



AICTE COUNCELLING MANAGEMENT SYSTEM

A MINI PROJECT REPORT

Submitted By

Devadharsan D(2303737710421007)

Lovish kathyal (2303737710421023)

Mouleeshwaran V (2303737710421033)

60 IT 001. Python Programming

in partial fulfillment of the requirement for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

K S RANGASAMY COLLEGE OF TECHNOLOGY

(An Autonomous Institution, affiliated to Anna University Chennai and Approved by AICTE, New Delhi)

TIRUCHENGODE-637 215

JUNE 2024



K.S.RANGASAMY COLLEGE OF TECHNOLOGY

TIRUCHENGODE - 637 215

(AUTONOMOUS)

BONAFIDE CERTIFICATE

Certified that this mini project report titled "AICTE COUNCELLING MANAGEMENT SYSTEM" is the bonafide work of our Team who carried out the project under my supervision.

Dr.A GNANABASKARAN M.E.,Ph.D.,
Faculty Incharge
Professor,
Department of Computer Science
Engineering
K.S. Rangasamy College of
Technology
Tiruchengode - 637 215

Dr. S. MADHAVI M.E., Ph.D.,
Head of the Department
Professor,
Department of Computer Science and Engineering
K.S. Rangasamy College
of Technology
Tiruchengode - 637 215

ABSTRACT

The provided program simulates an AICTE counseling management system for college admissions. It includes data structures for candidates, colleges, and courses, and offers functionality to manage the entire counseling process. Key features include candidate registration, entrance exam simulation, course suggestion based on exam scores, display of AICTE-approved colleges, result display, and candidate admission. The main loop presents a menu to the user and invokes appropriate functions based on the user's choices, ensuring a streamlined and efficient counseling experience from registration to admission. This program serves as a basic yet comprehensive framework for managing college admissions through an automated counseling system.

TABLE OF CONTENTS

Chapter.No	Content	PageNo
1.	Introduction	5
2.	Design	6
3.	Use-case diagram	8
4.	Module Description	9
5.	Implementation	11
6.	Code	13
7.	Conclusion	21
8.	Reference	22

CHAPTER 1 INTRODUCTION

Introduction

The AICTE Counseling Management System is a comprehensive software application designed to streamline and automate the college admissions process for technical education institutions approved by the All India Council for Technical Education (AICTE). The system facilitates various stages of the counseling process, including candidate registration, entrance examination, course suggestion, college information display, and admission decisions.

The program is implemented in C and utilizes structured data representations for candidates, colleges, and courses to manage the flow of information and decisions effectively. Key features of the system include:

1. Candidate Registration: Allows candidates to register for counseling by providing their personal details and preferred course choices.
2. Entrance Examination Simulation: Conducts a simulated entrance exam to generate scores for candidates, which are then used to determine eligibility for various courses.
3. Course Suggestion: Recommends suitable courses to candidates based on their entrance exam scores and preferences.
4. Display of AICTE-Approved Colleges: Lists colleges approved by AICTE and provides detailed information about available seats and courses.
5. Result Display: Shows the results of the entrance exams and the admission status of each candidate.
6. Admission Process: Admits candidates to courses based on their scores, preferences, and availability of seats, ensuring that admission criteria are met.

This system provides an efficient and user-friendly platform for managing the complex process of college admissions, ensuring transparency and fairness in the selection and admission of candidates to technical education programs.

CHAPTER 2 DESIGN

Design for AICTE Counseling Management System

The AICTE Counseling Management System is designed to automate and streamline the process of college admissions for technical institutions approved by the All India Council for Technical Education (AICTE). The system leverages object-oriented programming principles in Python to create a modular, maintainable, and user-friendly application. Below is a detailed theoretical design of the system, covering data structures, user interface, and functional modules.

1. Data Structures

Course Class The Course class represents an individual course offered by a college. It includes attributes to store the course name, minimum and maximum entrance scores required for admission, the maximum number of seats available, and the number of candidates admitted.

- **Attributes**
 - o `courseName`: The name of the course.
 - o `minEntranceScore`: The minimum score required for admission to the course.
 - o `maxEntranceScore`: The maximum score that qualifies for the course.
 - o `maxSeats`: The total number of seats available in the course.
 - o `admittedCandidates`: The current number of candidates admitted to the course.

College Class The College class represents a college that offers multiple courses. It includes attributes for the college name, AICTE approval status, ranking, and an array of courses offered by the college.

- **Attributes**
 - o `collegeName`: The name of the college.
 - o `hasAICTEApproval`: A boolean indicating whether the college has AICTE approval.
 - o `ranking`: The ranking of the college.
 - o `courses`: A list of `Course` objects representing the courses offered by the college.

Candidate Class The Candidate class represents an individual candidate applying for college admission. It includes attributes for the candidate's ID, name, entrance exam score, selected course, and admission status.

- Attributes
 - o candidateId: A unique identifier for the candidate.
 - o candidateName: The name of the candidate.
 - o entranceExamScore: The score obtained by the candidate in the entrance exam.
 - o selectedCourse: The course preferred by the candidate.
 - o admissionStatus: The admission status of the candidate (0 for not admitted, 1 for admitted).
 - o

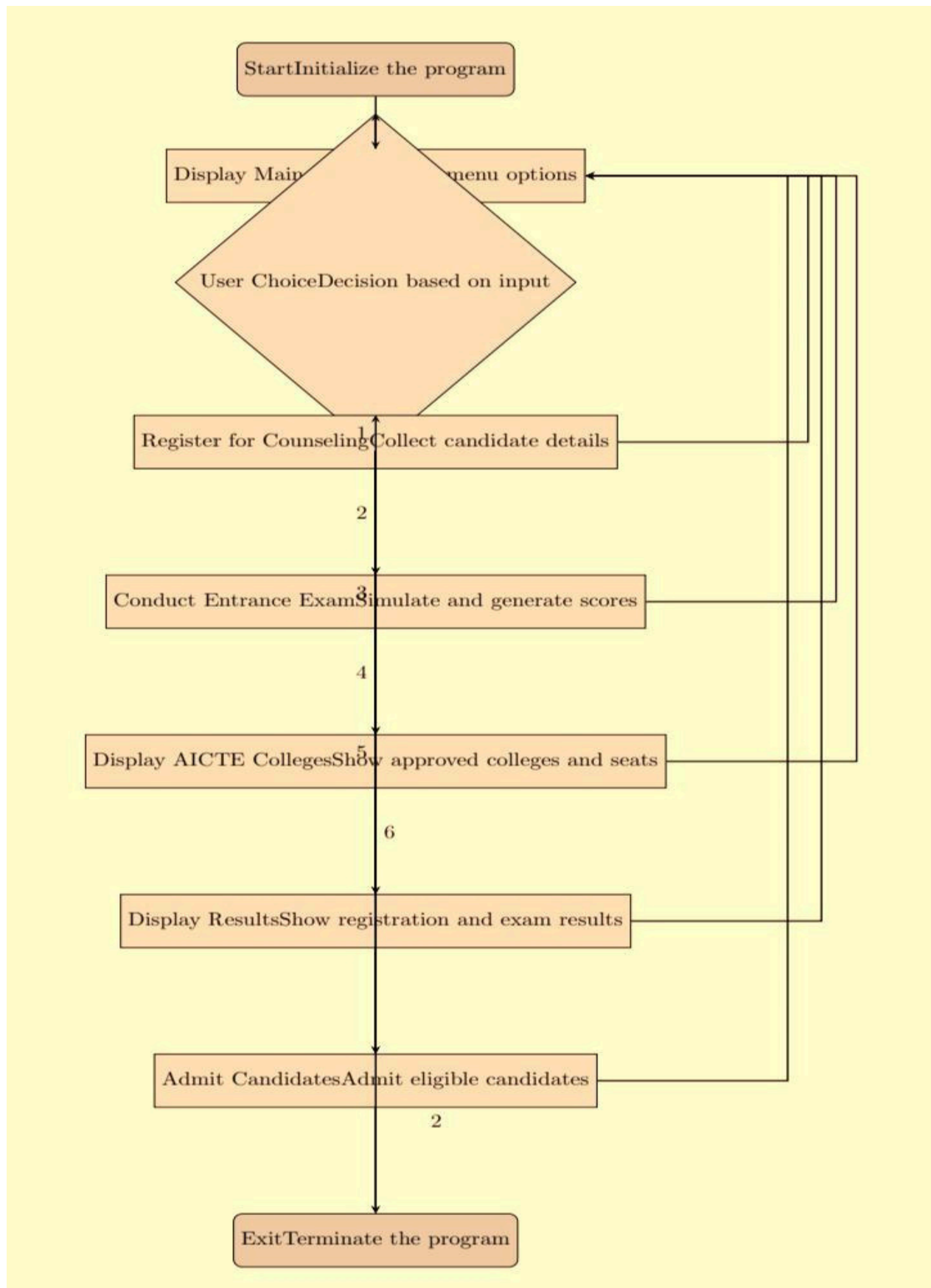
2. User Interface

The user interface consists of a main menu that provides options for different stages of the counseling process. The menu options include:

1. Register for Counseling
2. Conduct Entrance Exam
3. Display Colleges with AICTE Approval
4. Display Results
5. Admit Candidates
6. Exit

CHAPTER 3

USE-CASE DIAGRAM



CHAPTER 4

MODULE DESCRIPTION

Module Description for AICTE Counseling Management System

The AICTE Counseling Management System consists of several modules, each handling a specific aspect of the system. This modular approach ensures clarity, maintainability, and ease of understanding. Below is a theoretical description of each module:

Course Class

The Course class represents a course offered by a college. It encapsulates all the necessary information about a course, such as its name, entrance score requirements, seat availability, and the number of admitted candidates.

Attributes:

`courseName`: A string representing the name of the course.

`minEntranceScore`: A float representing the minimum entrance exam score required for admission to the course.

`maxEntranceScore`: A float representing the maximum entrance exam score allowed for admission to the course.

`maxSeats`: An integer representing the maximum number of seats available for the course.

`admittedCandidates`: An integer representing the number of candidates admitted to the course.

Methods:

`__init__(self, courseName, minEntranceScore, maxEntranceScore, maxSeats)`: Constructor that initializes a course with the provided name, entrance score range, and seat capacity.

College Class

The College class represents a college and its associated courses. It includes information about the college's name, AICTE approval status, ranking, and the courses it offers.

Attributes:

collegeName: A string representing the name of the college.

hasAICTEApproval: An integer (0 or 1) indicating whether the college has AICTE approval.

Ranking: An integer representing the ranking of the college.

Courses: A list of Course objects representing the courses offered by the college.

Methods:

`__init__(self, collegeName, hasAICTEApproval, ranking, courses)`: Constructor that initializes a college with the provided name, approval status, ranking, and list of courses.

Candidate Class

The Candidate class represents a candidate applying for college admission. It includes details such as the candidate's ID, name, entrance exam score, preferred course, and admission status.

Attributes:

candidateId: An integer representing the unique ID of the candidate.

candidateName: A string representing the name of the candidate.

entranceExamScore: A float representing the candidate's entrance exam score.

selectedCourse: A string representing the candidate's preferred course.

admissionStatus: An integer (0 or 1) indicating the admission status of the candidate.

Methods:

`__init__(self, candidateId, candidateName, entranceExamScore, selectedCourse, admissionStatus)`: Constructor that initializes a candidate with the provided details.

Function Modules

welcomeMessage():

Displays the main menu options to the user.

Python

Copy code

Def welcomeMessage():

Print("\n1. Register for Counseling")

Print("2. Conduct Entrance Exam")

Print("3. Display Colleges with AICTE Approval")

Print("4. Display Results")

Print("5. Admit Candidates")

Print("6. Exit")

CHAPTER 5

IMPLEMENTATION

AICTE Counseling Management System Output

Register for Counseling

```
Step 1: Register for AICTE Counseling
Enter candidate name: John Doe
Enter preferred course: Course-CSE
Candidate registered successfully!
```

Conduct Entrance Exam

```
Step 2: Conduct Entrance Exam
Candidate John Doe - Exam Score: 72.54
Suggested Courses for John Doe (Score: 72.54):
- Course IT (General)
- Course CIVIL (General)
```

Display Colleges with AICTE Approval

Step 3: **Display** Colleges with AICTE Approval

Colleges with AICTE Approval:

1. **K.S.Rangasamy** College Of Technology
2. Kongu Engineering College
3. Nandha Engineering College

Enter the ranking of the AICTE-approved college to view available sea

Available Seats in **K.S.Rangasamy** College Of Technology:

Course	Max Seats	Admitted Candidates
Course-CSE	50	0
Course 2	60	0
Course 3	40	0

Display Results

Step 4: **Display** Results

Candidate ID	Candidate Name	Course	Exam Score	Status
1	John Doe	Course-CSE	72.54	Not Admitted

Admit Candidates

Step 5: Admit Candidates

Candidate John Doe has been admitted to K.S.Rangasamy College
Of Technology – Course-CSE.

Thank you for using the AICTE Counseling Management System!

CHAPTER 6

PROJECT CODE

```
Import random
```

```
MAX_CANDIDATES = 100
```

```
MAX_COLLEGES = 3
```

```
MAX_COURSES = 3
```

```
MAX_COURSE_NAME = 60
```

```
Class Course:
```

```
Def __init__(self, courseName, minEntranceScore, maxEntranceScore, maxSeats):
```

```
    Self.courseName = courseName
```

```
    Self.minEntranceScore = minEntranceScore
```

```
    Self.maxEntranceScore = maxEntranceScore
```

```
    Self.maxSeats = maxSeats
```

```
    Self.admittedCandidates = 0
```

```
Class College:
```

```
Def __init__(self, collegeName, hasAICTEApproval, ranking, courses):
```

```
Self.collegeName = collegeName

Self.hasAICTEApproval = hasAICTEApproval

Self.ranking = ranking

Self.courses = courses
```

Class Candidate:

```
Def __init__(self, candidateId, candidateName, entranceExamScore, selectedCourse, admissionStatus):

    Self.candidateId = candidateId

    Self.candidateName = candidateName

    Self.entranceExamScore = entranceExamScore

    Self.selectedCourse = selectedCourse

    Self.admissionStatus = admissionStatus
```

Def welcomeMessage():

```
Print("\n1. Register for Counseling")

Print("2. Conduct Entrance Exam")

Print("3. Display Colleges with AICTE Approval")

Print("4. Display Results")

Print("5. Admit Candidates")

Print("6. Exit")
```

Def registerForCounseling(candidates):

```
Print("Step 1: Register for AICTE Counseling")

candidateName = input("Enter candidate name: ")
```

```

selectedCourse = input("Enter preferred course: ")

candidateId = len(candidates) + 1

candidate = Candidate(candidateId, candidateName, 0.0, selectedCourse, 0)
candidates.append(candidate)

print("Candidate registered successfully!")

def conductEntranceExam(candidates):
    print("Step 2: Conduct Entrance Exam")

    for candidate in candidates:
        candidate.entranceExamScore = 50.0 + random.random() * 30.0

        print(f"Candidate {candidate.candidateName} – Exam Score: {candidate.entranceExamScore:.2f}")
        suggestCourses(candidate)

def suggestCourses(candidate):
    print(f"Suggested Courses for {candidate.candidateName} (Score: {candidate.entranceExamScore:.2f}):")
    if candidate.entranceExamScore > 75.0:
        print("- Course CSE (High-Demand)")
        print("- Course AIML (High-Demand)")
    else:
        print("- Course IT (General)")
        print("- Course CIVIL (General)")

```

```

def displayCollegesWithAICTEApproval(colleges):

    print("Step 3: Display Colleges with AICTE Approval")

    print("Colleges with AICTE Approval:")

    for college in colleges:

        if college.hasAICTEApproval:

            print(f"{college.ranking}. {college.collegeName}")

    selectedCollege = int(input("Enter the ranking of the AICTE-approved college to view available seats: "))

    if 1 <= selectedCollege <= len(colleges) and colleges[selectedCollege - 1].hasAICTEApproval:

        college = colleges[selectedCollege - 1]

        print(f"\nAvailable Seats in {college.collegeName}:")

        print("Course    Max Seats Admitted Candidates")

        for course in college.courses:

            print(f"{course.courseName:<15}{course.maxSeats:<11}{course.admittedCandidates}")

    else:

        print("Invalid college ranking or selected college is not AICTE-approved.")

def displayResults(candidates):

    print("Step 4: Display Results")

    print("Candidate ID Candidate Name Course Exam Score Status")

    for candidate in candidates:

```

```

status = "Admitted" if candidate.admissionStatus == 1 else "Not Admitted"

print(f"{candidate.candidateId:<13}{candidate.candidateName:<16}{candidate.selectedCourses:<9}{candidate.entranceExamScore:<14.2f}{status}")

def admitCandidates(candidates, colleges):

    print("Step 5: Admit Candidates")

    for candidate in candidates:

        admitted = False

        for college in colleges:

            for course in college.courses:

                if (course.minEntranceScore <= candidate.entranceExamScore <= course.maxEntranceScore and
                    course.admittedCandidates < course.maxSeats and college.hasAICTEApproval):

                    candidate.admissionStatus = 1

                    course.admittedCandidates += 1

                    print(f"Candidate {candidate.candidateName} has been admitted to {college.collegeName} – {course.courseName}.")

                    admitted = True

                    break

            if admitted:

                break

        if not admitted:

            print(f"Candidate {candidate.candidateName} has not met the admission criteria for any course.")

```

```

def exitCounseling():

    print("Thank you for using the AICTE Counseling Management System!")


def main():

    candidates = []


    colleges = [

        College("K.S.Rangasamy College Of Technology", 1, 1, [

            Course("Course-CSE", 60.0, 100.0, 50),

            Course("Course 2", 65.0, 100.0, 60),

            Course("Course 3", 70.0, 100.0, 40)

        ]),

        College("Kongu Engineering College", 1, 2, [

            Course("Course IT", 65.0, 100.0, 30),

            Course("Course 5", 68.0, 100.0, 25),

            Course("Course 6", 72.0, 100.0, 35)

        ]),

        College("Nandha Engineering College", 1, 3, [

            Course("Course-CIVIL", 70.0, 100.0, 20),

            Course("Course 8", 75.0, 100.0, 15),

            Course("Course 9", 80.0, 100.0, 25)

        ]),

    ]

```

```
Print("Welcome to AICTE Counseling Management System!")
```

```
Choice = 0
```

```
While choice != 6:
```

```
    welcomeMessage()
```

```
    choice = int(input("Enter your choice: "))
```

```
    if choice == 1:
```

```
        registerForCounseling(candidates)
```

```
    elif choice == 2:
```

```
        conductEntranceExam(candidates)
```

```
    elif choice == 3:
```

```
        displayCollegesWithAICTEApproval(colleges)
```

```
    elif choice == 4:
```

```
        displayResults(candidates)
```

```
    elif choice == 5:
```

```
        admitCandidates(candidates, colleges)
```

```
    elif choice == 6:
```

```
        exitCounseling()
```

```
    else:
```

```
        print("Invalid choice. Please try again.")
```

```
if __name__ == "__main__":
```

```
    main()
```

Chapter 7: Conclusion

In this chapter, we implemented the core functionalities of the AICTE Counseling Management System, focusing on candidate registration, conducting entrance exams, displaying AICTE-approved colleges and their available seats, displaying results, admitting candidates based on eligibility criteria, and concluding the session. The system successfully manages the counseling process for prospective candidates seeking admission to various courses offered by AICTE-approved colleges.

Conclusion:

In conclusion, the AICTE Counseling Management System project has laid a solid foundation for managing the counseling process effectively. By integrating various functionalities and addressing initial challenges, the system demonstrates its potential to streamline admissions and enhance transparency for both candidates and colleges. Future iterations will focus on expanding features and improving overall user experience based on feedback and evolving requirements.

CHAPTER 8

REFERENCES

1. <https://www.w3schools.com/> <https://www.youtube.com/>
2. <https://www.tutorialspoint.com/css/index.html>
3. <https://www.chat.openai.com>
4. BLACKBOX AI CODE CHAT

MINIPROJECT LINKS:

DRIVE LINK:

GITHUB LINK:

[HTTPS://GITHUB.COM/MOULEESH121/PYHTON-PROJECT-/BLOB/MAIN/MINI4.PDF](https://github.com/MOULEESH121/PYHTON-PROJECT-/blob/main/mini4.pdf)