Assignment 3(AI)

Ques: Write a python program using durable-rules module with forward-chaining rules for course-and -extracurricular activities suggestion system for a non-graduating student of IIITD based on grades and interests. Make your own rules and test it out with facts.

File Submitted: I have run this program on google colab so submitting both the files .py as well as .ipynb file.

- AI_A3_Sakshi_MT21141.py:- Python file
- AI_A3_Sakshi_MT21141.ipynb:- colab file

Overview of code written:

Rules:-

- A rule is the fundamental structure square of durable_rules. The rule antecedent characterizes the conditions that should be fulfilled to execute the rule consequent (action).
- 'When_all' clarifies the antecedent meaning of a rule.

Facts:-

- Facts address the information that characterizes a knowledge base. Facts are declared as JSON objects and stored until they are retracted. At the point when a Fact fulfills a rule antecedent, the rule consequent is executed and this property is known as forward chaining.

Here I have defined three rulesets in my code :-

1. interests
2. Skills
3. Extracurricular_activities

I have made rules for 10 courses and also suggest Extracurricular_activities for them.

Code :-

```
#install the 'durable-rules' library to run this file

pip install durable-rules


from durable.lang import *


with ruleset('interests'):

    # will be triggered by 'interests' facts

    #facts defined for different courses here

    #for programming course

    @when_all((m.course_of_study == 'programming') & (m.type == 'practical') )

    def programingc(c):

        c.assert_fact('skills', { 'knowledge':'problem_solving' })

        c.assert_fact({ 'subject': 'choose', 'predicate': 'elective', 'object': 'programming and
problem solving skills' })

        c.assert_fact('extracurricular_activities', { 'type': 'online_coding'
,'course':'programming'})


    #knowledge/facts for algorithm course

    @when_all((m.course_of_study == 'algorithm') & (m.type == 'theory') )

    def programingc(c):

        c.assert_fact('skills', { 'knowledge':'problem_solving_algo' })
```

```
        c.assert_fact({ 'subject': 'choose', 'predicate': 'elective', 'object': 'Data Structures
and Algorithms' })

        c.assert_fact('extracurricular_activities', { 'type': 'online_coding'
,'course':'programming'})



    #knowledge/facts for operating system course

    @when_all((m.course_of_study == 'OS') & (m.type == 'theory') )

    def operating_system(c):

        c.assert_fact('skills', { 'knowledge':'computer_functionality' })

        c.assert_fact({ 'subject': 'choose', 'predicate': 'elective', 'object': 'Operating system'
})

        c.assert_fact('extracurricular_activities', { 'type': 'read' ,'course':'OS'})



    #knowledge/facts for computer architecture course

    @when_all((m.course_of_study == 'Computer_architecture') & (m.type == 'theory') )

    def operating_system(c):

        c.assert_fact('skills', { 'knowledge':'computer_functionality' })

        c.assert_fact({ 'subject': 'choose', 'predicate': 'elective', 'object': 'Computer
Architecture' })

        c.assert_fact('extracurricular_activities', { 'type': 'read'
,'course':'Computer_architecture'})



    #knowledge/facts for computer network course

    @when_all((m.course_of_study == 'CN') & (m.type == 'practical') )

    def computer_network(c):
```

```
        c.assert_fact('skills', { 'knowledge':'problem_solving' })

        c.assert_fact({ 'subject': 'choose', 'predicate': 'elective', 'object': 'Computer Network
practical' })

        c.assert_fact('extracurricular_activities', { 'type': 'workshop' ,'course':'CN'})


    #knowledge/facts for computer network course

    @when_all((m.course_of_study == 'CN') & (m.type == 'theory') )

    def computer_network(c):

        c.assert_fact('skills', { 'knowledge':'computer_functionality' })

        c.assert_fact({ 'subject': 'choose', 'predicate': 'elective', 'object': 'Computer Network'
})

        c.assert_fact('extracurricular_activities', { 'type': 'read' ,'course':'CN'})


    #knowledge/facts for DBMS course

    @when_all((m.course_of_study == 'DBMS') & (m.type == 'theory') )

    def database(c):

        c.assert_fact('skills', { 'knowledge':'SQL' })

        c.assert_fact({ 'subject': 'choose', 'predicate': 'elective', 'object': 'Database
management system' })

        c.assert_fact('extracurricular_activities', { 'type': 'read' ,'course':'DBMS'})


    @when_all((m.course_of_study == 'DBMS') & (m.type == 'practical') )

    def database(c):

        c.assert_fact('skills', { 'knowledge':'db_programming' })
```

```
        c.assert_fact({ 'subject': 'choose', 'predicate': 'elective', 'object': 'Database management system practical' })

        c.assert_fact('extracurricular_activities', { 'type': 'online_competition' ,'course':'DBMS'})


    #knowledge/facts for web_devlopment course

    @when_all((m.course_of_study == 'web_development') & (m.type == 'practical') )

    def webdev(c):

        c.assert_fact('skills', {'knowledge':'html_css_javascript'})

        c.assert_fact({'subject': 'choose', 'predicate': 'elective', 'object': 'web development'})

        c.assert_fact('extracurricular_activities', { 'type': 'develop_website' ,'course':'web_development'})


    #knowledge/facts for AI course

    @when_all((m.course_of_study == 'AI') & (m.type == 'theory') )

    def aritificial_intelligence(c):

        c.assert_fact('skills', { 'knowledge':'Statical_Mathematics' })

        c.assert_fact({ 'subject': 'choose', 'predicate': 'elective', 'object': 'Artificial intelligence' })

        c.assert_fact('extracurricular_activities', { 'type': 'conference' ,'course':'AI'})




    @when_all((m.course_of_study == 'AI') & (m.type == 'practical') )

    def aritificial_intelligence(c):

        c.assert_fact('skills', { 'knowledge':'declarative_programming' })
```

```
        c.assert_fact({ 'subject': 'choose', 'predicate': 'elective', 'object': 'Artificial
intelligence practical' })

        c.assert_fact('extracurricular_activities', { 'type': 'research','course':'AI' })



    #knowledge/facts for ML course

    @when_all((m.course_of_study == 'ML') & (m.type == 'theory') )

    def machine_learning(c):

        c.assert_fact('skills', { 'knowledge':'Statical_Mathematics' })

        c.assert_fact({ 'subject': 'choose', 'predicate': 'elective', 'object': 'Machine Learning'
})

        c.assert_fact('extracurricular_activities', { 'type': 'conference' ,'course':'ML'})



    @when_all((m.course_of_study == 'ML') & (m.type == 'practical') )

    def machine_learning(c):

        c.assert_fact('skills', { 'knowledge':'python_programming' })

        c.assert_fact({ 'subject': 'choose', 'predicate': 'elective', 'object': 'Machine Learning
practical' })

        c.assert_fact('extracurricular_activities', { 'type': 'research','course':'ML' })



    #knowledge/facts for Data_science course

    #internally i have used the AI and ML skills and extracurricular_activities for this
course

    @when_all((m.course_of_study == 'Data_Science') & (m.type == 'theory') )

    def data_science(c):

        c.assert_fact('skills', { 'knowledge':'Statical_Mathematics' })
```

```
        c.assert_fact({ 'subject': 'choose', 'predicate': 'elective', 'object': 'Data Science' })
        c.assert_fact('extracurricular_activities', { 'type': 'conference' ,'course':'AI'})
        c.assert_fact('extracurricular_activities', { 'type': 'conference' ,'course':'ML'})

    @when_all((m.course_of_study == 'Data_Science') & (m.type == 'practical') )
    def data_science(c):
        c.assert_fact('skills', { 'knowledge':'python_programming' })
        c.assert_fact('skills', { 'knowledge':'declarative_programming' })
        c.assert_fact({ 'subject': 'choose', 'predicate': 'elective', 'object': 'Data Science
practical' })
        c.assert_fact('extracurricular_activities', { 'type': 'research','course':'ML' })
        c.assert_fact('extracurricular_activities', { 'type': 'research','course':'AI' })



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


#skill ruleset defined here and different facts corresponding to different courses are
defined
with ruleset('skills'):
    @when_all((m.knowledge == 'problem_solving'))
    def problemc(d):
```

```
        d.assert_fact({ 'subject': 'learn how to solve problems by doing codes' })


    @when_all((m.knowledge == 'problem_solving_algo'))
    def problemc(d):
        d.assert_fact({ 'subject': 'learn how to design proper algorithm for given problem to
solve it' })


    @when_all((m.knowledge == 'html_css_javascript'))
    def webc(d):
        d.assert_fact({ 'subject': 'practice html,css and javascript codes' })


    @when_all((m.knowledge == 'computer_functionality'))
    def osc(d):
        d.assert_fact({ 'subject': 'Learn basic of computer functionality.' })


    @when_all((m.knowledge == 'SQL'))
    def sqlc(d):
        d.assert_fact({ 'subject': 'take some course to learn SQL' })


    @when_all((m.knowledge == 'db_programming'))
    def programc(d):
        d.assert_fact({ 'subject': 'take some course to learn database programming' })


    @when_all((m.knowledge == 'Mathematics'))
```

```
    def mathc(d):
        d.assert_fact({ 'subject': 'take Mathematics and clear the concept of Statical mathematics' })

    @when_all((m.knowledge == 'declarative_programming'))
    def programc(d):
        d.assert_fact({ 'subject': 'take course to learn python programming' })

    @when_all((m.knowledge == 'Statical_Mathematics'))
    def mathc(d):
        d.assert_fact({ 'subject': 'take Mathematics and clear the concept of statics, probability, algebra and calculus' })

    @when_all((m.knowledge == 'python_programming'))
    def programc(d):
        d.assert_fact({ 'subject': 'take course to learn python programming' })

     #printing the consequent
    @when_all(+m.subject)
    def output(d):
        print('Fact: {0}'.format(d.m.subject))
```

#extracurricular_activities ruleset is defined below and different facts corresponding to different courses are defined

```
with ruleset('extracurricular_activities'):

    @when_all((m.type == 'online_coding')&(m.course == 'programming'))

    def coding(e):

        e.assert_fact({ 'subject': 'Practice problem-solving online in any programming
language '})



    @when_all((m.type == 'develop_website')&(m.course == 'web_development'))

    def web_dev(e):

        e.assert_fact({ 'subject': 'Try building small website or participants in hackthons
related to web development'})



    @when_all((m.type == 'read')&(m.course == 'Computer_architecture'))

    def read(e):

        e.assert_fact({ 'subject': 'Read some books to gain knowledge about internal
processing done by computer and about processors evolution. '})



    @when_all((m.type == 'read')&(m.course == 'OS'))

    def read(e):

        e.assert_fact({ 'subject': 'Read some books to gain knowledge about different
operating System. '})



    @when_all((m.type == 'read')&(m.course == 'CN'))

    def read(e):

        e.assert_fact({ 'subject': 'Read some books to gain knowledge about computer
network. '})
```

```
    @when_all((m.type == 'workshop')&(m.course == 'CN'))

    def workshopc(e):

        e.assert_fact({ 'subject': 'participants in workshops of network also try to do some
server programming'})



    @when_all((m.type == 'read')&(m.course == 'DBMS'))

    def read(e):

        e.assert_fact({ 'subject': 'Read some books to gain knowledge about database
management. '})



    @when_all((m.type == 'online_competition')&(m.course == 'DBMS'))

    def db(e):

        e.assert_fact({ 'subject': 'participants in online competition and work directly on
database'})



    @when_all((m.type == 'conference')&(m.course == 'AI'))

    def confc(e):

        e.assert_fact({ 'subject': 'Attend some conference on AI Related topics and its
development. '})



    @when_all((m.type == 'research')&(m.course == 'AI'))

    def researchc(e):

        e.assert_fact({ 'subject': 'Do some research work in feild of AI to know more how it
works '})
```

```
    @when_all((m.type == 'conference')&(m.course == 'ML'))

    def conf(e):

        e.assert_fact({ 'subject': 'Attend some conference on Machine Learning Related topics and its development. '})



    @when_all((m.type == 'research')&(m.course == 'ML'))

    def researchc(e):

        e.assert_fact({ 'subject': 'Do some research work in feild of Machine Learning to know how it works '})



     #printing the consequent

    @when_all(+m.subject)

    def output(e):

        print('Fact: {0}'.format(e.m.subject))




#showing result of forward chaining

#either we can see one course at a time or all course at once we can see


#assert_fact('interests', { 'course_of_study': 'programming', 'type': 'practical' })

#assert_fact('interests', { 'course_of_study': 'DBMS', 'type': 'theory' })

#assert_fact('interests', { 'course_of_study': 'DBMS', 'type': 'practical' })

#assert_fact('interests', { 'course_of_study': 'AI', 'type': 'theory' })
```

#assert_fact('interests', { 'course_of_study': 'AI', 'type': 'practical' })

#assert_fact('interests', { 'course_of_study': 'ML', 'type': 'theory' })

#assert_fact('interests', { 'course_of_study': 'ML', 'type': 'practical' })

#assert_fact('interests', { 'course_of_study': 'Data_Science', 'type': 'theory' })

#assert_fact('interests', { 'course_of_study': 'Data_Science', 'type': 'practical' })

#assert_fact('interests', { 'course_of_study': 'web_development', 'type': 'practical' })

#assert_fact('interests', { 'course_of_study': 'OS', 'type': 'theory' })

#assert_fact('interests', { 'course_of_study': 'CN', 'type': 'theory' })

#assert_fact('interests', { 'course_of_study': 'CN', 'type': 'practical' })

#assert_fact('interests', { 'course_of_study': 'Computer_architecture', 'type': 'theory' })

assert_fact('interests', { 'course_of_study': 'algorithm', 'type': 'theory' })

Output :-

```
#assert_fact('interests', { 'course_of_study': 'programming', 'type': 'practical' })
#assert_fact('interests', { 'course_of_study': 'DBMS', 'type': 'theory' })
#assert_fact('interests', { 'course_of_study': 'DBMS', 'type': 'practical' })
#assert_fact('interests', { 'course_of_study': 'AI', 'type': 'theory' })
#assert_fact('interests', { 'course_of_study': 'AI', 'type': 'practical' })
#assert_fact('interests', { 'course_of_study': 'ML', 'type': 'theory' })
#assert_fact('interests', { 'course_of_study': 'ML', 'type': 'practical' })
#assert_fact('interests', { 'course_of_study': 'Data_Science', 'type': 'theory' })
#assert_fact('interests', { 'course_of_study': 'Data_Science', 'type': 'practical' })
#assert_fact('interests', { 'course_of_study': 'web_development', 'type': 'practical' })
#assert_fact('interests', { 'course_of_study': 'OS', 'type': 'theory' })
#assert_fact('interests', { 'course_of_study': 'CN', 'type': 'theory' })
#assert_fact('interests', { 'course_of_study': 'CN', 'type': 'practical' })
#assert_fact('interests', { 'course_of_study': 'Computer_architecture', 'type': 'theory' })
assert_fact('interests', { 'course_of_study': 'algorithm', 'type': 'theory' })
```

```
Fact: learn how to design proper algorithm for given problem to solve it
Fact: Practice problem-solving online in any programming language
Fact: choose elective Data Structures and Algorithms
{'$s': 1, 'id': 'sid-0', 'sid': '0'}
```

```
#assert_fact('interests', { 'course_of_study': 'programming', 'type': 'practical' })
#assert_fact('interests', { 'course_of_study': 'DBMS', 'type': 'theory' })
#assert_fact('interests', { 'course_of_study': 'DBMS', 'type': 'practical' })
#assert_fact('interests', { 'course_of_study': 'AI', 'type': 'theory' })
#assert_fact('interests', { 'course_of_study': 'AI', 'type': 'practical' })
#assert_fact('interests', { 'course_of_study': 'ML', 'type': 'theory' })
#assert_fact('interests', { 'course_of_study': 'ML', 'type': 'practical' })
#assert_fact('interests', { 'course_of_study': 'Data_Science', 'type': 'theory' })
#assert_fact('interests', { 'course_of_study': 'Data_Science', 'type': 'practical' })
#assert_fact('interests', { 'course_of_study': 'web_development', 'type': 'practical' })
#assert_fact('interests', { 'course_of_study': 'OS', 'type': 'theory' })
assert_fact('interests', { 'course_of_study': 'CN', 'type': 'theory' })
#assert_fact('interests', { 'course_of_study': 'CN', 'type': 'practical' })
#assert_fact('interests', { 'course_of_study': 'Computer_architecture', 'type': 'theory' })
#assert_fact('interests', { 'course_of_study': 'algorithm', 'type': 'theory' })
```

```
Fact: Learn basic of computer functionality.
Fact: Read some books to gain knowledge about computer network.
Fact: choose elective Computer Network
{'$s': 1, 'id': 'sid-0', 'sid': '0'}
```

```
#assert_fact('interests', { 'course_of_study': 'programming', 'type': 'practical' })
#assert_fact('interests', { 'course_of_study': 'DBMS', 'type': 'theory' })
#assert_fact('interests', { 'course_of_study': 'DBMS', 'type': 'practical' })
#assert_fact('interests', { 'course_of_study': 'AI', 'type': 'theory' })
#assert_fact('interests', { 'course_of_study': 'AI', 'type': 'practical' })
#assert_fact('interests', { 'course_of_study': 'ML', 'type': 'theory' })
#assert_fact('interests', { 'course_of_study': 'ML', 'type': 'practical' })
#assert_fact('interests', { 'course_of_study': 'Data_Science', 'type': 'theory' })
#assert_fact('interests', { 'course_of_study': 'Data_Science', 'type': 'practical' })
#assert_fact('interests', { 'course_of_study': 'web_development', 'type': 'practical' })
#assert_fact('interests', { 'course_of_study': 'OS', 'type': 'theory' })
#assert_fact('interests', { 'course_of_study': 'CN', 'type': 'theory' })
#assert_fact('interests', { 'course_of_study': 'CN', 'type': 'practical' })
assert_fact('interests', { 'course_of_study': 'Computer_architecture', 'type': 'theory' })
#assert_fact('interests', { 'course_of_study': 'algorithm', 'type': 'theory' })
```

```
Fact: Learn basic of computer functionality.
Fact: Read some books to gain knowledge about internal processing done by computer and about processors evolution.
Fact: choose elective Computer Architecture
{'$s': 1, 'id': 'sid-0', 'sid': '0'}
```

```
#assert_fact('interests', { 'course_of_study': 'programming', 'type': 'practical' })
#assert_fact('interests', { 'course_of_study': 'DBMS', 'type': 'theory' })
#assert_fact('interests', { 'course_of_study': 'DBMS', 'type': 'practical' })
#assert_fact('interests', { 'course_of_study': 'AI', 'type': 'theory' })
#assert_fact('interests', { 'course_of_study': 'AI', 'type': 'practical' })
#assert_fact('interests', { 'course_of_study': 'ML', 'type': 'theory' })
#assert_fact('interests', { 'course_of_study': 'ML', 'type': 'practical' })
assert_fact('interests', { 'course_of_study': 'Data_Science', 'type': 'theory' })
#assert_fact('interests', { 'course_of_study': 'Data_Science', 'type': 'practical' })
#assert_fact('interests', { 'course_of_study': 'web_development', 'type': 'practical' })
#assert_fact('interests', { 'course_of_study': 'OS', 'type': 'theory' })
#assert_fact('interests', { 'course_of_study': 'CN', 'type': 'theory' })
#assert_fact('interests', { 'course_of_study': 'CN', 'type': 'practical' })
#assert_fact('interests', { 'course_of_study': 'Computer_architecture', 'type': 'theory' })
#assert_fact('interests', { 'course_of_study': 'algorithm', 'type': 'theory' })
```

```
Fact: take Mathematics and clear the concept of statics, probability, algebra and calculus
Fact: Attend some conference on AI Related topics and its development.
Fact: Attend some conference on Machine Learning Related topics and its development.
Fact: choose elective Data Science
{'$s': 1, 'id': 'sid-0', 'sid': '0'}
```