AI – Assignment 3

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.

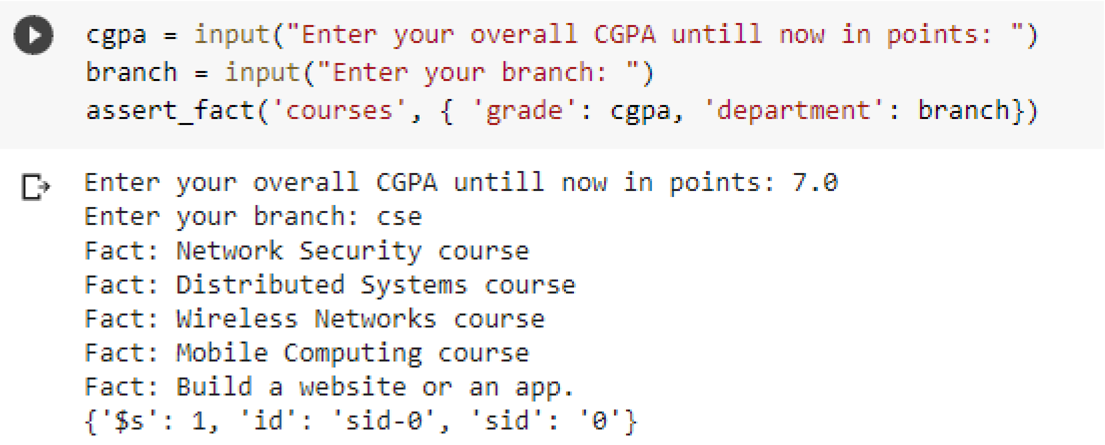
Assumption:

- User is a student of MTech in IIITD

Source Code:

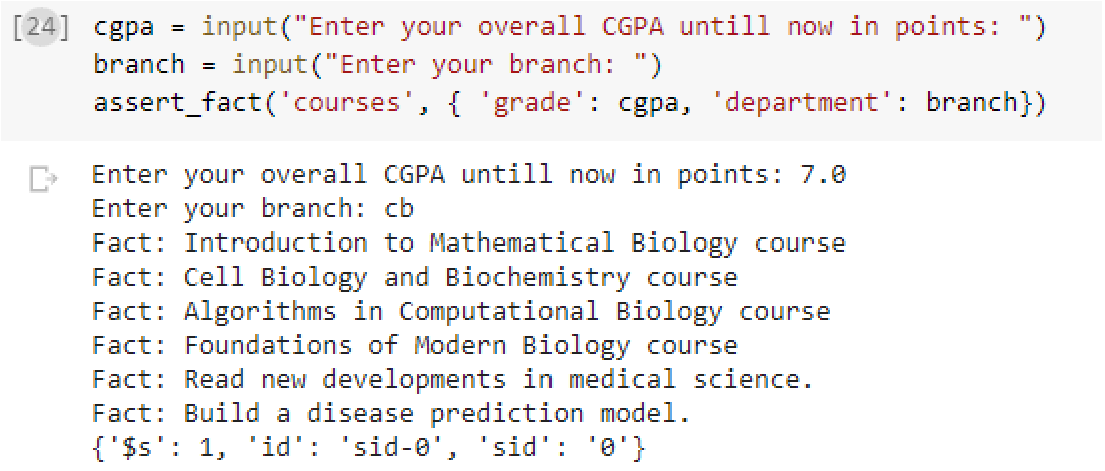
```
from durable.lang import ruleset, when_all, m, post, _main_host

if _main_host is not None:
    _main_host._ruleset_directory.clear()

from durable.lang import *

with ruleset('courses'):
  @when_all((m.grade == '9.0') & (m.department == 'cse') )
  def aiml(c):
        c.assert_fact('skill', { 'field': 'ai-ml' })
        c.assert_fact({ 'subject': 'Join', 'predicate': 'a', 'object': 'coding club.' })
        c.assert_fact({ 'subject': 'Build', 'predicate': 'a', 'object': 'predictive model.' })
        c.assert_fact({ 'subject': 'Travel', 'predicate': 'to', 'object': 'different places.' })

  @when_all((m.grade == '8.0') & (m.department == 'cse') )
  def de(c):
        c.assert_fact('skill', { 'field': 'Data Engineering' })
        c.assert_fact({ 'subject': 'Play', 'predicate': 'coding', 'object': 'games.' })
        c.assert_fact({ 'subject': 'Contribute', 'predicate': 'to', 'object': 'open source platforms.' })
        c.assert_fact({ 'subject': 'Travel', 'predicate': 'to', 'object': 'different places.' })

  @when_all((m.grade == '7.5') & (m.department == 'cse') )
  def info(c):
        c.assert_fact('skill', { 'field': 'Information Security' })
        c.assert_fact({ 'subject': 'Join', 'predicate': 'ethical hacking', 'object': 'club.' })
        c.assert_fact({ 'subject': 'Play', 'predicate': 'some', 'object': 'sport.' })

  @when_all((m.grade == '7.0') & (m.department == 'cse') )
  def mobile(c):
        c.assert_fact('skill', { 'field': 'Mobile Computing' })
        c.assert_fact({ 'subject': 'Build', 'predicate': 'a', 'object': 'website or an app.' })

  @when_all((m.grade == '8.0') & (m.department == 'ece') )
  def mobile(c):
        c.assert_fact('skill', { 'field': 'VLSI & Embedded Systems' })
        c.assert_fact({ 'subject': 'Play', 'predicate': 'some', 'object': 'sport.' })
        c.assert_fact({ 'subject': 'Travel', 'predicate': 'to', 'object': 'different places.' })


  @when_all((m.grade == '9.0') & (m.department == 'ece') )
  def mobile(c):
        c.assert_fact('skill', { 'field': 'Machine Learning' })
        c.assert_fact({ 'subject': 'Build', 'predicate': 'a', 'object': 'predictive model.' })
        c.assert_fact({ 'subject': 'Travel', 'predicate': 'to', 'object': 'different places.' })

  @when_all((m.grade == '7.0') & (m.department == 'ece') )
  def mobile(c):
        c.assert_fact('skill', { 'field': 'Cyber-Physical Systems' })
        c.assert_fact({ 'subject': 'Join', 'predicate': 'some', 'object': 'clubs.' })

  @when_all((m.grade == '7.0') & (m.department == 'cb') )
  def mobile(c):
        c.assert_fact('skill', { 'field': 'cb' })
        c.assert_fact({ 'subject': 'Build', 'predicate': 'a', 'object': 'disease prediction model.' })
        c.assert_fact({ 'subject': 'Read', 'predicate': 'new developments', 'object': 'in medical science.' })

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

with ruleset('skill'):
  @when_all((m.field == 'ai-ml'))
  def mathc(d):
        d.assert_fact({ 'subject': 'Artificial Intelligence course' })
        d.assert_fact({ 'subject': 'Machine Learning course' })
        d.assert_fact( {'subject': 'Reinforcement Learning course' })
        d.assert_fact( {'subject': 'Data Mining course' })

  @when_all((m.field == 'Information Security'))
  def mathc(d):
        d.assert_fact({ 'subject': 'Network Security course' })
        d.assert_fact({ 'subject': 'Foundations of Computer Security co
urse' })
        d.assert_fact( {'subject': 'Applied Cryptography course' })
        d.assert_fact( {'subject': 'Adaptive Security course' })

  @when_all((m.field == 'Mobile Computing'))
  def mathc(d):
        d.assert_fact({ 'subject': 'Mobile Computing course' })
        d.assert_fact({ 'subject': 'Wireless Networks course' })
        d.assert_fact( {'subject': 'Distributed Systems course' })
        d.assert_fact( {'subject': 'Network Security course' })

  @when_all((m.field == 'Data Engineering'))
  def mathc(d):
        d.assert_fact({ 'subject': 'Data Mining course' })
        d.assert_fact({ 'subject': 'Database System Implementation cour
se' })
        d.assert_fact( {'subject': 'Information Retrieval course' })
        d.assert_fact( {'subject': 'Big Data Analytics course' })

  @when_all((m.field == 'VLSI & Embedded Systems'))
  def mathc(d):
        d.assert_fact({ 'subject': 'SOC Design & Test course' })
        d.assert_fact({ 'subject': 'Mixed Signal Design course' })
        d.assert_fact( {'subject': 'Computer Architecture course' })
        d.assert_fact( {'subject': 'VLSI Design Flow course' })

  @when_all((m.field == 'Machine Learning'))
  def mathc(d):
        d.assert_fact({ 'subject': 'Statistical Signal Processing cours
e' })
        d.assert_fact({ 'subject': 'Theories of Deep Learning course' }
)
        d.assert_fact( {'subject': 'CProbabilistic Graphical Models cou
rse' })
        d.assert_fact( {'subject': 'Data Sciences and Management in Pyt
hon course' })

  @when_all((m.field == 'Cyber-Physical Systems'))
  def mathc(d):
        d.assert_fact({ 'subject': 'Wireless Communications course' })
        d.assert_fact({ 'subject': 'Wireless system implementation course' })
        d.assert_fact( {'subject': '3GPP Standards for Wireless Networks course' })
        d.assert_fact( {'subject': 'Mobile Computing course' })

  @when_all((m.field == 'cb'))
  def mathc(d):
        d.assert_fact({ 'subject': 'Foundations of Modern Biology course' })
        d.assert_fact({ 'subject': 'Algorithms in Computational Biology course' })
        d.assert_fact( {'subject': 'Cell Biology and Biochemistry course' })
        d.assert_fact( {'subject': 'Introduction to Mathematical Biology course' })

  @when_all(+m.subject)
  def output(c):
        print('Fact: {0}'.format(c.m.subject))
```

Sample Outputs:



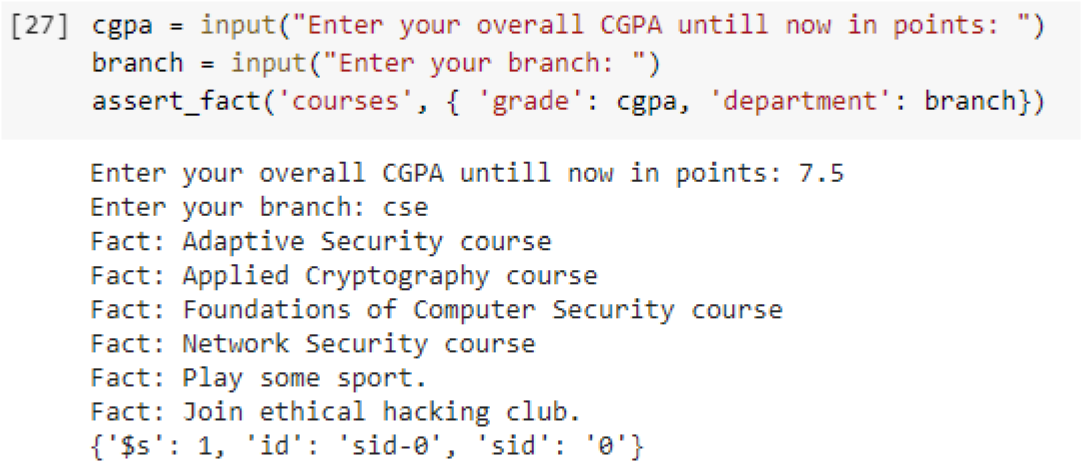
```
cgpa = input("Enter your overall CGPA untill now in points: ")
branch = input("Enter your branch: ")
assert_fact('courses', { 'grade': cgpa, 'department': branch})
```

```
Enter your overall CGPA untill now in points: 7.0
Enter your branch: cse
Fact: Network Security course
Fact: Distributed Systems course
Fact: Wireless Networks course
Fact: Mobile Computing course
Fact: Build a website or an app.
{'$s': 1, 'id': 'sid-0', 'sid': '0'}
```



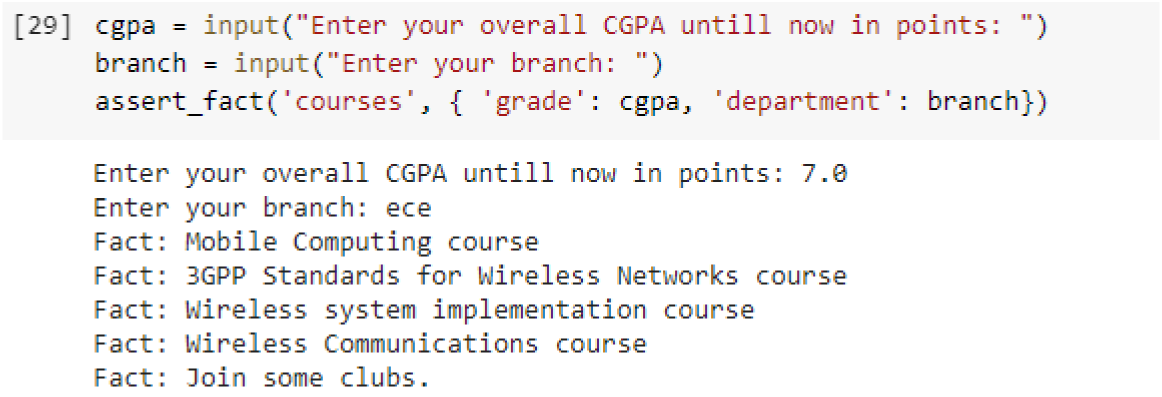
```
[24] cgpa = input("Enter your overall CGPA untill now in points: ")
     branch = input("Enter your branch: ")
     assert_fact('courses', { 'grade': cgpa, 'department': branch})
```

```
Enter your overall CGPA untill now in points: 7.0
Enter your branch: cb
Fact: Introduction to Mathematical Biology course
Fact: Cell Biology and Biochemistry course
Fact: Algorithms in Computational Biology course
Fact: Foundations of Modern Biology course
Fact: Read new developments in medical science.
Fact: Build a disease prediction model.
{'$s': 1, 'id': 'sid-0', 'sid': '0'}
```



```
[27] cgpa = input("Enter your overall CGPA untill now in points: ")
     branch = input("Enter your branch: ")
     assert_fact('courses', { 'grade': cgpa, 'department': branch})
```

```
Enter your overall CGPA untill now in points: 7.5
Enter your branch: cse
Fact: Adaptive Security course
Fact: Applied Cryptography course
Fact: Foundations of Computer Security course
Fact: Network Security course
Fact: Play some sport.
Fact: Join ethical hacking club.
{'$s': 1, 'id': 'sid-0', 'sid': '0'}
```



```
[29] cgpa = input("Enter your overall CGPA untill now in points: ")
     branch = input("Enter your branch: ")
     assert_fact('courses', { 'grade': cgpa, 'department': branch})
```

```
Enter your overall CGPA untill now in points: 7.0
Enter your branch: ece
Fact: Mobile Computing course
Fact: 3GPP Standards for Wireless Networks course
Fact: Wireless system implementation course
Fact: Wireless Communications course
Fact: Join some clubs.
```