ASSIGNMENT 3 - REPORT

1. Code for the assignment on durable rules engine program for course and extracurricular activities suggestion system using forward chaining

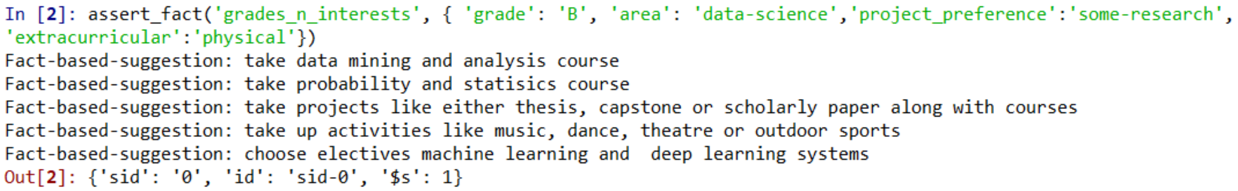
The code is as given below:

```
# course and extracurricular activities suggestion system
#using durable rules engine - python

# -*- coding: utf-8 -*-
"""
Created on Sun Oct 24 23:07:24 2021

@author: drishtide
"""

# Durable Rules Engine - with forward chaining
from durable.lang import *

with ruleset('grades_n_interests'):

    # will be triggered by 'grades_n_interests' facts
      # based on grades, area, project preference and extracurricular
activity preference the functions data() and software() print the fact
based suggestions for the students
      # grades are divided into two categories : (A,A-minus,B) and
(B-minus,C)
      # first type of grade gets advanced courses suggestion
      # second type of grade gets the general courses suggestion
      # there are two main areas for studies given to students: data
science and software engineering
      # two types of project preferences: either some project (thesis,
capstone or scholarly paper) or only courses
      # there are two different kinds of fields generated for the two
course areas : data mining and analysis or software development

    @when_all((m.grade == 'B-') | (m.grade == 'C') & (m.area ==
'data-science') & (m.project_preference == 'some-research') &
(m.extracurricular == 'physical'))
    def data(c):
        c.assert_fact('skillset', { 'field': 'data-mining-analysis' })
        c.assert_fact({ 'subject': 'choose', 'predicate': 'electives',
'object': 'information retrieval and database management systems' })
        c.assert_fact('project-preferences',{'project':'some-research'})

      c.assert_fact('extracurr-preferences',{'extracurricular':'physical'
      })
    @when_all((m.grade == 'A') | (m.grade == 'A-minus') | (m.grade ==
'B') & (m.area == 'data-science') & (m.project_preference ==
'some-research') & (m.extracurricular == 'physical'))
    def data(c):
        c.assert_fact('skillset', { 'field': 'data-mining-analysis' })
        c.assert_fact({ 'subject': 'choose', 'predicate': 'electives',
'object': 'machine learning and  deep learning systems' })
        c.assert_fact('project-preferences',{'project':'some-research'})

        c.assert_fact('extracurr-preferences',{'extracurricular':'physical
        '})

    @when_all((m.grade == 'B-') | (m.grade == 'C') & (m.area ==
'data-science') & (m.project_preference == 'only-course') &
(m.extracurricular == 'physical'))
    def data(c):
        c.assert_fact('skillset', { 'field': 'data-mining-analysis' })
        c.assert_fact({ 'subject': 'choose', 'predicate': 'electives',
'object': 'information retrieval and database management systems' })

        c.assert_fact('project-preferences',{'project':'only-course'})
        c.assert_fact('extracurr-preferences',{'extracurricular':'physical
        '})

    @when_all((m.grade == 'A') | (m.grade == 'A-minus') | (m.grade ==
'B') & (m.area == 'data-science') & (m.project_preference ==
'only-course') & (m.extracurricular == 'physical'))
    def data(c):
        c.assert_fact('skillset', { 'field': 'data-mining-analysis' })
        c.assert_fact({ 'subject': 'choose', 'predicate': 'electives',
'object': 'machine learning and  deep learning systems' })
        c.assert_fact('project-preferences',{'project':'only-course'})

      c.assert_fact('extracurr-preferences',{'extracurricular':'physical'
      })

    @when_all((m.grade == 'B-') | (m.grade == 'C') & (m.area ==
'data-science') & (m.project_preference == 'some-research') &
(m.extracurricular == 'mental'))
    def data(c):
        c.assert_fact('skillset', { 'field': 'data-mining-analysis' })
        c.assert_fact({ 'subject': 'choose', 'predicate': 'electives',
'object': 'information retrieval and database management systems' })

      c.assert_fact('project-preferences',{'project':'some-research'})
      c.assert_fact('extracurr-preferences',{'extracurricular':'mental'})

    @when_all((m.grade == 'A') | (m.grade == 'A-minus') | (m.grade ==
'B') & (m.area == 'data-science') & (m.project_preference ==
'some-research') & (m.extracurricular == 'mental'))
```

```
    def data(c):
        c.assert_fact('skillset', { 'field': 'data-mining-analysis' })
        c.assert_fact({ 'subject': 'choose', 'predicate': 'electives',
'object': 'machine learning and  deep learning systems' })
        c.assert_fact('project-preferences',{'project':'some-research'})

      c.assert_fact('extracurr-preferences',{'extracurricular':'mental'})

    @when_all((m.grade == 'B-') | (m.grade == 'C') & (m.area ==
'data-science') & (m.project_preference == 'only-course') &
(m.extracurricular == 'mental'))
    def data(c):
        c.assert_fact('skillset', { 'field': 'data-mining-analysis' })
        c.assert_fact({ 'subject': 'choose', 'predicate': 'electives',
'object': 'information retrieval and database management systems' })
        c.assert_fact('project-preferences',{'project':'only-course'})

       c.assert_fact('extracurr-preferences',{'extracurricular':'mental'}
       )

    @when_all((m.grade == 'A') | (m.grade == 'A-minus') | (m.grade ==
'B') & (m.area == 'data-science') & (m.project_preference ==
'only-course') & (m.extracurricular == 'mental'))
    def data(c):
        c.assert_fact('skillset', { 'field': 'data-mining-analysis' })
        c.assert_fact({ 'subject': 'choose', 'predicate': 'electives',
'object': 'machine learning and  deep learning systems' })
        c.assert_fact('project-preferences',{'project':'only-course'})

        c.assert_fact('extracurr-preferences',{'extracurricular':'mental'}
        )

    @when_all((m.grade == 'B-') | (m.grade == 'C') & (m.area ==
'software-engg') & (m.project_preference == 'some-research') &
(m.extracurricular == 'physical'))
    def software(c):
        c.assert_fact('skillset', { 'field': 'software-development' })
        c.assert_fact({ 'subject': 'choose', 'predicate': 'electives',
'object': 'software program analysis and software development techniques'
})
        c.assert_fact('project-preferences',{'project':'some-research'})

      c.assert_fact('extracurr-preferences',{'extracurricular':'physical'
      })

    @when_all((m.grade == 'A') | (m.grade == 'A-minus') | (m.grade ==
'B') & (m.area == 'software-engg') & (m.project_preference ==
'some-research') & (m.extracurricular == 'physical'))
    def software(c):
        c.assert_fact('skillset', { 'field': 'software-development' })
        c.assert_fact({ 'subject': 'choose', 'predicate': 'electives',
'object': 'agile development techniques and advanced software development
processes' })
        c.assert_fact('project-preferences',{'project':'some-research'})

       c.assert_fact('extracurr-preferences',{'extracurricular':'physical
       '})

    @when_all((m.grade == 'B-') | (m.grade == 'C') & (m.area ==
'software-engg') & (m.project_preference == 'only-course') &
(m.extracurricular == 'physical'))
    def software(c):
        c.assert_fact('skillset', { 'field': 'software-development' })
        c.assert_fact({ 'subject': 'choose', 'predicate': 'electives',
'object': 'software program analysis and software development techniques'
})
        c.assert_fact('project-preferences',{'project':'only-course'})

      c.assert_fact('extracurr-preferences',{'extracurricular':'physical'
      })

    @when_all((m.grade == 'A') | (m.grade == 'A-minus') | (m.grade ==
'B') & (m.area == 'software-engg') & (m.project_preference ==
'only-course') & (m.extracurricular == 'physical'))
    def software(c):
        c.assert_fact('skillset', { 'field': 'software-development' })
        c.assert_fact({ 'subject': 'choose', 'predicate': 'electives',
'object': 'agile development techniques and advanced software development
processes' })
        c.assert_fact('project-preferences',{'project':'only-course'})

       c.assert_fact('extracurr-preferences',{'extracurricular':'physical
       '})

    @when_all((m.grade == 'B-') | (m.grade == 'C') & (m.area ==
'software-engg') & (m.project_preference == 'some-research') &
(m.extracurricular == 'mental'))
    def software(c):
        c.assert_fact('skillset', { 'field': 'software-development' })
        c.assert_fact({ 'subject': 'choose', 'predicate': 'electives',
'object': 'software program analysis and software development techniques'
})
        c.assert_fact('project-preferences',{'project':'some-research'})

       c.assert_fact('extracurr-preferences',{'extracurricular':'mental'}
       )

    @when_all((m.grade == 'A') | (m.grade == 'A-minus') | (m.grade ==
'B') & (m.area == 'software-engg') & (m.project_preference ==
'some-research') & (m.extracurricular == 'mental'))
    def software(c):
        c.assert_fact('skillset', { 'field': 'software-development' })
        c.assert_fact({ 'subject': 'choose', 'predicate': 'electives',
'object': 'agile development techniques and advanced software development
processes' })
        c.assert_fact('project-preferences',{'project':'some-research'})

        c.assert_fact('extracurr-preferences',{'extracurricular':'mental'}
        )

    @when_all((m.grade == 'B-') | (m.grade == 'C') & (m.area ==
'software-engg') & (m.project_preference == 'only-course') &
(m.extracurricular == 'mental'))
    def software(c):
        c.assert_fact('skillset', { 'field': 'software-development' })
        c.assert_fact({ 'subject': 'choose', 'predicate': 'electives',
'object': 'software program analysis and software development techniques'
})
        c.assert_fact('project-preferences',{'project':'only-course'})

        c.assert_fact('extracurr-preferences',{'extracurricular':'mental'}
        )

    @when_all((m.grade == 'A') | (m.grade == 'A-minus') | (m.grade ==
'B') & (m.area == 'software-engg') & (m.project_preference ==
'only-course') & (m.extracurricular == 'mental'))
    def software(c):
        c.assert_fact('skillset', { 'field': 'software-development' })
        c.assert_fact({ 'subject': 'choose', 'predicate': 'electives',
'object': 'agile development techniques and advanced software development
processes' })
        c.assert_fact('project-preferences',{'project':'only-course'})

        c.assert_fact('extracurr-preferences',{'extracurricular':'mental'}
        )

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


with ruleset('skillset'):
    @when_all((m.field == 'data-mining-analysis'))
    def skill(d):
        d.assert_fact({ 'subject': 'take probability and statisics
course' })
        d.assert_fact({ 'subject': 'take data mining and analysis course'
})

    @when_all((m.field == 'software-development'))
    def skill(d):
        d.assert_fact({ 'subject': 'take software engineering principles course' })

    @when_all(+m.subject)
    def output(d):
        print('Fact-based-suggestion: {0}'.format(d.m.subject))


with ruleset('project-preferences'):
    @when_all((m.project == 'some-research'))
    def pref(e):
        e.assert_fact({ 'subject': 'take projects like either thesis, capstone or scholarly paper along with courses'})

    @when_all((m.project == 'only-course'))
    def pref(e):
        e.assert_fact({ 'subject': 'take only courses based on your field'})

    @when_all(+m.subject)
    def output(e):
        print('Fact-based-suggestion: {0}'.format(e.m.subject))


with ruleset('extracurr-preferences'):
    @when_all((m.extracurricular == 'physical'))
    def extra(f):
        f.assert_fact({ 'subject': 'take up activities like music, dance, theatre or outdoor sports'})

    @when_all((m.extracurricular == 'mental'))
    def extra(f):
        f.assert_fact({ 'subject': 'take up activities like chess, finance, astronomy or graphic designing'})

    @when_all(+m.subject)
    def output(f):
        print('Fact-based-suggestion: {0}'.format(f.m.subject))
```

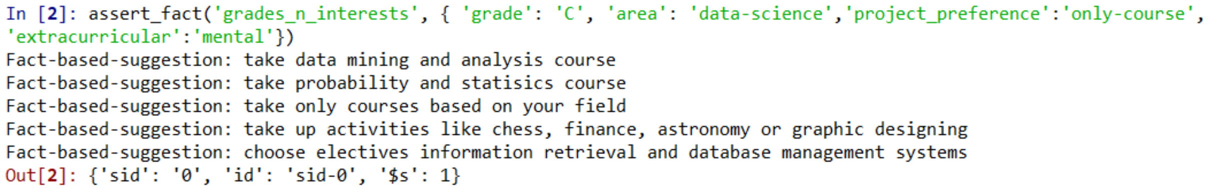
Outputs:

Example Query 1:



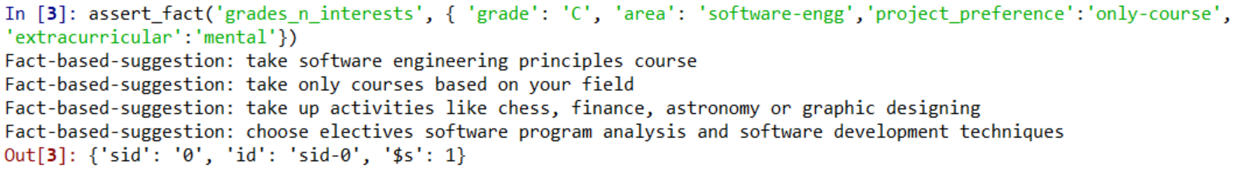
```
In [2]: assert_fact('grades_n_interests', { 'grade': 'B', 'area': 'data-science','project_preference':'some-research',
'extracurricular':'physical'})
Fact-based-suggestion: take data mining and analysis course
Fact-based-suggestion: take probability and statisics course
Fact-based-suggestion: take projects like either thesis, capstone or scholarly paper along with courses
Fact-based-suggestion: take up activities like music, dance, theatre or outdoor sports
Fact-based-suggestion: choose electives machine learning and  deep learning systems
Out[2]: {'sid': '0', 'id': 'sid-0', '$s': 1}
```

Example Query 2:



```
In [2]: assert_fact('grades_n_interests', { 'grade': 'C', 'area': 'data-science','project_preference':'only-course',
'extracurricular':'mental'})
Fact-based-suggestion: take data mining and analysis course
Fact-based-suggestion: take probability and statisics course
Fact-based-suggestion: take only courses based on your field
Fact-based-suggestion: take up activities like chess, finance, astronomy or graphic designing
Fact-based-suggestion: choose electives information retrieval and database management systems
Out[2]: {'sid': '0', 'id': 'sid-0', '$s': 1}
```

Example Query 3:



```
In [3]: assert_fact('grades_n_interests', { 'grade': 'C', 'area': 'software-engg','project_preference':'only-course',
'extracurricular':'mental'})
Fact-based-suggestion: take software engineering principles course
Fact-based-suggestion: take only courses based on your field
Fact-based-suggestion: take up activities like chess, finance, astronomy or graphic designing
Fact-based-suggestion: choose electives software program analysis and software development techniques
Out[3]: {'sid': '0', 'id': 'sid-0', '$s': 1}
```