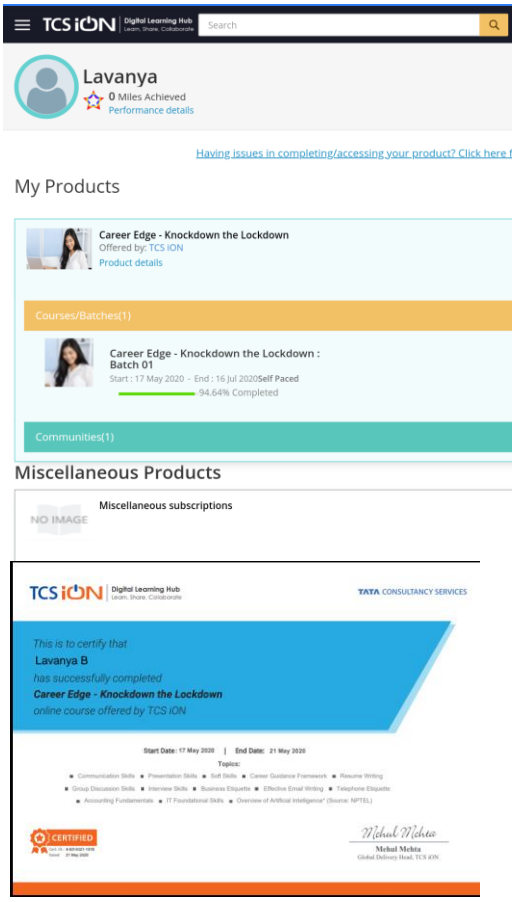
DAILY ASSESSMENT FORMAT

Date:	22/05/2020	Name:	Lavanya B
Course:	TCSION	USN:	4al17ec043
Topic:	Artificial intelligence 01 and 02	Semester & Section:	6th A
Github Repository:	Lavanya-B		

FORENOON SESSION DETAILS

Image of session



Report –

Artificial intelligence

Artificial Intelligence is a broad branch of computer science, aiming to create systems that can function intelligently and independently just like humans.

AI is an imitation of human intelligence processes by machines. The intelligence processes include learning, reasoning, and self-correction. Specific AI applications include machine vision, speech recognition, and expert systems.

machine learning

It uses statistical techniques to give machines the ability to learn from data without being explicitly given any instruction on how do it.

deep learning

It mimics the activity in the layers of neurons inthe brain to learn how to recognize complex patterns in data.

reinforcement learning

It is a software agents that learn goal-oriented behaviour by trial and error in an environment that provides rewards or penalties for achieving that goal.

transfer learning

It focuses on storing knowledge gained in one problem and applying it to a different or related problems,thereby reducing the need for additional training data and compute.

good old-fashionable ai(gofai)

A name given to an early symbolic ai paradigm that fell out of favour amongst researches. computer vision.

natural language.

speech processing.

predictive analytics.

multi-agent collaboration.

Date:	**22/05/2020**	**Name:**	**Lavanya B**
Course:	**Python**	**USN:**	**4al17ec043**
Topic:	**Application-02**	**Semester & Section:**	**6th A**

AFTERNOON SESSION DETAILS
Image of session

Lectures More

134	Color Points Video - 07:55 mins	
135	Add and Style Points (Practice) Article	
136	Tip: Add and Style Points Article	
137	Solution Video - 01:53 mins	
138	GeoJson Data Video - 05:34 mins - Resources (1)	
139	Adding a GeoJson Polygon Layer Video - 03:20 mins	
140	**Choropleth Map** Video - 09:34 mins	
141	Layer Control Panel Video - 06:23 mins	

Report –

Application 02
Create web map with python and folium

- **This folium shows how to create a leaflet web map from scratch with python and the folium library.**

- **That should generate a map.html file.later,you can simpply put that HTML file on a live server and have the map online.**

- **Folium has been able to generate HTML,javascript and css and thee three render the map o the browser.**

- **We can server this HTML as a static file in basic webserver,but you can also go more advanced and create a python application and have folium dynamically generate such leaflet maps on demand.**

The general approach we take here is to first create a folium basemap and then add two layers to it:

1. **A choropleth of census tracts,symbolised crime density**
2. **Crime point location.**

A separate function to plot each of these two layer,each of which takes a geodataframe as its input.

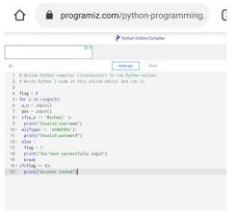
Folium takes unprojected lat/long coordinates for all of its plotting,so in make sure to concert all my projected geodata frames toWGS84 within the plotting functions.

Write python code to verify user_name = "Micheal" and password ="e3$WT89x". The total number of attempts are 03. For every wrong user_name and password Print - Invalid username or Password, upon three attempts fails print- Account locked

If inputs are correct Print - You have successfully login

```
flag = 0
for i in range(3):
 u_n = input()
 pas = input()
 if(u_n != 'Micheal' ):
```

```
  print("Invalid username")
 elif(pas != 'e3$WT89x'):
  print("Invalid password")
 else :
  flag = 1
  print("You have successfully login")
  break
if(flag == 0):
  print("Account locked")
```



Eg. for switch case

```
switch = lambda value,dict: exec(dict[value]) if value in dict else
exec(dict.get(None,"pass"))
case = lambda value:value
default = None

age = 70

switch(age,{
   case(16):"print('Too young')",
   case(42):"print('Adult')",
   case(70):"print('Senior')",
   default :"print('Invalid age')"
})
```


Code Playground
switch = lambda value,dict:
exec(dict[value]) if value in dict else
exec(dict.get(None,"pass"))
case = lambda value:value
default = None
age = 70
switch(age,{
case(16):"print('Too young')",
case(42):"print('Adult')",
case(70):"print('Senior')",
default :"print('Invalid age')"
})
OUTPUT
Senior