REPORT MAY 22

Date:	22 MAY 2020	Name:	Rakshith B
Course:	TCSION CARRIER EDGE	USN:	4AL16EC409
Topic:	Understand Artificial Intelligence-Part-1, Part-2, Assessment	Semester & Section:	6th SEM B
Github Repository:	Rakshith-B		

FORENOON SESSION DETAILS

Image of session

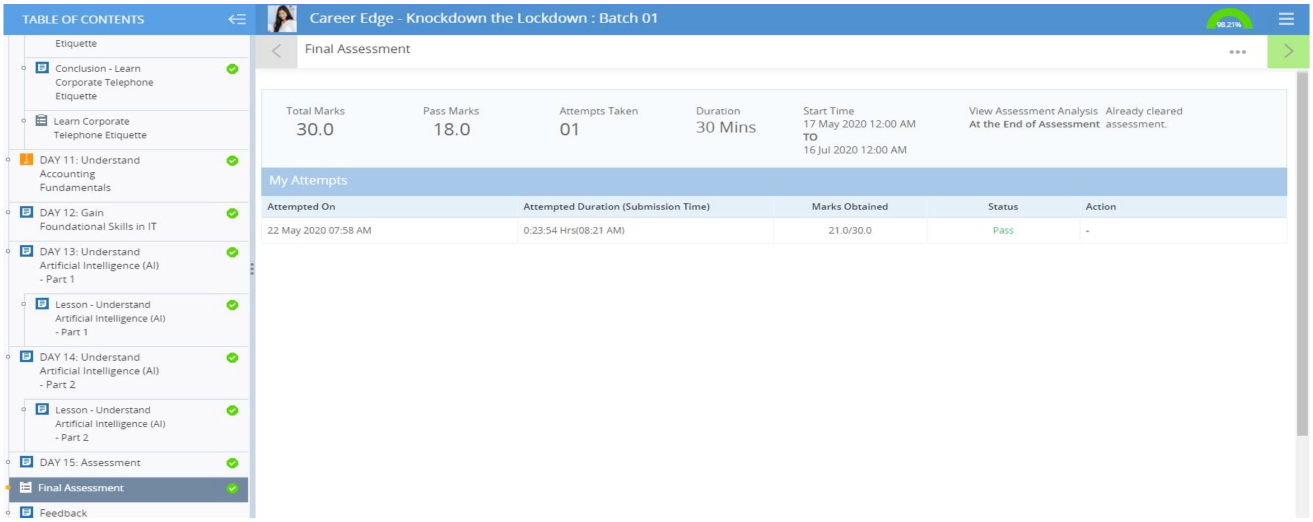


TABLE OF CONTENTS
Etiquette
Conclusion - Learn Corporate Telephone Etiquette
Learn Corporate Telephone Etiquette
DAY 11: Understand Accounting Fundamentals
DAY 12: Gain Foundational Skills in IT
DAY 13: Understand Artificial Intelligence (AI) - Part 1
Lesson - Understand Artificial Intelligence (AI) - Part 1
DAY 14: Understand Artificial Intelligence (AI) - Part 2
Lesson - Understand Artificial Intelligence (AI) - Part 2
DAY 15: Assessment
Final Assessment
Feedback
98.21%

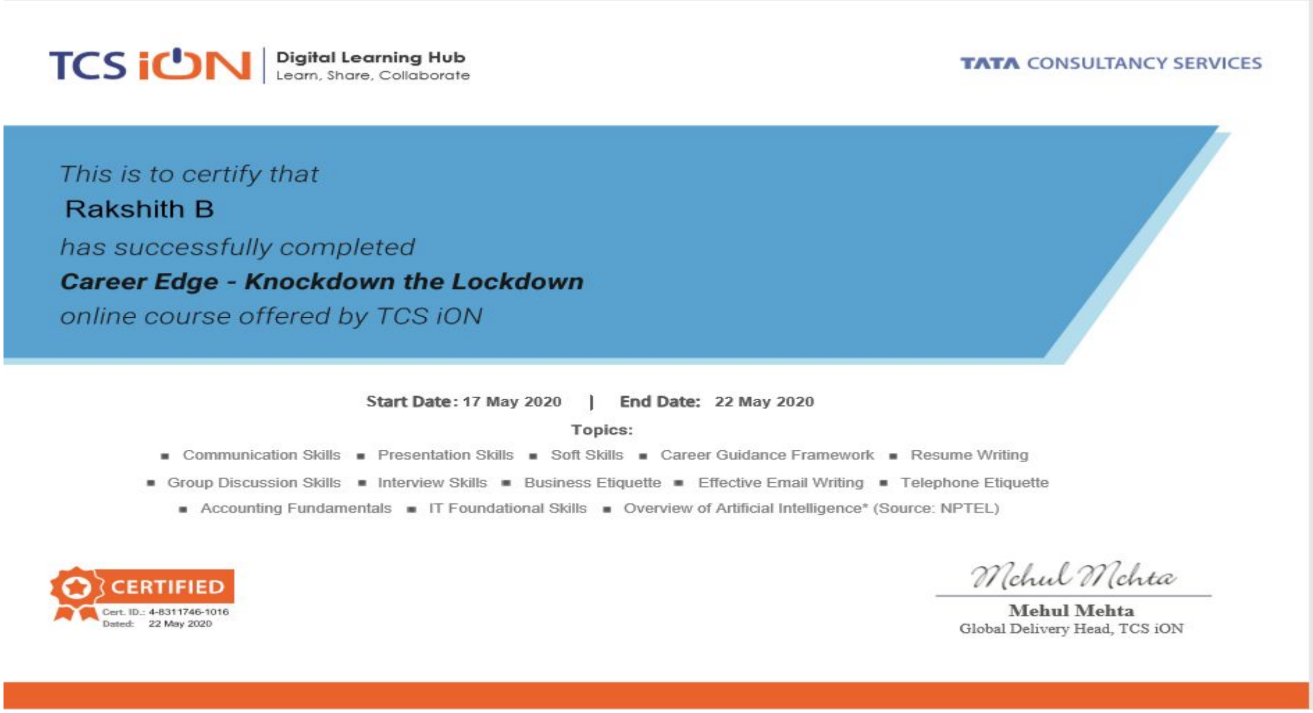
Career Edge - Knockdown the Lockdown : Batch 01

Final Assessment

Total Marks	Pass Marks	Attempts Taken	Duration	Start Time	View Assessment Analysis	Already cleared
30.0	18.0	01	30 Mins	17 May 2020 12:00 AM TO 16 Jul 2020 12:00 AM	At the End of Assessment	assessment.

My Attempts

Attempted On	Attempted Duration (Submission Time)	Marks Obtained	Status	Action
22 May 2020 07:58 AM	0:23:54 Hrs(08:21 AM)	21.0/30.0	Pass	-



TCS iON | Digital Learning Hub
Learn, Share, Collaborate

TATA CONSULTANCY SERVICES

This is to certify that

Rakshith B

has successfully completed

Career Edge - Knockdown the Lockdown

online course offered by TCS iON

Start Date: 17 May 2020 | **End Date:** 22 May 2020

Topics:

- Communication Skills
- Presentation Skills
- Soft Skills
- Career Guidance Framework
- Resume Writing
- Group Discussion Skills
- Interview Skills
- Business Etiquette
- Effective Email Writing
- Telephone Etiquette
- Accounting Fundamentals
- IT Foundational Skills
- Overview of Artificial Intelligence* (Source: NPTEL)

CERTIFIED
Cert. ID.: 4-8311746-1016
Dated: 22 May 2020

Mehul Mehta

Mehul Mehta
Global Delivery Head, TCS iON

Report –

Understanding Artificial Intelligence

Goals of Artificial Intelligence:

- **Introduce you to the Yield of AI**
- **To Explain the Challenges in Building on Intelligent System**
- **To Explain the**
  - ❑ **Key Paradigms**
  - ❑ **Core Techniques**
  - ❑ **Algorithms**
- **After this Course you will be able to**
  - ❑ **Formulate Problems as State Space Search,Problems and Efficiently Solve Them**
  - ❑ **Write Game Playing Programs**
  - ❑ **Use Machine Learning to Find Patterns to Data**
  - ❑ **Building Expert Systems**

Introduction :

- **Definition of AI**
- **Example Systems**
- **Approaches to AI**
- **Brief History**

What is AI ?

- **It is Concerned with the Design of Intelligence in an Artificial Device.**
- **Term Coined by Mc.Carthy in 1956**
- **Artificial Intelligence is Concerned with the Design of Intelligence in an Artificial Device**

What is Intelligence ?

- **Behave as Intelligently as Humans**
- **Behave in the Best Possible Manner**
- **Thinking**
- **Acting**

The Turing Test: Results

It is Interrogator Cannot Reliably Distinguish the Human from the Computer, Then the Computer does Possess Intelligence

Typical AI Problems :

- **Intelligent Entities need to be able to do both "Mundane "and" Expert" Tasks.**
  - ❑ **Planning Route,Activity**
  - ❑ **Recognizing People,Objects**
  - ❑ **Communicating**
  - ❑ **Navigating Around Obstacles on the Street**
- **Expert Tasks :**
  - ❑ **Medical Diagnosis**
  - ❑ **Mathematical Problem Solving**

Intelligence Behaviour

- Perception
- Reasoning
- learning
- Understanding Language
- Solving Problems

Applications:

- Computer Vision
- Image Recognition
- Robotics
- Language Processing
- Speech Processing

Internet Agents :

- Monitor user Tasks
- Seeks Needed Information
- Learn Which Information is Most Useful

AI Topics

- Core Areas
- Perception
- Uncertainty
- General Algorithms
- Applications
- Decision Theory Reasoning with Symbolic Data

Limits of AI Today

- Today Successful AI System
    - Operate in well-Defined Domains
    - Employ Narrow,Specialized Knowledge
- Commonsense Knowledge
    - Needed in Complex,Opens Ended Worlds
    - Understand Unconstrained Natural Language

What can AI Systems do ?

- Computer Vision :Face Recognition
- Robotics : Autonomous Automobile
- Natural Language Processing :Simple Machine Translation
- Expert Systems : Medical Diagnosis in a Narrow Domain
- Spoken Language : 1000 Words Continuous Speech
- Planning and Scheduling : Hubble Telescope Experiments
- Learning
- Gaming : Grand Master Level in Chess(World Champion),Checkers,etc..

What AI Cannot do ?

- **Understand Natural Language Robustly**
- **Read and Understand article in a Newspaper**
- **Surf the Web**
- **Learn a Natural Language**

Date: 22 MAY 2020

Course: PYTHON On Udemy

Topic: Create Web Maps with Python and Folium

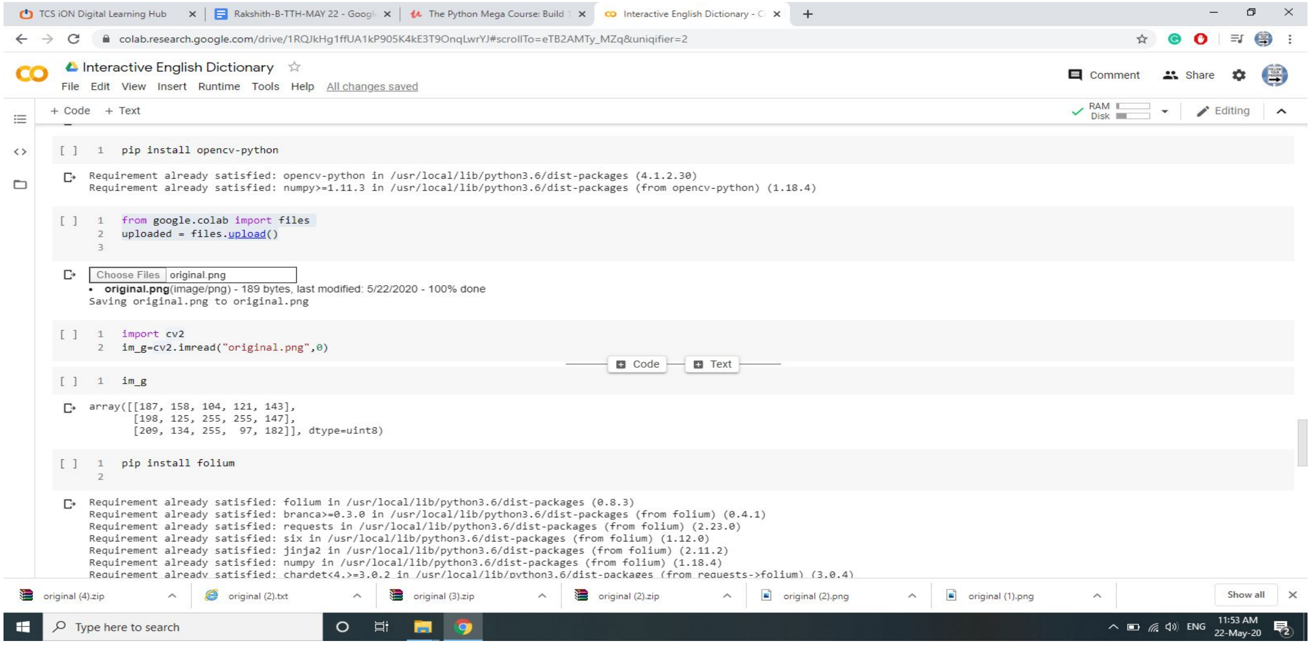
Name:RAKSHITH B

USN:4AL16EC409

Semester & Section:6 B

AFTERNOON SESSION DETAILS

Image of session



TCS iON Digital Learning Hub
Rakshith-B-TTH-MAY 22 - Googl
The Python Mega Course: Build
Interactive English Dictionary - C
colab.research.google.com/drive/1RQJkHg1ffUA1kP905K4kE3T9OnqLwrYJ#scrollTo=eTB2AMTy_MZq&uniqifier=2
Interactive English Dictionary
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
+ Code + Text
RAM Disk
Editing
Code Text
original (4).zip
original (2).txt
original (3).zip
original (2).zip
original (2).png
original (1).png
Show all
Type here to search
ENG
11:53 AM
22-May-20

```
[ ]  pip install opencv-python
```

```
Requirement already satisfied: opencv-python in /usr/local/lib/python3.6/dist-packages (4.1.2.30)
Requirement already satisfied: numpy>=1.11.3 in /usr/local/lib/python3.6/dist-packages (from opencv-python) (1.18.4)
```

```
[ ]  from google.colab import files
     uploaded = files.upload()

```

Choose Files original.png

- **original.png**(image/png) - 189 bytes, last modified: 5/22/2020 - 100% done

```
Saving original.png to original.png
```

```
[ ]  import cv2
     im_g=cv2.imread("original.png",0)
```

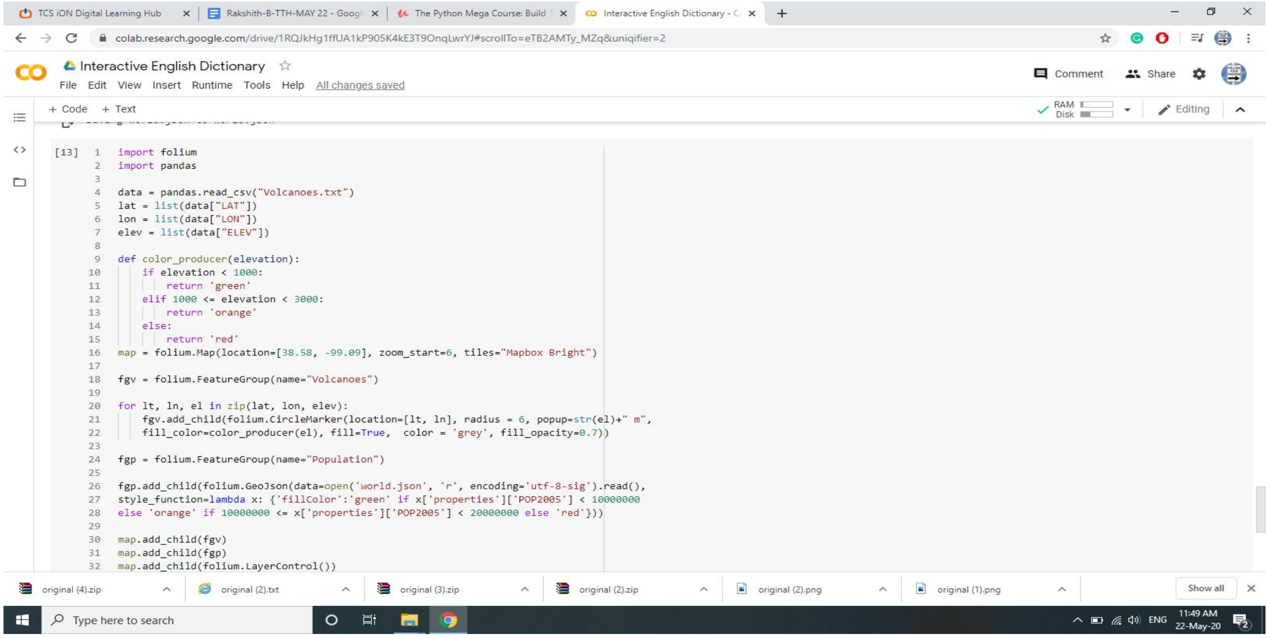
```
[ ]  im_g
```

```
array([[187, 158, 104, 121, 143],
       [198, 125, 255, 255, 147],
       [209, 134, 255,  97, 182]], dtype=uint8)
```

```
[ ]  pip install folium

```

```
Requirement already satisfied: folium in /usr/local/lib/python3.6/dist-packages (0.8.3)
Requirement already satisfied: branca>=0.3.0 in /usr/local/lib/python3.6/dist-packages (from folium) (0.4.1)
Requirement already satisfied: requests in /usr/local/lib/python3.6/dist-packages (from folium) (2.23.0)
Requirement already satisfied: six in /usr/local/lib/python3.6/dist-packages (from folium) (1.12.0)
Requirement already satisfied: jinja2 in /usr/local/lib/python3.6/dist-packages (from folium) (2.11.2)
Requirement already satisfied: numpy in /usr/local/lib/python3.6/dist-packages (from folium) (1.18.4)
Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.6/dist-packages (from requests->folium) (3.0.4)
```



TCS iON Digital Learning Hub
Rakshith-B-TTH-MAY 22 - Googl
The Python Mega Course: Build
Interactive English Dictionary - C
colab.research.google.com/drive/1RQJkHg1ffUA1kP905K4kE3T9OnqLwrYJ#scrollTo=eTB2AMTy_MZq&uniqifier=2
Interactive English Dictionary
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
+ Code + Text
RAM Disk
Editing
original (4).zip
original (2).txt
original (3).zip
original (2).zip
original (2).png
original (1).png
Show all
Type here to search
ENG
11:49 AM
22-May-20

```
[13]  import folium
      import pandas

      data = pandas.read_csv("Volcanoes.txt")
      lat = list(data["LAT"])
      lon = list(data["LON"])
      elev = list(data["ELEV"])

      def color_producer(elevation):
          if elevation < 1000:
              return 'green'
          elif 1000 <= elevation < 3000:
              return 'orange'
          else:
              return 'red'
      map = folium.Map(location=[38.58, -99.09], zoom_start=6, tiles="Mapbox Bright")

      fgv = folium.FeatureGroup(name="Volcanoes")

      for lt, ln, el in zip(lat, lon, elev):
          fgv.add_child(folium.CircleMarker(location=[lt, ln], radius = 6, popup=str(el)+" m",
          fill_color=color_producer(el), fill=True,  color = 'grey', fill_opacity=0.7))

      fgp = folium.FeatureGroup(name="Population")

      fgp.add_child(folium.GeoJson(data=open('world.json', 'r', encoding='utf-8-sig').read(),
      style_function=lambda x: {'fillColor':'green' if x['properties']['POP2005'] < 10000000
      else 'orange' if 10000000 <= x['properties']['POP2005'] < 20000000 else 'red'}))

      map.add_child(fgv)
      map.add_child(fgp)
      map.add_child(folium.LayerControl())
```

Report –

```
import folium
import pandas

data = pandas.read_csv("Volcanoes.txt")
lat = list(data["LAT"])
lon = list(data["LON"])
elev = list(data["ELEV"])

def color_producer(elevation):
    if elevation < 1000:
        return 'green'
    elif 1000 <= elevation < 3000:
        return 'orange'
    else:
        return 'red'
map = folium.Map(location=[38.58, -99.09], zoom_start=6, tiles="Mapbox Bright")

fgv = folium.FeatureGroup(name="Volcanoes")

for lt, ln, el in zip(lat, lon, elev):
    fgv.add_child(folium.CircleMarker(location=[lt, ln], radius = 6, popup=str(el)+" m",
    fill_color=color_producer(el), fill=True,  color = 'grey', fill_opacity=0.7))

fgp = folium.FeatureGroup(name="Population")

fgp.add_child(folium.GeoJson(data=open('world.json', 'r', encoding='utf-8-sig').read(),
style_function=lambda x: {'fillColor':'green' if x['properties']['POP2005'] < 10000000
else 'orange' if 10000000 <= x['properties']['POP2005'] < 20000000 else 'red'}))

map.add_child(fgv)
map.add_child(fgp)
map.add_child(folium.LayerControl())

map.save("Map1.html")
```