



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](http://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.GANESH**

**21KQ1A0234**

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 type of courses and students are needed to enroll the courses and to know details the following process is usefull.

**1.Description:** At ABC Institute,we offer a variety of courses with different costs Each course has a limit of members and students. Our courses are designed to provide high - quality education. If a student comes providing this project gives a variety of options,we can change with respect to course and cost .

The program provided is focused on managing course details such as names, fees, enrollment, available seats, and timings, primarily intended for administrators or instructors to manage courses within an educational institution. It lacks features specific to student navigation through courses, which are characteristic of a Course Navigator Probe.

## Requirements

### 2.1 Functional Requirements

- **Add course details:** Users can input course name, fee, enrollment, total seats, and timings.
- **Query course fee:** Users can enter the name of a course to display its fee.
- **Query available seats:** Users can view courses with available seats.
- **Display specific course details:** Users can input a course name to display its detailed information, including fee, enrolled students, seats left, and timings.

### 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:

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

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

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

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

### **3. 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.

### **4.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.

### **5. 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.

## **6. 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.

## **7. 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.

## **8. 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.

## **9. 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.

## PROGRAM:

```
4.py - C:\Users\Batu\OneDrive\Documents\python project\4.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 Duration"))
    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
    return False
```

```

c4.py - C:\Users\Babu\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"Total 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:
            count += 1

```

```

c4.py - C:\Users\Babu\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"Total Number of Courses: {len(course_details)}")

def query_total_seats(course_details):
    total_seats = sum(course[3] for course in course_details)
    print(f"Total 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"Enter 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':
            query_courses_enrollment(course_details)

```

```

c4.py - C:\Users\Babul\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:

```
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
Enter number of enrolled students: 50

Enter details for Course 6:
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
Enter number of enrolled students: 100
```



```
IDE 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
Enter number of enrolled students: 20
```

```
IDE 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: VL68I
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
Enter number of enrolled students: 50
```

```
IDE 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 structurs | $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 |
| VLSI | $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
```

```
IDE 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 structurs
The fee for Data structurs 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 structurs: Enrolled students - 60, Seats left - 60
Auto CAD: Enrolled students - 30, Seats left - 50
EV: Enrolled students - 40, Seats left - 40
Ship design: Enrolled students - 30, Seats left - 30
Ln 368 Col 4
```

**Explanation:** The Course Navigator Probe is a sophisticated tool designed to help users explore and navigate the vast array of available courses, programs, and educational opportunities. It serves as a comprehensive resource for individuals seeking to further their education, advance their careers, or pursue personal interests.

**1. User Input:** The navigator probe begins by gathering input from the user regarding their preferences, goals, and requirements. This input may include factors such as the subject area of interest, level of study (e.g., beginner, intermediate, advanced), preferred learning format (e.g., online, in-person, hybrid), duration, location, and budget constraints.

**2. Search and Filtering:** Based on the user's input, the navigator probe conducts searches across a database of available courses and programs. It employs advanced search algorithms and filtering mechanisms to narrow down the options and present relevant results. Users can refine their search criteria to further tailor the results to their needs.

**3. Recommendations:** In addition to search results, the navigator probe provides personalized recommendations based on the user's profile, preferences, and browsing history. It leverages recommendation algorithms to suggest courses that align with the user's interests and goals, even if they may not have explicitly searched for them.

**4. Course Comparison:** The navigator probe offers tools for comparing key attributes of multiple courses side by side. Users can evaluate factors such as course content, duration, cost, accreditation, faculty credentials, student reviews, and career outcomes to make informed decisions.

**5. User Interaction:** Throughout the exploration process, the navigator probe engages users through interactive features such as user profiles, saved searches, favorites lists, and notifications. This enhances user engagement and facilitates a personalized experience tailored to each individual's needs and preferences.

**6. Continuous Improvement:** The navigator probe continually refines its search, recommendation, and comparison algorithms based on user feedback and usage patterns. It adapts to evolving user needs and preferences, ensuring that it remains a valuable and effective resource over time.

Overall, the Course Navigator Probe empowers users to navigate the complex landscape of educational offerings with confidence and ease. Whether they are searching for a specific course, exploring new learning opportunities, or planning their educational journey, the navigator probe serves as a trusted guide, helping users find the right path to achieve their goals.

**Conclusion:**

The Course Navigator Probe represents a significant advancement in assisting users in navigating the vast landscape of available courses. By leveraging advanced search, filtering, recommendation, and comparison features, it empowers users to make informed decisions tailored to their preferences and requirements. Throughout the development process, careful consideration was given to user needs, system architecture, data acquisition, feature implementation, and user interaction. The result is a robust and user-friendly tool that simplifies the process of exploring and selecting courses. With its intuitive interface and personalized recommendations, the Course Navigator Probe enhances the user experience, increases efficiency, and promotes lifelong learning. By providing access to a wealth of course information and facilitating decision-making, it serves as a valuable resource for students, professionals, and career counsellors alike.

As the Course Navigator Probe continues to evolve, ongoing evaluation, iteration, and maintenance will be essential to ensure its effectiveness and relevance in an ever-changing educational landscape. By staying responsive to user feedback, incorporating new features, and adapting to emerging trends, the navigator probe will remain a trusted companion for navigating the world of learning opportunities.