href="#">S.No</a>	Course Name	Course ID	Faculty ID	Seats Available
2	1	Introduction to CS	CICS101	ICSA	7
3	2	Introduction to CS	CICS101	ICSB	7
4	3	Introduction to CS	CICS101	ICSC	7
5	4	Prog. Fundamentals	CPF102	PFA	7
6	5	Prog. Fundamentals	CPF102	PFB	7
7	6	Prog. Fundamentals	CPF102	PFC	7
8	7	Data Structures	CDS103	DSA	7
9	8	Data Structures	CDS103	DSB	7
10	9	Data Structures	CDS103	DSC	7
11	10	Database Managemen	CDBMS104	DBMSA	7
12	11	Database Managemen	CDBMS104	DBMSB	7
13	12	Database Managemen	CDBMS104	DBMSC	7
14	13	Operating Systems	COS105	OSA	7
15	14	Operating Systems	COS105	OSB	7
16	15	Operating Systems	COS105	OSC	7
17	16	Computer Networks	CCN106	CNA	7
18	17	Computer Networks	CCN106	CNB	7
19	18	Computer Networks	CCN106	CNC	7
20	19	Web Development	EWD107	WDA	7
21	20	Web Development	EWD107	WDB	7
22	21	Machine Learning	EML108	MLA	7
23	22	Machine Learning	EML108	MLB	7
24	23	Artificial Intelligence	EAI109	AIA	7
25	24	Artificial Intelligence	EAI109	AIB	7
26	25	Cybersecurity	ECS110	CSA	7
27	26	Cybersecurity	ECS110	CSB	7

**Fig: seats.csv**

The “Courses.csv” file is used to store S.no, Course name, course ID, Faculty ID, number of credits for each course and the respective prerequisite for the course.

	A	B	C	D	E	F
1						
2	1	Introduction to CS	CICS101	ICSA	4	NA
3	2	Introduction to CS	CICS101	ICSB	4	NA
4	3	Introduction to CS	CICS101	ICSC	4	NA
5	4	Prog. Fundamentals	CPF102	PFA	4	CICS101
6	5	Prog. Fundamentals	CPF102	PFB	4	CICS101
7	6	Prog. Fundamentals	CPF102	PFC	4	CICS101
8	7	Data Structures	CDS103	DSA	4	CPF102
9	8	Data Structures	CDS103	DSB	4	CPF102
10	9	Data Structures	CDS103	DSC	4	CPF102
11	10	Database Managem	CDBMS104	DBMSA	4	CPF102
12	11	Database Managem	CDBMS104	DBMSB	4	CPF102
13	12	Database Managem	CDBMS104	DBMSC	4	CPF102
14	13	Operating Systems	COS105	OSA	4	CDS103
15	14	Operating Systems	COS105	OSB	4	CDS103
16	15	Operating Systems	COS105	OSC	4	CDS103
17	16	Computer Networks	CCN106	CNA	4	CDS103
18	17	Computer Networks	CCN106	CNB	4	CDS103
19	18	Computer Networks	CCN106	CNC	4	CDS103
20	19	Web Development	EWD107	WDA	2	CPF102
21	20	Web Development	EWD107	WDB	2	CPF102
22	21	Machine Learning	EML108	MLA	2	CDS103
23	22	Machine Learning	EML108	MLB	2	CDS103
24	23	Artificial Intelligence	EAI109	AIA	2	CDS103
25	24	Artificial Intelligence	EAI109	AIB	2	CDS103
26	25	Cybersecurity	ECS110	CSA	2	COS105
27	26	Cybersecurity	ECS110	CSB	2	COS105

**Fig: Courses.csv**

The “**choicelist.csv**” to store the student name and the choices inputted by the user.

	A	B	C	D	E
1	John Doe	11	15	16	20
2	Alice Smith	11	15	22	16
3	Sam Johnson	11	14	22	25
4	Emma Taylor	11	15	22	25
5	Michael Lee	11	15	19	22
6	Lisa Garcia	11	15	20	22
7	David Martinez	11	13	16	22
8	Sophia Robinson	11	13	16	22
9	James Clark	11	15	17	22
10	Emily Brown	11	13	18	22

**Fig: choicelist.csv**

The “**ranklist.csv**” file stores the rank, the S.no in which their registered, Student name, Student ID, prerequisite course, cgpa , their attendance and ranking score.

	A	B	C	D	E	F	G	H
1	Rank	<a href="#">S.No</a>	Name	Student ID	Prerequisites	CGPA	Attendance	Ranking Score
2	1	9	James Clark	S009	COS105	8.6	95	25.88
3	2	4	Emily Brown	S004	CICS101	8	92	24.8
4	3	2	Alice Smith	S002	CICS101	7.8	90	24.24
5	4	7	David Martinez	S007	CPF102	8.2	87	23.96
6	5	1	John Doe	S001	CICS101	8.5	85	23.8
7	6	5	Michael Lee	S005	CDS103	7.5	88	23.6
8	7	8	Sophia Robinson	S008	CDBMS104	7.7	82	22.56
9	8	3	Sam Johnson	S003	CPF102	7.2	80	21.76
10	9	10	Emma Taylor	S010	COS105	7	78	21.2
<	10	6	Lisa Garcia	S006	CICS101	6.9	75	20.52

**Fig: ranklist.csv**

The “**allotmentlist.csv**” is used to store the name of the student and courses for which they have been allotted.

	A	B	C	D	E	F
1	Name	Allocated Courses				
2	James Clark	11	15	17	22	8
3	Emily Brown	11	13	18	22	8
4	Alice Smith	11	15	22	16	8
5	David Martinez	11	13	16	22	8
6	John Doe	11	15	16	20	
7	Michael Lee	11	15	19	22	8
8	Sophia Robinson	11	13	16	22	8
9	Sam Johnson	14	22	25	8	
10	Emma Taylor	15	25			
<	Lisa Garcia	15	20			

**Fig: allotmentlist.csv**

The “**prefinalallottment.csv**” consists of the Student name along with the pre final list of allotted courses where changes can be made.

	A	B	C	D	E
1	John Doe	11	15	16	20
2	Alice Smith	11	15	22	16
3	Sam Johnson	14	22	25	
4	Emma Taylor	15	25		
5	Michael Lee	11	15	19	22
6	Lisa Garcia	15	20		
7	David Martinez	11	13	16	22
8	Sophia Robinson	11	13	16	22
9	James Clark	11	15	17	22
10	Emily Brown	11	15	18	22

**Fig: prefinalallottment.csv**

The “**waitlist.csv**” contains the student name along with the S.no of the course they have been waitlisted for.

	A	B	C
1	Name	Waitlisted Courses	
2	Sam Johnson		
3	Emma Taylor	8	
4	Lisa Garcia	22	8

**Fig: waitlist.csv**

The “**successfulwaitlist.csv**” stores the finalised list of all students name and their waitlisted courses.

	A	B	C
1	Name	Successful Allotted Waitlisted Courses	
2	Sam Johnson	11	
3	Emma Taylor	11	22
4	Lisa Garcia	11	

**Fig:successfulwaitlist.csv**

The “**finalallottment.csv**” file contains the student names and their allotted courses after the entire seat allotment process is over. Irrespective of whether the waitlisted courses are allotted or not and these are the finalised courses for each student. In case the waitlisted course has not been allotted, a course is allotted to the student by random.

	A	B	C	D	E	F
1	John Doe	11	15	16	20	
2	Alice Smith	15	22	16	8	
3	Sam Johnson	14	22	25	8	11
4	Emma Taylor	15	25	11	22	
5	Michael Lee	15	19	22	8	
6	Lisa Garcia	15	20	11	5	
7	David Martinez	11	13	16	8	
8	Sophia Robinson	13	16	22	8	
9	James Clark	11	15	17	22	8
10	Emily Brown	11	13	18	22	8

**Fig: finalallottment.csv**

The “**remaining\_seats.csv**” gets updated accordingly and stores all remaining seats available for each course after the final allotment is done.

	A	B	C	D	E
1	<a href="#">S.No</a>	Course Name	Course ID	Faculty ID	Seats Available
2	1	Introduction to CS	CICS101	ICSA	7
3	2	Introduction to CS	CICS101	ICSB	7
4	3	Introduction to CS	CICS101	ICSC	7
5	4	Prog. Fundamentals	CPF102	PFA	7
6	5	Prog. Fundamentals	CPF102	PFB	7
7	6	Prog. Fundamentals	CPF102	PFC	7
8	7	Data Structures	CDS103	DSA	7
9	8	Data Structures	CDS103	DSB	0
10	9	Data Structures	CDS103	DSC	7
11	10	Database Managemt	CDBMS104	DBMSA	7
12	11	Database Managemt	CDBMS104	DBMSB	0
13	12	Database Managemt	CDBMS104	DBMSC	7
14	13	Operating Systems	COS105	OSA	4
15	14	Operating Systems	COS105	OSB	6
16	15	Operating Systems	COS105	OSC	1
17	16	Computer Networks	CCN106	CNA	3
18	17	Computer Networks	CCN106	CNB	6
19	18	Computer Networks	CCN106	CNC	6
20	19	Web Development	EWD107	WDA	6
21	20	Web Development	EWD107	WDB	5
22	21	Machine Learning	EML108	MLA	7
23	22	Machine Learning	EML108	MLB	0
24	23	Artificial Intelligence	EAI109	AIA	7
25	24	Artificial Intelligence	EAI109	AIB	7
26	25	Cybersecurity	ECS110	CSA	5
27	26	Cybersecurity	ECS110	CSB	7

**Fig: remaining\_seats.csv**



## 6. Validation through Detailed Test Cases for various scenarios

When the student enters the page, a main menu to enter or exit is shown.

```
~~~~~  
WELCOME TO COURSE REGISTRATION PORTAL!  
~~~~~  
MAIN MENU:  
1. Enter  
2. Exit  
Enter your CHOICE: |
```

The candidate tries to log in using their Student ID and password.

- In case of incorrect password or username, the system shows “Login Failed!”
- If the correct details have been entered, login becomes successful.

```
~~~~~  
MAIN MENU:  
1. Enter  
2. Exit  
Enter your CHOICE: 1  
Enter Username: emily44  
Enter Password: emily_pas  
Login Failed!  
~~~~~
```

```
~~~~~  
MAIN MENU:  
1. Enter  
2. Exit  
Enter your CHOICE: 1  
Enter Username: emily44  
Enter Password: emily_pass  
~~~~~
```

- Once login is successful, all the available courses for choice filling are displayed.

```
Welcome Emily Brown!

1. Choice Filling
2. Check Seat Allotment Status
3. Exit

Enter your CHOICE: 1
Enter Username: emily44
Enter Password: emily_pass

Welcome Emily Brown!

AVAILABLE COURSES:
S.No  COURSE NAME          COURSE ID  FACULTY ID  CREDITS  PREREQUISITE
1      Introduction to CS      CICS101    ICSA        4         NA
2      Introduction to CS      CICS101    ICSB        4         NA
3      Introduction to CS      CICS101    ICSC        4         NA
4      Prog. Fundamentals      CPF102     PFA         4         CICS101
5      Prog. Fundamentals      CPF102     PFB         4         CICS101
6      Prog. Fundamentals      CPF102     PFC         4         CICS101
7      Data Structures         CDS103     DSA         4         CPF102
8      Data Structures         CDS103     DSB         4         CPF102
9      Data Structures         CDS103     DSC         4         CPF102
10     Database Management     CDBMS104   DBMSA       4         CPF102
11     Database Management     CDBMS104   DBMSB       4         CPF102
12     Database Management     CDBMS104   DBMSC       4         CPF102
13     Operating Systems       COS105     OSA         4         CDS103
14     Operating Systems       COS105     OSB         4         CDS103
15     Operating Systems       COS105     OSC         4         CDS103
16     Computer Networks       CCN106     CNA         4         CDS103
17     Computer Networks       CCN106     CNB         4         CDS103
18     Computer Networks       CCN106     CNC         4         CDS103
19     Web Development         EWD107     WDA         2         CPF102
```

- Next, the choice filling process starts. The student is asked to fill the courses they have already completed, number of courses and the S.no of the course they want to choose in the upcoming semester.

**Test case 1:** When the student does not satisfy or has not completed the prerequisites for the chosen course.

```
How many courses do you want to choose? (4 to 6): 4

Enter the S.No of COURSE 1: 5
You have not completed the prerequisites for this course.
```

**Test case 2:** When the number of courses is less than 4 or more than 6

```

5      Prog. Fundamentals      CPF102      PFB      4      CICS101
6      Prog. Fundamentals      CPF102      PFC      4      CICS101
7      Data Structures         CDS103      DSA      4      CPF102
8      Data Structures         CDS103      DSB      4      CPF102
9      Data Structures         CDS103      DSC      4      CPF102
10     Database Management     CDBMS104    DBMSA    4      CPF102
11     Database Management     CDBMS104    DBMSB    4      CPF102
12     Database Management     CDBMS104    DBMSC    4      CPF102
13     Operating Systems       COS105      OSA      4      CDS103
14     Operating Systems       COS105      OSB      4      CDS103
15     Operating Systems       COS105      OSC      4      CDS103
16     Computer Networks       CCN106      CNA      4      CDS103
17     Computer Networks       CCN106      CNB      4      CDS103
18     Computer Networks       CCN106      CNC      4      CDS103
19     Web Development         EWD107      WDA      2      CPF102
20     Web Development         EWD107      WDB      2      CPF102
21     Machine Learning        EML108      MLA      2      CDS103
22     Machine Learning        EML108      MLB      2      CDS103
23     Artificial Intelligence  EAI109      AIA      2      CDS103
24     Artificial Intelligence  EAI109      AIB      2      CDS103
25     Cybersecurity           ECS110      CSA      2      COS105
26     Cybersecurity           ECS110      CSB      2      COS105

Enter the course IDs of all the courses you have completed: CDS103,CPF102,COS105

How many courses do you want to choose? (4 to 6): 2
Invalid number of courses. Please enter a number between 4 and 6.

How many courses do you want to choose? (4 to 6): 8
Invalid number of courses. Please enter a number between 4 and 6.

```

**Test case 3:** When the user picks the same course as already picked.

```

Enter the S.No of COURSE 1: 11
Continue or remove this course (C/R)? C

Enter the S.No of COURSE 2: 11
Course with S.No 11 is already selected.

```

**Test case 4:** When the sum of credits does not fall within the limit of 10-20 credits

```

How many courses do you want to choose? (4 to 6): 4

Enter the S.No of COURSE 1: 19
Continue or remove this course (C/R)? C

Enter the S.No of COURSE 2: 21
Continue or remove this course (C/R)? C

Enter the S.No of COURSE 3: 23
Continue or remove this course (C/R)? C

Enter the S.No of COURSE 4: 25
Continue or remove this course (C/R)? C

SELECTED COURSES:
S.No  COURSE NAME      COURSE ID      FACULTY ID      CREDITS  PREREQUISITE
19     Web Development     EWD107         WDA             2         CPF102
21     Machine Learning     EML108         MLA             2         CDS103
23     Artificial Intelligence EAI109         AIA             2         CDS103
25     Cybersecurity         ECS110         CSA             2         COS105
Total credits: 8

Do you want to confirm your choices? (Y/N): Y
Error: Total credits must be between 10 and 20. Please choose again.

Do you want to fill choices again? (Y/N):

```

- After valid choices have been filled, the program prompts the user to confirm their choice.

```

Do you want to fill choices again? (Y/N): Y

Enter the course IDs of all the courses you have completed: CDS103,CPF102,COS105

How many courses do you want to choose? (4 to 6): 4

Enter the S.No of COURSE 1: 11
Continue or remove this course (C/R)? C

Enter the S.No of COURSE 2: 13
Continue or remove this course (C/R)? R
Course removed.

Enter the S.No of COURSE 2: 15
Continue or remove this course (C/R)? C

Enter the S.No of COURSE 3: 18
Continue or remove this course (C/R)? C

Enter the S.No of COURSE 4: 22
Continue or remove this course (C/R)? C

SELECTED COURSES:
S.No  COURSE NAME          COURSE ID  FACULTY ID  CREDITS  PREREQUISITE
11    Database Management    CDBMS104   DBMSB       4         CPF102
15    Operating Systems      COS105     OSC         4         CDS103
18    Computer Networks      CCN106     CNC         4         CDS103
22    Machine Learning       EML108     MLB         2         CDS103
Total credits: 18

Do you want to confirm your choices? (Y/N):

```

- After this, the user is given the option to check their allotment status (after the number of entries in choicelist.csv=10)

```

SELECTED COURSES:
S.No  COURSE NAME          COURSE ID  FACULTY ID  CREDITS  PREREQUISITE
11    Database Management    CDBMS104   DBMSB       4         CPF102
15    Operating Systems      COS105     OSC         4         CDS103
18    Computer Networks      CCN106     CNC         4         CDS103
22    Machine Learning       EML108     MLB         2         CDS103
Total credits: 18

Do you want to confirm your choices? (Y/N): Y

Do you want to fill choices again? (Y/N): N

1. Choice Filling
2. Check Seat Allotment Status
3. Exit

Enter your CHOICE: 3
Logging out...

=====

MAIN MENU:
1. Enter
2. Exit

Enter your CHOICE:

```

Once the choice filling has been done, the next step, seat allotment takes place. The csv files are updated and the allotment statuses of each student can be checked using the **Check allotment status** function.

**Test case 1:** When the user gets all the courses they chose.

```
1. Choice Filling
2. Check Seat Allotment Status
3. Exit

Enter your CHOICE: 2
Enter student name: John Doe

Processing Student:
Student: John Doe

ALLOTTED COURSES:
11      Database Management      CDBMS104      DBMSB
15      Operating Systems        COS105        OSC
16      Computer Networks        CCN106        CNA
20      Web Development          EWD107        WDB
Do you want to accept the course Database Management (Y/N)? Y
Do you want to accept the course Operating Systems (Y/N)? Y
Do you want to accept the course Computer Networks (Y/N)? Y
Do you want to accept the course Web Development (Y/N)? Y

FINAL ALLOTTED COURSES:
11      Database Management      CDBMS104      DBMSB
15      Operating Systems        COS105        OSC
16      Computer Networks        CCN106        CNA
20      Web Development          EWD107        WDB

Congratulations! Courses have been successfully chosen!
```

**Test case 2:** When the user gets waitlisted for a course they chose.

```
1. Choice Filling
2. Check Seat Allotment Status
3. Exit

Enter your CHOICE: 2
Enter student name: Lisa Garcia

Processing Student:
Student: Lisa Garcia

ALLOTTED COURSES:
15      Operating Systems        COS105        OSC
20      Web Development          EWD107        WDB

Waitlisted courses:
11      Database Management      CDBMS104      DBMSB
22      Machine Learning         EML108        MLB
8       Data Structures          CDS103        DSB

Do you want to accept the course Operating Systems (Y/N)? Y
Do you want to accept the course Web Development (Y/N)? Y

FINAL ALLOTTED COURSES:
15      Operating Systems        COS105        OSC
20      Web Development          EWD107        WDB

Waitlisted courses:
11      Database Management      CDBMS104      DBMSB
22      Machine Learning         EML108        MLB
8       Data Structures          CDS103        DSB

Congratulations! Courses have been successfully chosen!
```

- After allotment, the student is allowed to accept or withdraw the course that have been offered to them.

**Test case 1:** When the user accepts all the courses.

```

1. Choice Filling
2. Check Seat Allotment Status
3. Exit

Enter your CHOICE: 2
Enter student name: John Doe

Processing Student:
Student: John Doe

ALLOTTED COURSES:
11      Database Management      CDBMS104      DBMSB
15      Operating Systems        COS105        OSC
16      Computer Networks        CCN106        CNA
20      Web Development          EWD107        WDB
Do you want to accept the course Database Management (Y/N)? Y
Do you want to accept the course Operating Systems (Y/N)? Y
Do you want to accept the course Computer Networks (Y/N)? Y
Do you want to accept the course Web Development (Y/N)? Y

FINAL ALLOTTED COURSES:
11      Database Management      CDBMS104      DBMSB
15      Operating Systems        COS105        OSC
16      Computer Networks        CCN106        CNA
20      Web Development          EWD107        WDB

Congratulations! Courses have been successfully chosen!

```

**Test case 2:** When the user withdraws more courses than the minimum amount.

```

Welcome John Doe!

1. Choice Filling
2. Check Seat Allotment Status
3. Exit

Enter your CHOICE: 2
Enter student name: John Doe

Processing Student:
Student: John Doe

ALLOTTED COURSES:
11      Database Management      CDBMS104      DBMSB
15      Operating Systems        COS105        OSC
16      Computer Networks        CCN106        CNA
20      Web Development          EWD107        WDB
Do you want to accept the course Database Management (Y/N)? Y
Do you want to accept the course Operating Systems (Y/N)? Y
Do you want to accept the course Computer Networks (Y/N)? Y
Do you want to accept the course Web Development (Y/N)? N
Error: You can only decline up to 0 course(s) [Minimum 4 courses].
Do you want to accept the course Web Development (Y/N)? Y

```

- The final allotted courses per student is displayed.

```
FINAL ALLOTTED COURSES:
11 Database Management CDBMS104 DBMSB
15 Operating Systems COS105 OSC
16 Computer Networks CCN106 CNA
20 Web Development EWD107 WDB

Congratulations! Courses have been successfully chosen!
```

- Once any waitlisted student slides in and gets the vacant seat, and he enters the portal and clicks 2.Check seat allotment status. The allocated waitlistes courses and unallocated courses is displayed

```
Welcome Lisa Garcia!

1. Choice Filling
2. Check Seat Allotment Status
3. Exit

Enter your CHOICE: 2
Enter student name: Lisa Garcia

UNALLOCATED COURSES:
8 Data Structures CDS103 DSB

ALLOCATED WAITLISTED COURSES:
11 Database Management CDBMS104 DBMSB
22 Machine Learning EML108 MLB
```

- Once we exit the program, the final allotment list of all the students is printed

FINAL COURSE ALLOTTMENT LIST			
Student: John Doe			
S.no	Course Name	Course ID	Faculty ID
11	Database Management	CDBMS104	DBMSB
15	Operating Systems	COS105	OSC
16	Computer Networks	CCN106	CNA
20	Web Development	EWD107	WDB
Student: Alice Smith			
S.no	Course Name	Course ID	Faculty ID
15	Operating Systems	COS105	OSC
22	Machine Learning	EML108	MLB
16	Computer Networks	CCN106	CNA
8	Data Structures	CDS103	DSB

Student: Sam Johnson			
S.no	Course Name	Course ID	Faculty ID
14	Operating Systems	COS105	OSB
22	Machine Learning	EML108	MLB
25	Cybersecurity	ECS110	CSA
8	Data Structures	CDS103	DSB
11	Database Management	CDBMS104	DBMSB
Student: Emma Taylor			
S.no	Course Name	Course ID	Faculty ID
15	Operating Systems	COS105	OSC
25	Cybersecurity	ECS110	CSA
11	Database Management	CDBMS104	DBMSB
4	Prog. Fundamentals	CPF102	PFA

Student: Michael Lee			
S.no	Course Name	Course ID	Faculty ID
15	Operating Systems	COS105	OSC
19	Web Development	EWD107	WDA
22	Machine Learning	EML108	MLB
8	Data Structures	CDS103	DSB
Student: Lisa Garcia			
S.no	Course Name	Course ID	Faculty ID
15	Operating Systems	COS105	OSC
20	Web Development	EWD107	WDB
11	Database Management	CDBMS104	DBMSB
22	Machine Learning	EML108	MLB



Student: David Martinez

S.no	Course Name	Course ID	Faculty ID
13	Operating Systems	COS105	OSA
16	Computer Networks	CCN106	CNA
22	Machine Learning	EML108	MLB
8	Data Structures	CDS103	DSB

Student: Sophia Robinson

S.no	Course Name	Course ID	Faculty ID
11	Database Management	CDBMS104	DBMSB
13	Operating Systems	COS105	OSA
16	Computer Networks	CCN106	CNA
8	Data Structures	CDS103	DSB

Student: James Clark

S.no	Course Name	Course ID	Faculty ID
11	Database Management	CDBMS104	DBMSB
15	Operating Systems	COS105	OSC
17	Computer Networks	CCN106	CNB
8	Data Structures	CDS103	DSB

Student: Emily Brown

S.no	Course Name	Course ID	Faculty ID
11	Database Management	CDBMS104	DBMSB
13	Operating Systems	COS105	OSA
17	Computer Networks	CCN106	CNB
8	Data Structures	CDS103	DSB

## 7. Scopes and Limitations

### Scopes:

**Online Course Registration:** Develop a system enabling students to conveniently and flexibly register for courses online.

**Automated Prerequisite Checking:** Design a system to automatically verify if students meet the prerequisites for courses, reducing errors and saving time.

**Real-Time Course Availability:** Implement a system that offers up-to-date information on course availability, allowing students to view which courses are full and which have openings.

**Waitlist Functionality:** Incorporate a feature enabling students to join waitlists for full courses and receive notifications if spots become available.

**Personalized Recommendation:** Create a system that delivers personalized course recommendations to students based on their academic history, utilizing machine learning algorithms.

### Limitations:

**Restricted Course Availability:** A primary constraint of course registration systems is the limited availability of courses. Once a course reaches full capacity, students may face waitlists or need to select alternative options.

**Absence of Personal Interaction:** Course registration platforms often lack the personal touch of in-person registration, making it challenging for students to receive immediate answers to their inquiries or guidance regarding course selection.

**Limited Options for Interdisciplinary Courses:** In instances where there are few interdisciplinary or cross-listed course offerings, students may have limited opportunities to explore diverse academic fields.

## **8. Concerns with respect to society, legal and ethical perspectives**

The project should address societal, legal, and ethical considerations to ensure responsible and fair use of data and resources.

### **Societal**

- Additionally, the system should promote equal access and opportunity for all students, while avoiding biases or discrimination based on factors such as gender, race, or socioeconomic status.

### **Legal**

- Candidate details are verified before proceeding to the choice filling to ensure credibility of the user.
- The entire process is made transparent and merit-based to ensure fairness and provide equal opportunity.
- It also includes safeguarding student privacy, adhering to data protection laws, and maintaining transparency in decision-making processes.

### **Ethical**

- The counseling process ensures transparency at all stages, which holds the responsible parties accountable in case discrepancies arise.
- Candidate's personal information is kept confidential to maintain data privacy and digital security.

## **9. Learning Outcomes**

- The project provides the opportunity to practice and improve coding skills in C programming language by working with conditionals, arrays, strings, structures and files.
- The breaking down of the problem statement into smaller modules of simpler complexity, helps to manage, organise, implement and maintain the code in an efficient manner.
- By implementing the University Course Registration program, we learn about the design and implementation involved in implementing different concepts of C for modules such as obtaining student choices, checking seat availability, allotting seats and waitlisting them.
- The project involves the use of reading and writing data with the help of files, which helps improve file handling skills.
- During the implementation phase, if and when bugs/errors are encountered, troubleshooting and debugging provide an opportunity to explore the program further and resolve the issues and validate the solution provided.
- During the course of the project, the importance of teamwork and collaborative skills in designing, implementing, testing and debugging the code

## 10. References

1. An improvisation method where we can recommend courses for students using collaborative filtering Algorithm and can be used only for newly registered students.

<https://www.bing.com/ck/a?!&&p=018abe3824d16ddcJmltdHM9MTexODg0MTYwMCZpZ3VpZD0xMWYzYzc5MC0xNzdILTY1NWQtM2UzOS1kNDBjMTZkMzY0NTQmaW5zaWQ9NTIyOA&pptn=3&ver=2&hsh=3&fclid=11f3c790-177e-655d-3e39-d40c16d36454&psq=doris+personalised+recomendation&u=a1aHR0cHM6Ly93d3cubmNiaS5ubG0ubmloLmdvdi9wbWMvYXJ0aWNsZXNmUE1DMTAyMzc0OTYv&ntb=1>

2. Reference the below code to implement our csv files.

[jkh394/Course-Registration-System: A course registration system created using Java object oriented programming. \(github.com\)](#)