SRINIVASA EDUCATIONAL SOCIETY'S
PACE INSTITUTE OF TECHNOLOGY & SCIENCES
(AUTONOMOUS)
Approved by AICTE, UGC, New Delhi & Govt. of Andhra Pradesh | Permanently Affiliated to JNTUK, Kakinada, A.P.
ACCREDITED BY NAAC WITH 'A' GRADE | ACCREDITED BY NBA
An ISO 9001 : 2008 Certified Institution |'A' Grade Engineering College by Government of A.P.
NH-16, Near Valluramma Temple, ONGOLE - 523 272, A.P., Contact No.: 08592 278315, 9581456310 | www.pace.ac.in

DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING
(2023-2024)

Internship and mini project based on python programming with Data Engineer

Project title: Course navigator pro

In accordance with requirement of degree of
BACHELOR OF TECHNOLOGY
In
ELECTRICAL AND ELECTRONICS ENGINEERING

Submitted by:
B.Akash
21KQ1A0233
Under the Mentorship
M.sravan kumar

Date: 11-06-2024

Project title

COURSE NAVIGATOR PRO

Abstract: This project is about brief explanation and display about courses in a college . a college is having different types of courses and students are needed to enroll the courses and to know details the following process is useful.

1.Description: A Course Navigator Probe is typically a software tool or system.This program is a course management system. It allows users to input course details such as name, fee, number of enrolled students, total number of seats, and duration.

It calculates and displays the total number of courses in the system. It lists courses with a fee less than $1000 and provides the total count of such courses.

Requirements

2.1 Functional Requirements

- **Print Specific Course Details** :Users should have the option to input a course name, and the system should display specific details of that course, such as fee, enrolled students, seats left, and duration.
- **Query Total Filled Seats** :The system should calculate and display the total number of filled seats across all courses
- **Query Course Fee :**Users should be able to input a course name, and the system should display the fee for that specific course .

2.2 Non-functional Requirements

- **Usability:** The system should have an intuitive user interface for easy interaction.
- **Performance:** The system should handle a large number of courses efficiently.
- **Security:** User authentication and authorization mechanisms should be implemented to ensure data security.

Approach:

1. Understanding the Objective:

-> Define the purpose of the Course Navigator Probe, which is to assist users in exploring available courses based on their preferences and requirements.

2. Requirement Analysis:

-> Identify the target audience and their needs, such as students, professionals, or career counselors.

- > Determine the features and functionalities required for effective course navigation, such as search filters, recommendations, and course comparison.

3. Design:

-> Design the user interface to be intuitive and user-friendly, allowing users to easily navigate through available courses.

-> Determine the data sources for course information, including course databases, educational institutions' websites, and online learning platforms.

-> Plan the architecture of the navigator probe, considering factors such as scalability, performance, and data retrieval mechanisms.

4. Data Acquisition and Integration:

-> Develop mechanisms to collect and integrate course data from various sources into a centralized repository.

-> Implement data preprocessing techniques to clean, standardize, and organize the course information for efficient navigation.

5. Feature Implementation:

-> Develop search functionalities to allow users to find courses based on criteria such as subject, level, duration, and location.

-> Implement filtering options to refine search results according to user preferences, such as price range, accreditation, and course format.

-> Incorporate recommendation algorithms to suggest relevant courses based on user profiles, browsing history, and similarity to preferred courses.

-> Create course comparison tools to enable users to compare key attributes of multiple courses side by side, facilitating informed decision-making.

6. User Interaction:

-> Design interactive features to engage users and provide personalized experiences, such as user profiles, saved searches, and notifications.

-> Implement feedback mechanisms to gather user input and improve the navigator probe's effectiveness over time.

7. Testing:

-> Conduct extensive testing to ensure the navigator probe's functionality, usability, and performance.

-> Perform usability testing with target users to gather feedback and identify areas for improvement.

-> Validate the accuracy of search results, recommendations, and course comparisons against known benchmarks and user expectations.

8. Deployment:

-> Prepare the Course Navigator Probe for deployment on the desired platform, such as a web application, mobile app, or standalone software.

-> Ensure compatibility with different devices and operating systems to maximize accessibility for users.

-> Provide documentation and support resources to guide users in using the navigator probe effectively.

9. Evaluation and Iteration:

-> Monitor user interactions and feedback to evaluate the navigator probe's effectiveness in meeting users' needs and goals.

- >Analyze usage metrics, such as search patterns, click-through rates, and user satisfaction scores, to identify areas for improvement.

-> Iterate on the navigator probe based on evaluation results, incorporating new features, enhancing existing functionalities, and addressing user feedback to continuously enhance the user experience.

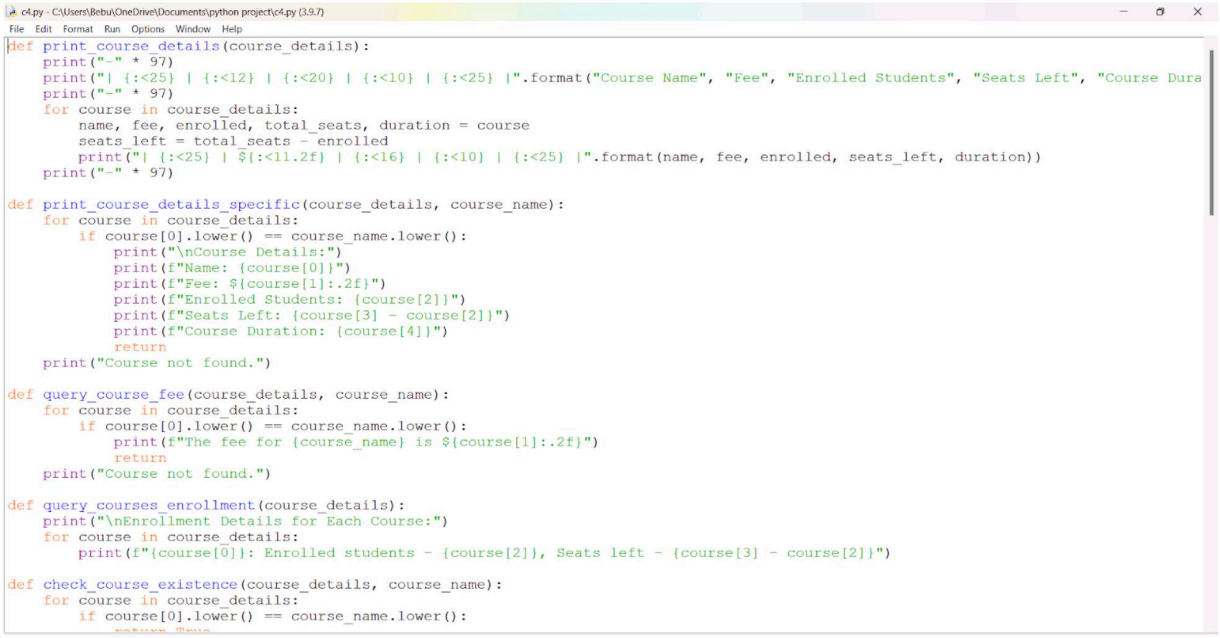
10. Maintenance and Updates:

-> Maintain the Course Navigator Probe by monitoring data sources, updating course information, and addressing technical issues as they arise.

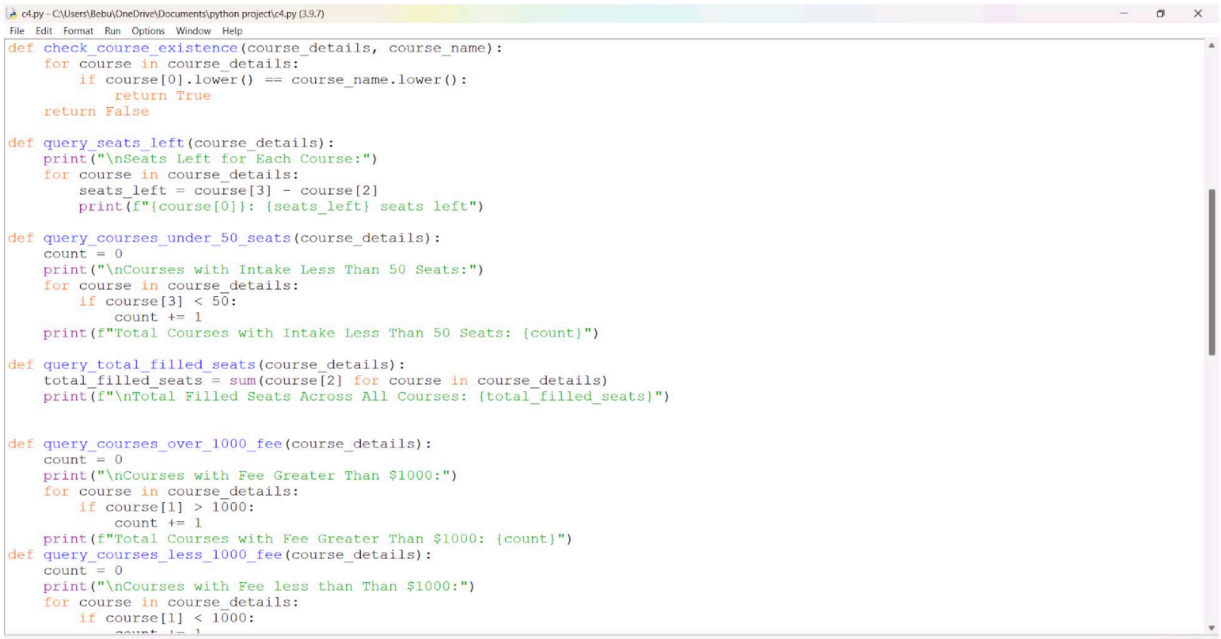
-> Regularly update the navigator probe with new features, course offerings, and improvements to ensure its relevance and usefulness to users.

-> Stay informed about emerging trends and technologies in education and course navigation to incorporate cutting-edge features and capabilities into the navigator probe.

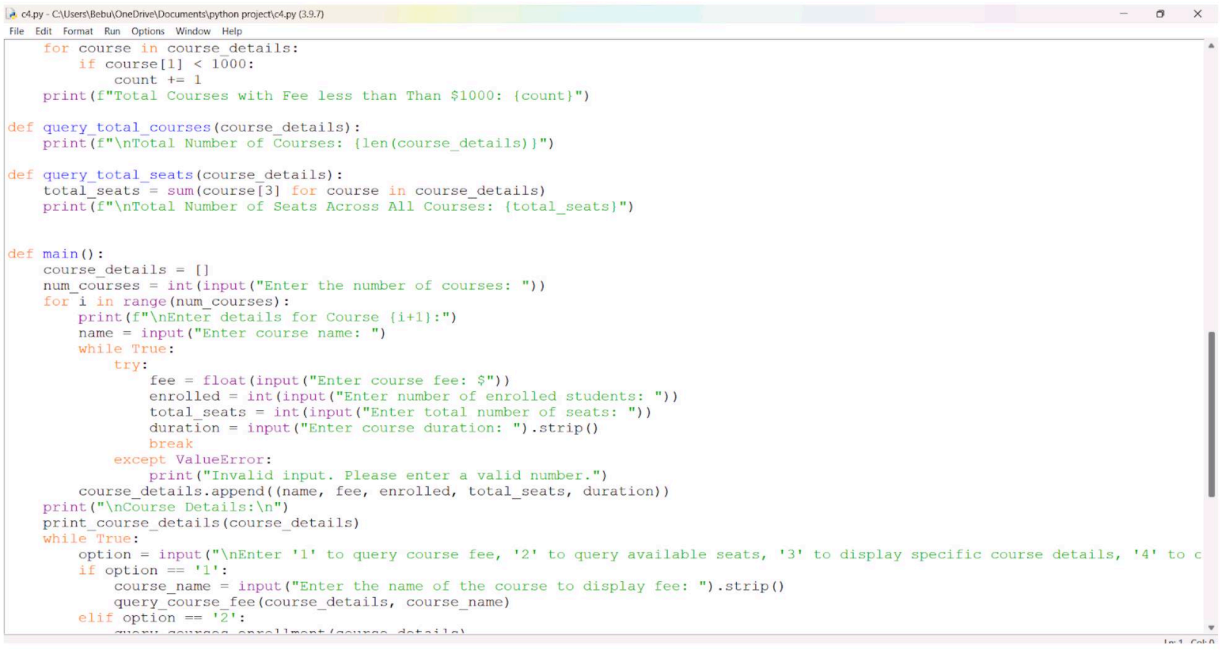
PROGRAM:



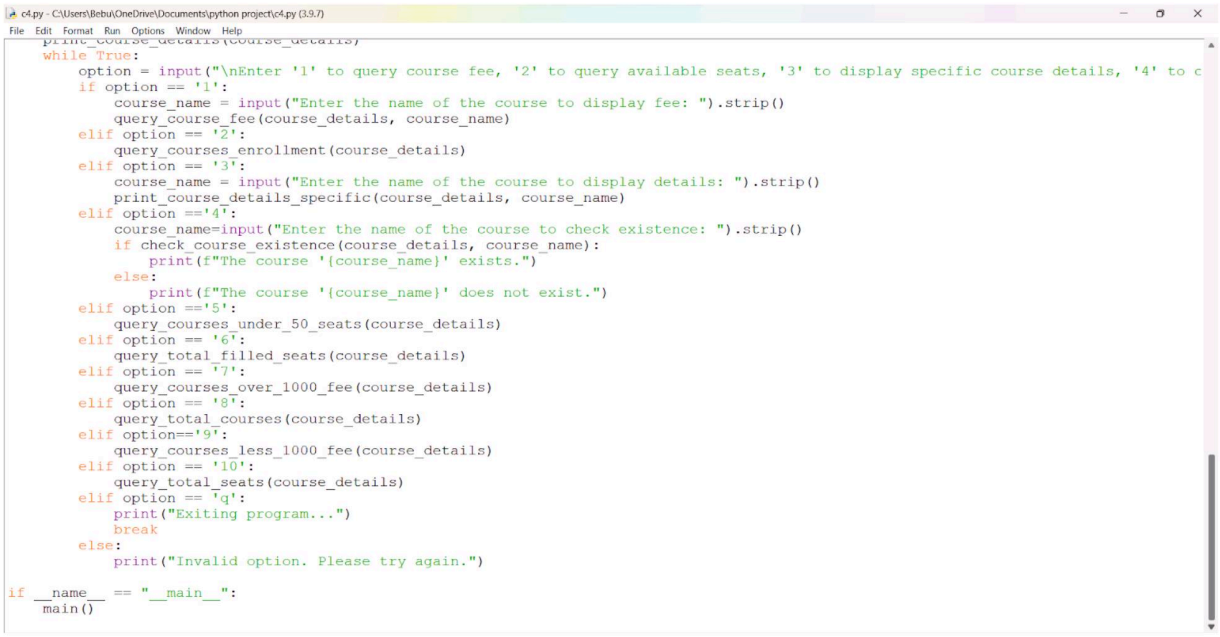
```
c4.py - C:\Users\Bebu\OneDrive\Documents\python project\c4.py (3.9.7)
File  Edit  Format  Run  Options  Window  Help
def print_course_details(course_details):
    print("-" * 97)
    print("| {:<25} | {:<12} | {:<20} | {:<10} | {:<25} |".format("Course Name", "Fee", "Enrolled Students", "Seats Left", "Course Dura
    print("-" * 97)
    for course in course_details:
        name, fee, enrolled, total_seats, duration = course
        seats_left = total_seats - enrolled
        print("| {:<25} | ${:<11.2f} | {:<16} | {:<10} | {:<25} |".format(name, fee, enrolled, seats_left, duration))
    print("-" * 97)

def print_course_details_specific(course_details, course_name):
    for course in course_details:
        if course[0].lower() == course_name.lower():
            print("\nCourse Details:")
            print(f"Name: {course[0]}")
            print(f"Fee: ${course[1]:.2f}")
            print(f"Enrolled Students: {course[2]}")
            print(f"Seats Left: {course[3] - course[2]}")
            print(f"Course Duration: {course[4]}")
            return
    print("Course not found.")

def query_course_fee(course_details, course_name):
    for course in course_details:
        if course[0].lower() == course_name.lower():
            print(f"The fee for {course_name} is ${course[1]:.2f}")
            return
    print("Course not found.")

def query_courses_enrollment(course_details):
    print("\nEnrollment Details for Each Course:")
    for course in course_details:
        print(f"{course[0]}: Enrolled students - {course[2]}, Seats left - {course[3] - course[2]}")

def check_course_existence(course_details, course_name):
    for course in course_details:
        if course[0].lower() == course_name.lower():
            return True
```



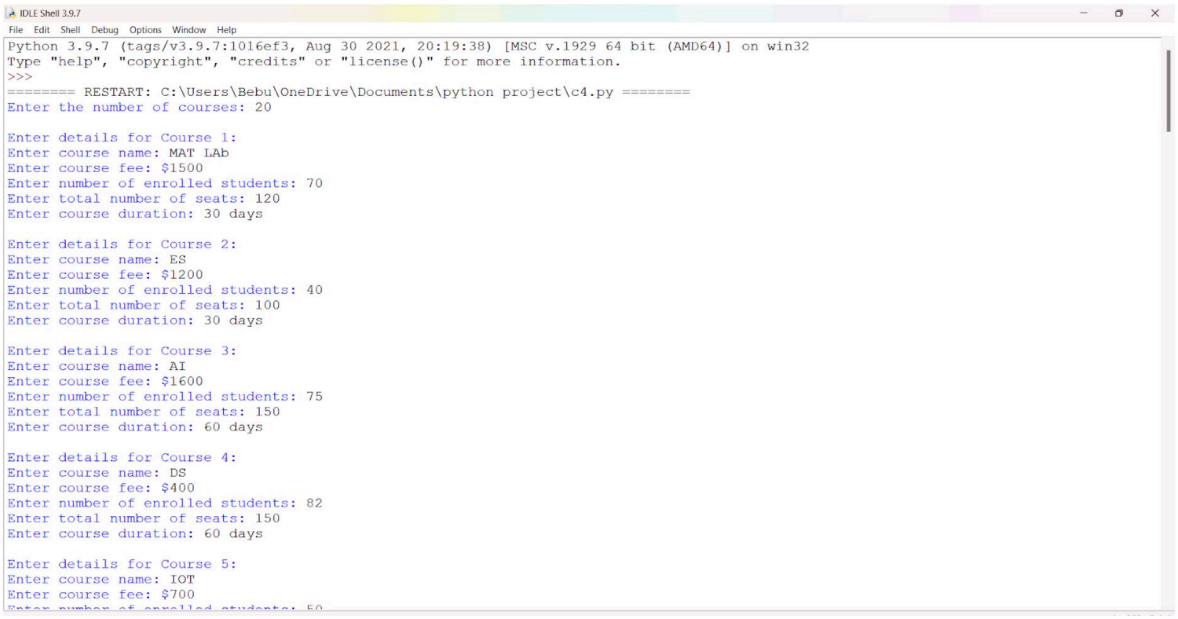
```
c4.py - C:\Users\Bebu\OneDrive\Documents\python project\c4.py (3.9.7)
File  Edit  Format  Run  Options  Window  Help
def check_course_existence(course_details, course_name):
    for course in course_details:
        if course[0].lower() == course_name.lower():
            return True
    return False

def query_seats_left(course_details):
    print("\nSeats Left for Each Course:")
    for course in course_details:
        seats_left = course[3] - course[2]
        print(f"{course[0]}: {seats_left} seats left")

def query_courses_under_50_seats(course_details):
    count = 0
    print("\nCourses with Intake Less Than 50 Seats:")
    for course in course_details:
        if course[3] < 50:
            count += 1
    print(f"Total Courses with Intake Less Than 50 Seats: {count}")

def query_total_filled_seats(course_details):
    total_filled_seats = sum(course[2] for course in course_details)
    print(f"\nTotal Filled Seats Across All Courses: {total_filled_seats}")


def query_courses_over_1000_fee(course_details):
    count = 0
    print("\nCourses with Fee Greater Than $1000:")
    for course in course_details:
        if course[1] > 1000:
            count += 1
    print(f"Total Courses with Fee Greater Than $1000: {count}")
def query_courses_less_1000_fee(course_details):
    count = 0
    print("\nCourses with Fee less than Than $1000:")
    for course in course_details:
        if course[1] < 1000:
```

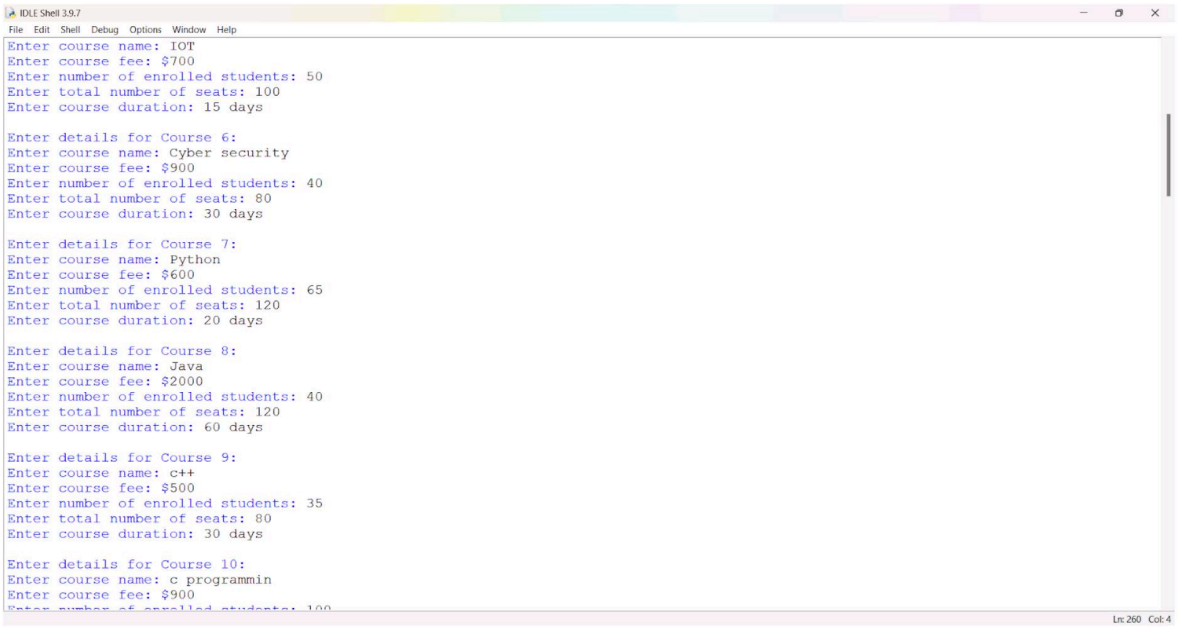


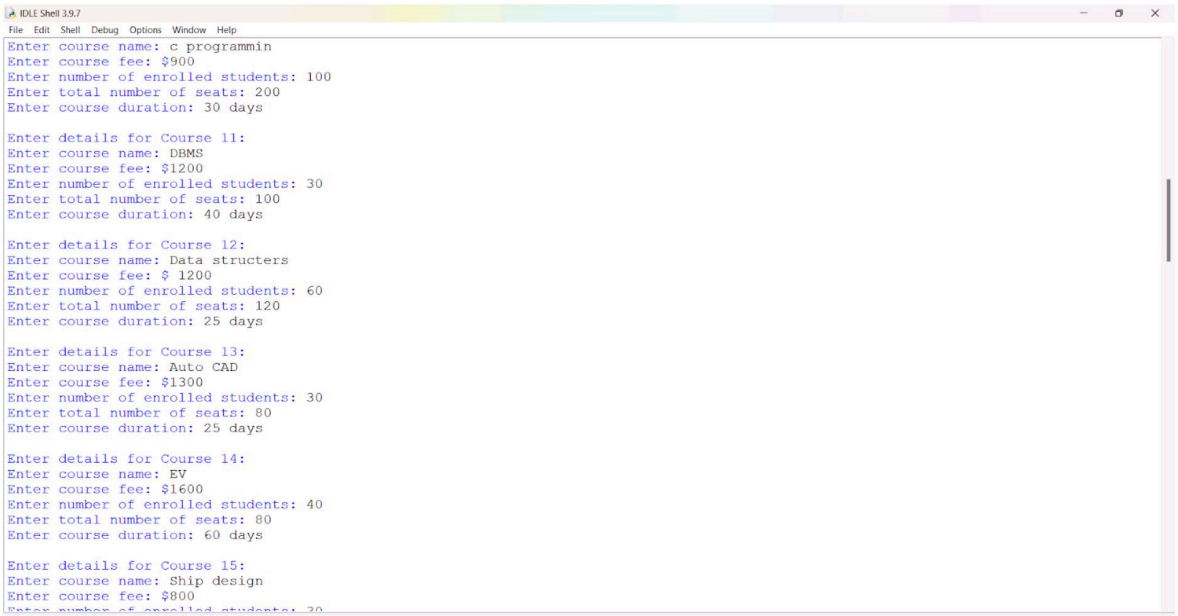
```
c4.py - C:\Users\Bebu\OneDrive\Documents\python project\c4.py (3.9.7)
File  Edit  Format  Run  Options  Window  Help
    for course in course_details:
        if course[1] < 1000:
            count += 1
    print(f"Total Courses with Fee less than Than $1000: {count}")

def query_total_courses(course_details):
    print(f"\nTotal Number of Courses: {len(course_details)}")

def query_total_seats(course_details):
    total_seats = sum(course[3] for course in course_details)
    print(f"\nTotal Number of Seats Across All Courses: {total_seats}")


def main():
    course_details = []
    num_courses = int(input("Enter the number of courses: "))
    for i in range(num_courses):
        print(f"\nEnter details for Course {i+1}:")
        name = input("Enter course name: ")
        while True:
            try:
                fee = float(input("Enter course fee: $"))
                enrolled = int(input("Enter number of enrolled students: "))
                total_seats = int(input("Enter total number of seats: "))
                duration = input("Enter course duration: ").strip()
                break
            except ValueError:
                print("Invalid input. Please enter a valid number.")
        course_details.append((name, fee, enrolled, total_seats, duration))
    print("\nCourse Details:\n")
    print_course_details(course_details)
    while True:
        option = input("\nEnter '1' to query course fee, '2' to query available seats, '3' to display specific course details, '4' to c
        if option == '1':
            course_name = input("Enter the name of the course to display fee: ").strip()
            query_course_fee(course_details, course_name)
        elif option == '2':
```

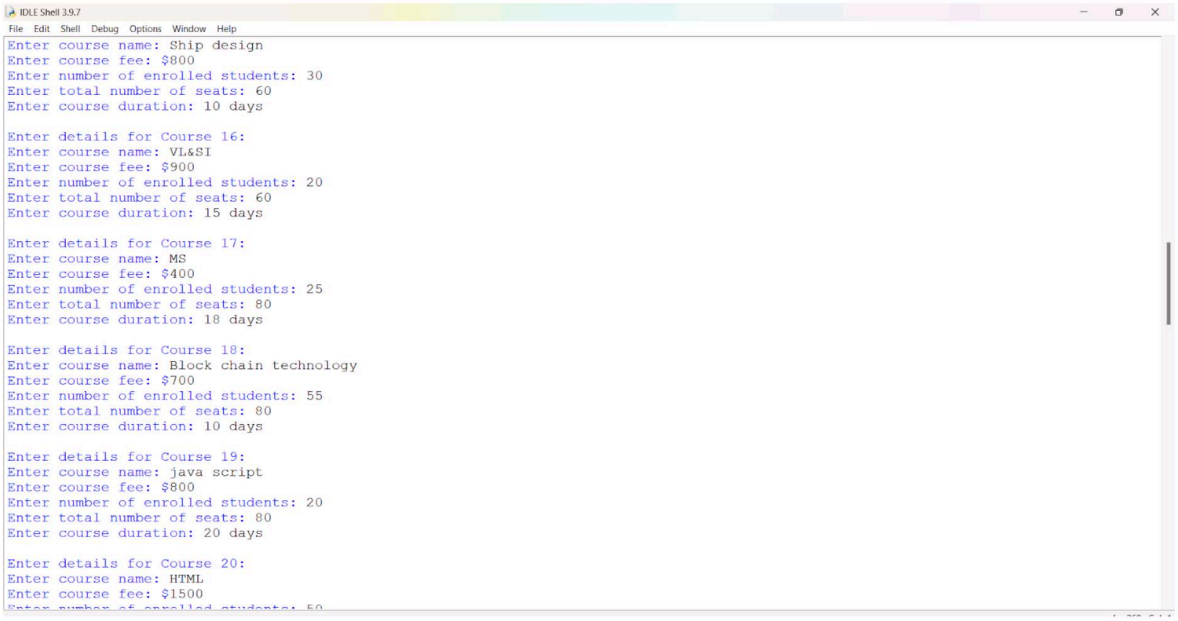

c4.py - C:\Users\Bebu\OneDrive\Documents\python project\c4.py (3.9.7)
File Edit Format Run Options Window Help
print_course_details(course_details)
while True:
option = input("\nEnter '1' to query course fee, '2' to query available seats, '3' to display specific course details, '4' to c
if option == '1':
course_name = input("Enter the name of the course to display fee: ").strip()
query_course_fee(course_details, course_name)
elif option == '2':
query_courses_enrollment(course_details)
elif option == '3':
course_name = input("Enter the name of the course to display details: ").strip()
print_course_details_specific(course_details, course_name)
elif option =='4':
course_name=input("Enter the name of the course to check existence: ").strip()
if check_course_existence(course_details, course_name):
print(f"The course '{course_name}' exists.")
else:
print(f"The course '{course_name}' does not exist.")
elif option =='5':
query_courses_under_50_seats(course_details)
elif option == '6':
query_total_filled_seats(course_details)
elif option == '7':
query_courses_over_1000_fee(course_details)
elif option == '8':
query_total_courses(course_details)
elif option=='9':
query_courses_less_1000_fee(course_details)
elif option == '10':
query_total_seats(course_details)
elif option == 'q':
print("Exiting program...")
break
else:
print("Invalid option. Please try again.")
if __name__ == "__main__":
main()

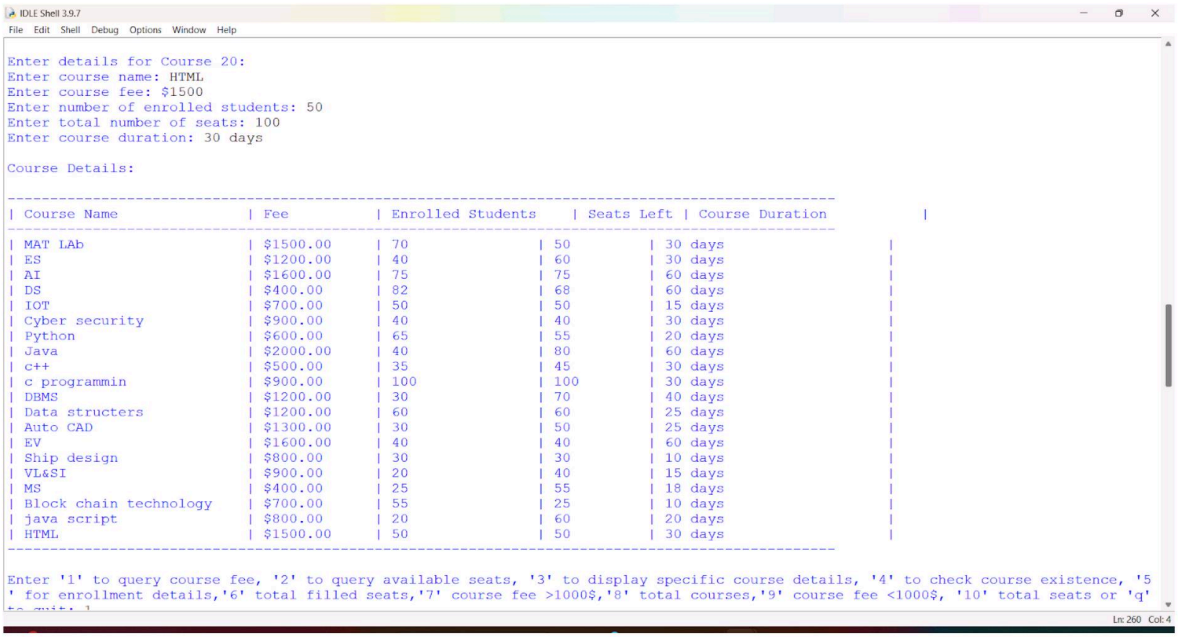
OUTPUT:

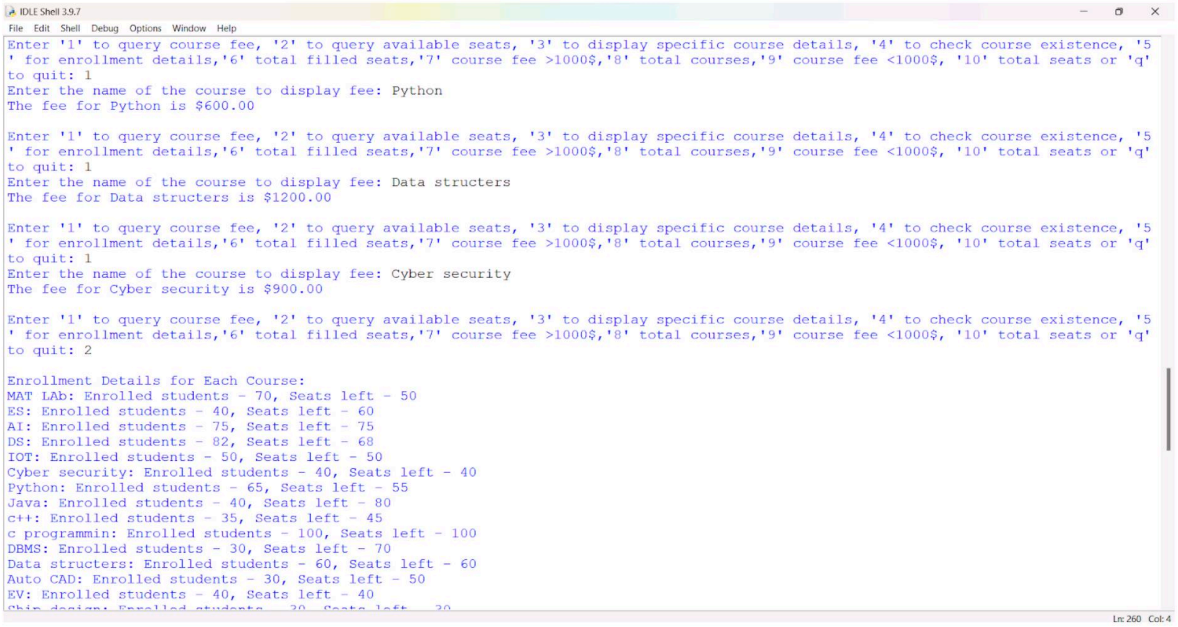

IDLE Shell 3.9.7
File Edit Shell Debug Options Window Help
Python 3.9.7 (tags/v3.9.7:1016ef3, Aug 30 2021, 20:19:38) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
======== RESTART: C:\Users\Bebu\OneDrive\Documents\python project\c4.py ========
Enter the number of courses: 20
Enter details for Course 1:
Enter course name: MAT LAb
Enter course fee: $1500
Enter number of enrolled students: 70
Enter total number of seats: 120
Enter course duration: 30 days
Enter details for Course 2:
Enter course name: ES
Enter course fee: $1200
Enter number of enrolled students: 40
Enter total number of seats: 100
Enter course duration: 30 days
Enter details for Course 3:
Enter course name: AI
Enter course fee: $1600
Enter number of enrolled students: 75
Enter total number of seats: 150
Enter course duration: 60 days
Enter details for Course 4:
Enter course name: DS
Enter course fee: $400
Enter number of enrolled students: 82
Enter total number of seats: 150
Enter course duration: 60 days
Enter details for Course 5:
Enter course name: IOT
Enter course fee: $700


IDLE Shell 3.9.7
File Edit Shell Debug Options Window Help
Enter course name: IOT
Enter course fee: $700
Enter number of enrolled students: 50
Enter total number of seats: 100
Enter course duration: 15 days
Enter details for Course 6:
Enter course name: Cyber security
Enter course fee: $900
Enter number of enrolled students: 40
Enter total number of seats: 80
Enter course duration: 30 days
Enter details for Course 7:
Enter course name: Python
Enter course fee: $600
Enter number of enrolled students: 65
Enter total number of seats: 120
Enter course duration: 20 days
Enter details for Course 8:
Enter course name: Java
Enter course fee: $2000
Enter number of enrolled students: 40
Enter total number of seats: 120
Enter course duration: 60 days
Enter details for Course 9:
Enter course name: c++
Enter course fee: $500
Enter number of enrolled students: 35
Enter total number of seats: 80
Enter course duration: 30 days
Enter details for Course 10:
Enter course name: c programmin
Enter course fee: $900
Ln: 260 Col: 4


IDLE Shell 3.9.7
File Edit Shell Debug Options Window Help
Enter course name: c programmin
Enter course fee: $900
Enter number of enrolled students: 100
Enter total number of seats: 200
Enter course duration: 30 days
Enter details for Course 11:
Enter course name: DBMS
Enter course fee: $1200
Enter number of enrolled students: 30
Enter total number of seats: 100
Enter course duration: 40 days
Enter details for Course 12:
Enter course name: Data structers
Enter course fee: $ 1200
Enter number of enrolled students: 60
Enter total number of seats: 120
Enter course duration: 25 days
Enter details for Course 13:
Enter course name: Auto CAD
Enter course fee: $1300
Enter number of enrolled students: 30
Enter total number of seats: 80
Enter course duration: 25 days
Enter details for Course 14:
Enter course name: EV
Enter course fee: $1600
Enter number of enrolled students: 40
Enter total number of seats: 80
Enter course duration: 60 days
Enter details for Course 15:
Enter course name: Ship design
Enter course fee: $800


IDLE Shell 3.9.7
File Edit Shell Debug Options Window Help
Enter course name: Ship design
Enter course fee: $800
Enter number of enrolled students: 30
Enter total number of seats: 60
Enter course duration: 10 days
Enter details for Course 16:
Enter course name: VL&SI
Enter course fee: $900
Enter number of enrolled students: 20
Enter total number of seats: 60
Enter course duration: 15 days
Enter details for Course 17:
Enter course name: MS
Enter course fee: $400
Enter number of enrolled students: 25
Enter total number of seats: 80
Enter course duration: 18 days
Enter details for Course 18:
Enter course name: Block chain technology
Enter course fee: $700
Enter number of enrolled students: 55
Enter total number of seats: 80
Enter course duration: 10 days
Enter details for Course 19:
Enter course name: java script
Enter course fee: $800
Enter number of enrolled students: 20
Enter total number of seats: 80
Enter course duration: 20 days
Enter details for Course 20:
Enter course name: HTML
Enter course fee: $1500


IDLE Shell 3.9.7
File Edit Shell Debug Options Window Help
Enter details for Course 20:
Enter course name: HTML
Enter course fee: $1500
Enter number of enrolled students: 50
Enter total number of seats: 100
Enter course duration: 30 days
Course Details:
Course Name	Fee	Enrolled Students	Seats Left	Course Duration
MAT LAb	$1500.00	70	50	30 days
ES	$1200.00	40	60	30 days
AI	$1600.00	75	75	60 days
DS	$400.00	82	68	60 days
IOT	$700.00	50	50	15 days
Cyber security	$900.00	40	40	30 days
Python	$600.00	65	55	20 days
Java	$2000.00	40	80	60 days
c++	$500.00	35	45	30 days
c programmin	$900.00	100	100	30 days
DBMS	$1200.00	30	70	40 days
Data structers	$1200.00	60	60	25 days
Auto CAD	$1300.00	30	50	25 days
EV	$1600.00	40	40	60 days
Ship design	$800.00	30	30	10 days
VL&SI	$900.00	20	40	15 days
MS	$400.00	25	55	18 days
Block chain technology	$700.00	55	25	10 days
java script	$800.00	20	60	20 days
HTML	$1500.00	50	50	30 days
Enter '1' to query course fee, '2' to query available seats, '3' to display specific course details, '4' to check course existence, '5' for enrollment details,'6' total filled seats,'7' course fee >1000$,'8' total courses,'9' course fee <1000$, '10' total seats or 'q' to quit: 1
Ln: 260 Col: 4


IDLE Shell 3.9.7
File Edit Shell Debug Options Window Help
Enter '1' to query course fee, '2' to query available seats, '3' to display specific course details, '4' to check course existence, '5' for enrollment details,'6' total filled seats,'7' course fee >1000$,'8' total courses,'9' course fee <1000$, '10' total seats or 'q' to quit: 1
Enter the name of the course to display fee: Python
The fee for Python is $600.00
Enter '1' to query course fee, '2' to query available seats, '3' to display specific course details, '4' to check course existence, '5' for enrollment details,'6' total filled seats,'7' course fee >1000$,'8' total courses,'9' course fee <1000$, '10' total seats or 'q' to quit: 1
Enter the name of the course to display fee: Data structers
The fee for Data structers is $1200.00
Enter '1' to query course fee, '2' to query available seats, '3' to display specific course details, '4' to check course existence, '5' for enrollment details,'6' total filled seats,'7' course fee >1000$,'8' total courses,'9' course fee <1000$, '10' total seats or 'q' to quit: 1
Enter the name of the course to display fee: Cyber security
The fee for Cyber security is $900.00
Enter '1' to query course fee, '2' to query available seats, '3' to display specific course details, '4' to check course existence, '5' for enrollment details,'6' total filled seats,'7' course fee >1000$,'8' total courses,'9' course fee <1000$, '10' total seats or 'q' to quit: 2
Enrollment Details for Each Course:
MAT LAb: Enrolled students - 70, Seats left - 50
ES: Enrolled students - 40, Seats left - 60
AI: Enrolled students - 75, Seats left - 75
DS: Enrolled students - 82, Seats left - 68
IOT: Enrolled students - 50, Seats left - 50
Cyber security: Enrolled students - 40, Seats left - 40
Python: Enrolled students - 65, Seats left - 55
Java: Enrolled students - 40, Seats left - 80
c++: Enrolled students - 35, Seats left - 45
c programmin: Enrolled students - 100, Seats left - 100
DBMS: Enrolled students - 30, Seats left - 70
Data structers: Enrolled students - 60, Seats left - 60
Auto CAD: Enrolled students - 30, Seats left - 50
EV: Enrolled students - 40, Seats left - 40
Ln: 260 Col: 4

Explanation: The course management system is designed to facilitate the administration and management of various courses within an educational institution or organization. It provides a set of functionalities to input, display, and analyze course details. Here's an explanation of its key components:

1. **Course Details Input**: Users are prompted to input details for each course, including the course name, fee, number of enrolled students, total number of seats, and duration. This ensures that all relevant information about each course is captured accurately.
2. **Display Course Details**: The program can print a formatted table displaying all course details. This includes information such as the course name, fee, number of enrolled students, seats left, and course duration. This feature allows users to quickly review the details of all courses in one place.
3. **Query Specific Course Details**: Users can query specific details of a particular course by inputting its name. The system then retrieves and displays information such as the fee, enrolled students, seats left, and duration for that course. This functionality provides targeted information for individual courses.
4. **Query Course Fee**: Users can input a course name to query and display the fee for that specific course. This allows users to quickly find out the cost of a particular course.
5. **Query Courses Enrollment**: The system provides a feature to display enrollment details for each course. This includes information about the number of enrolled students and available seats for each course. This functionality helps users monitor course enrollments and availability.
6. **Check Course Existence**: Users can input a course name to check if it exists in the system. This feature ensures that users can verify whether a course is included in the system's database.
7. **Query Seats Left**: The system displays the number of seats left for each course. This information helps users understand the availability of seats for each course and make informed decisions about enrollment.
8. **Analysis Functions**: The program includes functions for analyzing course data, such as identifying courses with fewer than 50 seats or courses with fees above or below a certain threshold. These functions provide valuable insights into the distribution of courses based on various criteria.
9. **User Interaction Loop**: The program runs in a loop, allowing users to perform multiple operations sequentially without restarting the program. This enhances user experience and efficiency by enabling seamless interaction with the system.

Overall, the course management system offers a comprehensive set of functionalities for managing and analyzing course details, making it a valuable tool for educational institutions, administrators, and users involved in course administration and enrollment.

Conclusion:

The course navigation program helps for managing course details and providing various functionalities for users. Here's a concise conclusion:

The course navigation program efficiently handles course information, allowing users to input, view, and analyze data with ease. It presents a user-friendly interface, guiding users through options to perform tasks such as querying fees, checking seat availability, and displaying enrollment details.

Overall, the course navigation program serves as a valuable tool for administrators or users to organize, track, and analyze course data effectively. It sets a strong foundation for further enhancements and customization to meet evolving needs and requirements in educational settings.