AI Assignment 3

Source Code-

```python
# career advisory system for a graduating student of IIITD based on courses done, grades got and interest areas.
# Assumption: courses of interest taken.

from durable.lang import *

# Knowledge base of courses available
data_scientist_courses = ['DBMS', 'Database System Implementation', 'Big Data Analytics']
ai_engineer_courses = ['AI', 'ML', 'Trustworthy AI Systems']
ml_engineer_courses = ['Statistical Machine Learning', 'Advanced Machine Learning', 'Machine Learning']
robotics_engineer_courses = ['Robotics', 'Social Robotics', 'Non Linear and Adaptive Control of Robotic Systems']
electronics_engineer_courses = ['Integrated Electronics', 'Circuit theory and devices', 'Fields and Waves']
bioinformatics_engineer_courses = ['Practical Bioinformatics', 'Algorithms in BioInformatics', 'Algorithms in Computational Biology']

career_interest_areas = ['Data Science Engineer', 'AI Engineer', 'ML Engineer', 'Robotics Engineer', 'Electronics Engineer', 'Bio-Informatics Engineer']


with ruleset('field_classification'):
    # DATA SCIENCE
    @when_all((m.interest == 'Data Science Engineer') & ((m.course1 == 'DBMS') & (m.course2 == 'Database System Implementation') & (m.course3 == 'Big Data Analytics'))
     & (m.cgpa >= 8))
    def data(c):
        # print('Heavy data fact asserted')
        c.assert_fact('career', {'field': 'data', 'type': 'heavy'})
    @when_all((m.interest == 'Data Science Engineer') & ((m.course1 ==
'DBMS') & (m.course2 == 'Database System Implementation') & (m.course3 ==
'Big Data Analytics'))
     & (m.cgpa < 8))
    def data(c):
        # print('light data fact asserted')
        c.assert_fact('career', {'field': 'data', 'type': 'light'})

    # ARTIFICIAL INTELLIGENCE
    @when_all((m.interest == 'AI Engineer') & ((m.course1 == 'AI') &
(m.course2 == 'ML') & (m.course3 == 'Trustworthy AI Systems')) & (m.cgpa
>= 8))
    def data(c):
        # print('Heavy ai fact asserted')
        c.assert_fact('career', {'field': 'ai', 'type': 'heavy'})

    @when_all((m.interest == 'AI Engineer') & ((m.course1 == 'AI') &
(m.course2 == 'ML') & (m.course3 == 'Trustworthy AI Systems')) & (m.cgpa <
8))
    def data(c):
        # print('light ai fact asserted')
        c.assert_fact('career', {'field': 'ai', 'type': 'light'})

    # MACHINE LEARNING
    @when_all((m.interest == 'ML Engineer') & ((m.course1 == 'Statistical
Machine Learning') & (m.course2 == 'Advanced Machine Learning') &
(m.course3 == 'Machine Learning'))
     & (m.cgpa >= 8))
    def data(c):
        # print('Heavy ml fact asserted')
        c.assert_fact('career', {'field': 'ml', 'type': 'heavy'})

    @when_all((m.interest == 'ML Engineer') & ((m.course1 == 'Statistical
Machine Learning') & (m.course2 == 'Advanced Machine Learning') &
(m.course3 == 'Machine Learning'))
     & (m.cgpa < 8))
    def data(c):
        # print('light ml fact asserted')
        c.assert_fact('career', {'field': 'ml', 'type': 'light'})
    # ROBOTICS
    @when_all((m.interest == 'Robotics Engineer') & ((m.course1 ==
'Robotics') & (m.course2 == 'Social Robotics') & (m.course3 == 'Non Linear
and Adaptive Control of Robotic Systems'))
     & (m.cgpa >= 8))
    def data(c):
        # print('Heavy robotics fact asserted')
        c.assert_fact('career', {'field': 'robotics', 'type': 'heavy'})

    @when_all((m.interest == 'Robotics Engineer') & ((m.course1 ==
'Robotics') & (m.course2 == 'Social Robotics') & (m.course3 == 'Non Linear
and Adaptive Control of Robotic Systems'))
     & (m.cgpa < 8))
    def data(c):
        # print('light robotics fact asserted')
        c.assert_fact('career', {'field': 'robotics', 'type': 'light'})

    # ELECTRONICS
    @when_all((m.interest == 'Electronics Engineer') & ((m.course1 ==
'Integrated Electronics') & (m.course2 == 'Circuit theory and devices') &
(m.course3 == 'Fields and Waves'))
     & (m.cgpa >= 8))
    def data(c):
        # print('Heavy electronics fact asserted')
        c.assert_fact('career', {'field': 'electronics', 'type': 'heavy'})

    @when_all((m.interest == 'Electronics Engineer') & ((m.course1 ==
'Integrated Electronics') & (m.course2 == 'Circuit theory and devices') &
(m.course3 == 'Fields and Waves'))
     & (m.cgpa < 8))
    def data(c):
        # print('light electronics fact asserted')
        c.assert_fact('career', {'field': 'electronics', 'type': 'light'})

    # BIO_INFORMATICS
    @when_all((m.interest == 'Bio-Informatics Engineer') & ((m.course1 ==
'Practical Bioinformatics') & (m.course2 == 'Algorithms in
BioInformatics')
     & (m.course3 == 'Algorithms in Computational Biology')) & (m.cgpa >=
8))
```

```
    def data(c):
        # print('Heavy bio-informatics fact asserted')
        c.assert_fact('career', {'field': 'bio-informatics', 'type':
'heavy'})

    @when_all((m.interest == 'Bio-Informatics Engineer') & ((m.course1 ==
'Practical Bioinformatics') & (m.course2 == 'Algorithms in
BioInformatics')
     & (m.course3 == 'Algorithms in Computational Biology')) & (m.cgpa <
8))
    def data(c):
        # print('light bio-informatics fact asserted')
        c.assert_fact('career', {'field': 'bio-informatics', 'type':
'light'})



with ruleset('career'):
    # Data Science field
    @when_all((m.field == 'data') & (m.type == 'heavy'))
    def data(d):
        d.assert_fact({ 'subject': 'Take', 'object': 'Data Science',
'predicate': 'as a career.'})

    @when_all((m.field == 'data') & (m.type == 'light'))
    def data(d):
        d.assert_fact({ 'subject': 'Take', 'object': 'Data Science',
'predicate': 'as a career (Tip: Improve your DSA for interviews).' })

    # AI Field
    @when_all((m.field == 'ai') & (m.type == 'heavy'))
    def ai(d):
        d.assert_fact({ 'subject': 'Take', 'object': 'Artificial
Intelligence', 'predicate': 'as a career.' })

    @when_all((m.field == 'ai') & (m.type == 'light'))
    def ai(d):
        d.assert_fact({ 'subject': 'Take', 'object': 'Artificial
Intelligence', 'predicate': 'as a career (Tip: Improve your DSA for
interviews).' })
```

```python
    # ML Field
    @when_all((m.field == 'ml') & (m.type == 'heavy'))
    def ml(d):
        d.assert_fact({ 'subject': 'Take', 'object': 'Machine Learning',
'predicate': 'as a career.' })

    @when_all((m.field == 'ml') & (m.type == 'light'))
    def ml(d):
        d.assert_fact({ 'subject': 'Take', 'object': 'Machine Learning',
'predicate': 'as a career (Tip: Improve your DSA for interviews).' })

    # Robotics Field
    @when_all((m.field == 'robotics') & (m.type == 'heavy'))
    def robotics(d):
        d.assert_fact({ 'subject': 'Take', 'object': 'Robotics',
'predicate': 'as a career.' })

    @when_all((m.field == 'robotics') & (m.type == 'light'))
    def robotics(d):
        d.assert_fact({ 'subject': 'Take', 'object': 'Robotics',
'predicate': 'as a career (Tip: Improve your DSA for interviews).' })

    # Electronics Field
    @when_all((m.field == 'electronics') & (m.type == 'heavy'))
    def electronics(d):
        d.assert_fact({ 'subject': 'Take', 'object': 'Electronics',
'predicate': 'as a career.' })

    @when_all((m.field == 'electronics') & (m.type == 'light'))
    def electronics(d):
        d.assert_fact({ 'subject': 'Take', 'object': 'Electronics',
'predicate': 'as a career (Tip: Improve your DSA for interviews).' })

    # Bio-Informatics Field
    @when_all((m.field == 'bio-informatics') & (m.type == 'heavy'))
    def bio(d):
        d.assert_fact({ 'subject': 'Take', 'object': 'Bio-Informatics',
'predicate': 'as a career.' })
```

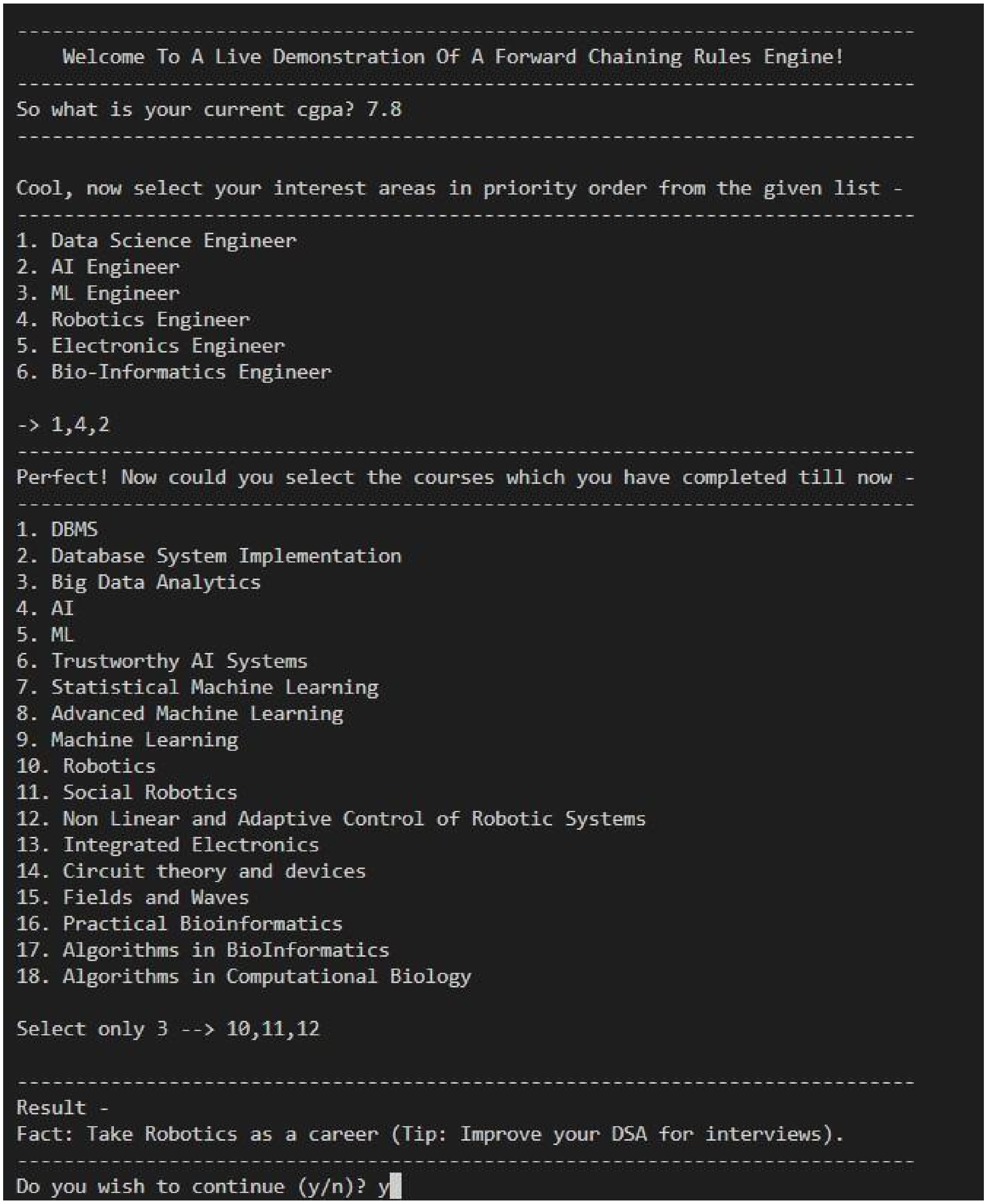
```
    @when_all((m.field == 'bio-informatics') & (m.type == 'light'))
    def bio(d):
        d.assert_fact({ 'subject': 'Take', 'object': 'Bio-Informatics',
'predicate': 'as a career (Tip: Improve your DSA for interviews).' })

    # Output
    @when_all(+m.subject)
    def output(c):
        print('Fact: {0} {1} {2}'.format(c.m.subject, c.m.object,
c.m.predicate))


def get_courses_done():
    list_of_courses = [data_scientist_courses, ai_engineer_courses,
ml_engineer_courses, robotics_engineer_courses,
electronics_engineer_courses,
    bioinformatics_engineer_courses]

    count = 1
    for course_list in list_of_courses:
        for course in course_list:
            print(str(count)+". "+course)
            count += 1
    courses_done = input('\nSelect only 3 --> ').strip('').split(',')

    for i in range(len(courses_done)):
        course_no = int(courses_done[i])
        if course_no > len(data_scientist_courses):
            if course_no >
len(data_scientist_courses)+len(ai_engineer_courses):
                if course_no >
len(data_scientist_courses)+len(ai_engineer_courses)+len(ml_engineer_cours
es):
                    if course_no >
len(data_scientist_courses)+len(ai_engineer_courses)+len(ml_engineer_cours
es)+len(robotics_engineer_courses):
                        if course_no >
len(data_scientist_courses)+len(ai_engineer_courses)+len(ml_engineer_cours
es)+len(robotics_engineer_courses)+len(electronics_engineer_courses):
```

```
                            index =
course_no-1-len(data_scientist_courses)-len(ai_engineer_courses)-len(ml_en
gineer_courses)-len(robotics_engineer_courses)-len(electronics_engineer_co
urses)
                            courses_done[i] =
bioinformatics_engineer_courses[index]
                        else:
                            index =
course_no-1-len(data_scientist_courses)-len(ai_engineer_courses)-len(ml_en
gineer_courses)-len(robotics_engineer_courses)
                            courses_done[i] =
electronics_engineer_courses[index]
                    else:
                        index =
course_no-1-len(data_scientist_courses)-len(ai_engineer_courses)-len(ml_en
gineer_courses)
                        courses_done[i] = robotics_engineer_courses[index]
                else:
                    index =
course_no-1-len(data_scientist_courses)-len(ai_engineer_courses)
                    courses_done[i] = ml_engineer_courses[index]
            else:
                index = course_no-1-len(data_scientist_courses)
                courses_done[i] = ai_engineer_courses[index]
        else:
            index = course_no-1
            courses_done[i] = data_scientist_courses[index]

    return courses_done


def printline():

print('-------------------------------------------------------------------
----------')


def start():
    print("""
-----------------------------------------------------------------------
---
    Welcome To A Live Demonstration Of A Forward Chaining Rules Engine!
-----------------------------------------------------------------------
---""")

    cgpa = float(input("So what is your current cgpa? "))
    printline()
    print('\nCool, now select your interest areas in priority order from
the given list -')
    printline()
    count = 1
    for interest in career_interest_areas:
        print(str(count)+". "+interest)
        count += 1
    interested_areas = input('\n-> ').strip('').split(',')
    ptr = 0
    for i in interested_areas:
        interested_areas[ptr] = career_interest_areas[int(i)-1]
        ptr += 1

    printline()
    print('Perfect! Now could you select the courses which you have
completed till now - ')
    printline()
    courses_done = get_courses_done()
    print()
    printline()
    print('Result -')

    # asserting fact(s)
    no_of_exceptions_risen = 0
    i = 0
    for interest in interested_areas:
        try:
            assert_fact('field_classification', {'interest': interest,
'course1': courses_done[0], 'course2': courses_done[1], 'course3':
courses_done[2], 'cgpa': cgpa})
        except Exception as e:
            no_of_exceptions_risen += 1
```

```python
            pass
        i += 1

    if no_of_exceptions_risen == len(interested_areas):
        print('Unfortunately you did not complete all base courses of any particular interest, so complete your courses first!!')
    printline()


start()
end_loop = input('Do you wish to continue (y/n)? ')
while end_loop == 'y':
    start()
    end_loop = input('Do you wish to continue (y/n)? ')
```

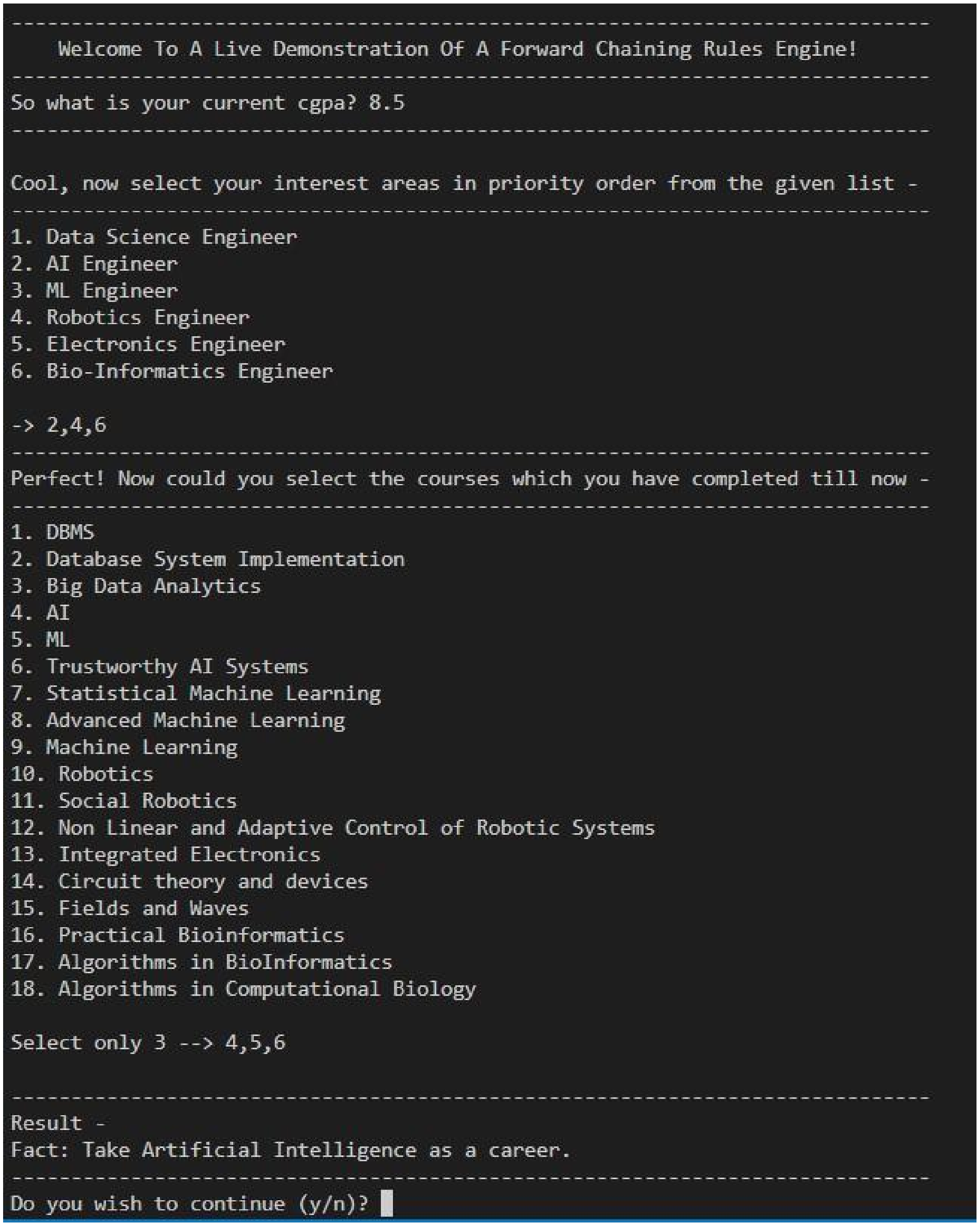
Screenshot Of Output

1. For cgpa less than 8



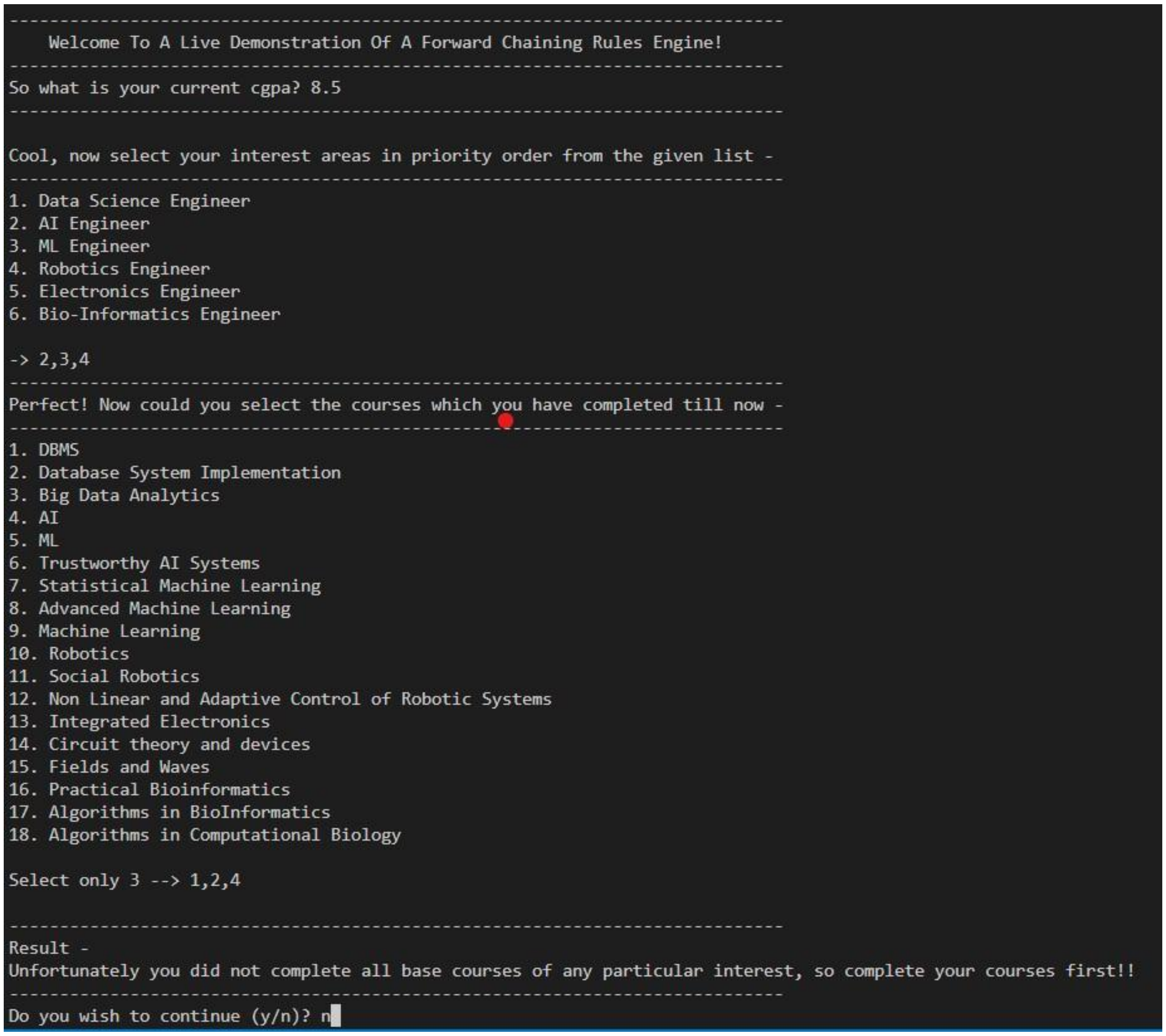
```
-----------------------------------------------------------------------------
    Welcome To A Live Demonstration Of A Forward Chaining Rules Engine!
-----------------------------------------------------------------------------
So what is your current cgpa? 7.8
-----------------------------------------------------------------------------

Cool, now select your interest areas in priority order from the given list -
-----------------------------------------------------------------------------
1. Data Science Engineer
2. AI Engineer
3. ML Engineer
4. Robotics Engineer
5. Electronics Engineer
6. Bio-Informatics Engineer

-> 1,4,2
-----------------------------------------------------------------------------
Perfect! Now could you select the courses which you have completed till now -
-----------------------------------------------------------------------------
1. DBMS
2. Database System Implementation
3. Big Data Analytics
4. AI
5. ML
6. Trustworthy AI Systems
7. Statistical Machine Learning
8. Advanced Machine Learning
9. Machine Learning
10. Robotics
11. Social Robotics
12. Non Linear and Adaptive Control of Robotic Systems
13. Integrated Electronics
14. Circuit theory and devices
15. Fields and Waves
16. Practical Bioinformatics
17. Algorithms in BioInformatics
18. Algorithms in Computational Biology

Select only 3 --> 10,11,12

-----------------------------------------------------------------------------
Result -
Fact: Take Robotics as a career (Tip: Improve your DSA for interviews).
-----------------------------------------------------------------------------
Do you wish to continue (y/n)? y
```

2. For cgpa greater than 8



```
-----------------------------------------------------------------------------
    Welcome To A Live Demonstration Of A Forward Chaining Rules Engine!
-----------------------------------------------------------------------------
So what is your current cgpa? 8.5
-----------------------------------------------------------------------------


Cool, now select your interest areas in priority order from the given list -
-----------------------------------------------------------------------------
1. Data Science Engineer
2. AI Engineer
3. ML Engineer
4. Robotics Engineer
5. Electronics Engineer
6. Bio-Informatics Engineer

-> 2,4,6
-----------------------------------------------------------------------------
Perfect! Now could you select the courses which you have completed till now -
-----------------------------------------------------------------------------
1. DBMS
2. Database System Implementation
3. Big Data Analytics
4. AI
5. ML
6. Trustworthy AI Systems
7. Statistical Machine Learning
8. Advanced Machine Learning
9. Machine Learning
10. Robotics
11. Social Robotics
12. Non Linear and Adaptive Control of Robotic Systems
13. Integrated Electronics
14. Circuit theory and devices
15. Fields and Waves
16. Practical Bioinformatics
17. Algorithms in BioInformatics
18. Algorithms in Computational Biology

Select only 3 --> 4,5,6

-----------------------------------------------------------------------------
Result -
Fact: Take Artificial Intelligence as a career.
-----------------------------------------------------------------------------
Do you wish to continue (y/n)?
```

3. For handling cases where a student hasn’t completed all base courses of any interest



```
-----------------------------------------------------------------------------
    Welcome To A Live Demonstration Of A Forward Chaining Rules Engine!
-----------------------------------------------------------------------------
So what is your current cgpa? 8.5
-----------------------------------------------------------------------------

Cool, now select your interest areas in priority order from the given list -
-----------------------------------------------------------------------------
1. Data Science Engineer
2. AI Engineer
3. ML Engineer
4. Robotics Engineer
5. Electronics Engineer
6. Bio-Informatics Engineer

-> 2,3,4
-----------------------------------------------------------------------------
Perfect! Now could you select the courses which you have completed till now -
-----------------------------------------------------------------------------
1. DBMS
2. Database System Implementation
3. Big Data Analytics
4. AI
5. ML
6. Trustworthy AI Systems
7. Statistical Machine Learning
8. Advanced Machine Learning
9. Machine Learning
10. Robotics
11. Social Robotics
12. Non Linear and Adaptive Control of Robotic Systems
13. Integrated Electronics
14. Circuit theory and devices
15. Fields and Waves
16. Practical Bioinformatics
17. Algorithms in BioInformatics
18. Algorithms in Computational Biology

Select only 3 --> 1,2,4

-----------------------------------------------------------------------------
Result -
Unfortunately you did not complete all base courses of any particular interest, so complete your courses first!!
-----------------------------------------------------------------------------
Do you wish to continue (y/n)? n
```