|  |  |
| --- | --- |
| Details | Coding Challenge! |
| Level | Software Engineer / Senior Software Engineer (Upto 6 Years of exp) |
| Candidate Name | Naveen |
| Email ID | Naveen151094@gmail.com |
| Biz2Credit Official Email | [codechallenge@biz2credit.com](mailto:codechallenge@biz2credit.com) |
| Process Owner | HR (Recruiters) |
| Technical Evaluation by | Technical Panel (Biz2Credit – Talent Acquisition Group) |

**Preface:** At onset, we would like to appreciate and thank you for completing the aptitude and technical (Mettl test) as part of recruitment process at Biz2Credit. As part of next steps of recruitment process, we would like to offer a coding challenge assignment as an integral part of our recruitment process along with few instructions and details on assignment (details appended below) for your reference & further action.

**General Instructions:**

* This coding challenge is specifically prepared to assess the competency skills for the Software Developers upto 6 years of experience by Biz2Credit technical team part of Talent Acquisition group
* This coding challenge will be offered to incumbents who have cleared the aptitude test and the technical (Mettl) test administered for the specific roles in Software Development.
* Incumbent need to complete the coding challenge within 3 days and to submit back on email ID [codechallenge@biz2credit.com](mailto:codechallenge@biz2credit.com) for the further evaluation
* Once submitted, candidate will get feedback within 2 business days from the HR team on the next steps (Either Selection or Rejection) as per the process.
* During completing this coding challenge or overall recruitment process. If any candidate need support, they can reach out to the HR Team members.

**Assignment – Coding Challenge:**

* We have some customer records in a text file (customers.txt) -- one customer per line, JSON lines formatted. We want to invite any customer within 100km of Dublin for some food and drinks on us. Write a program that will read the full list of customers and output the names and user ids of matching customers (within 100km), sorted by User ID (ascending).
* Don't forget, you'll need to convert degrees to radians.
* The GPS coordinates for Dublin area is 53.339428, -6.257664. We're looking for you to produce working code, with enough room to demonstrate how to structure components in a small program. Package the searching algorithm as an abstraction layer and installable component through npm package.
* Please refer to below attached file to understand the coding challenge and start working and submit at the earliest.



We appreciate your efforts and wish for your success in this assignment!

HR Team – Biz2Credit

**Solution:** (**File\_Name :** biz2credit\_code.py)

# Import inbuild module

import math

import json

import os

# Build a class object

class customerPortal(object):

# Assign a init constructor and pass it

def \_\_init\_\_(self):

pass

# Assign a public class for customers details

class \_customer\_details(object):

def \_\_init\_\_(self,name,user\_id,latitiude,longitude,distanceFromReference):

self.name = name

self.user\_id = user\_id

self.latitude = latitiude

self.longitude = longitude

self.distanceFromReference = distanceFromReference

# Change the value from degree to radaian

def \_degtorad(self,deg):

return deg\*math.pi/180

# Define the function to calculate distance in KMs from Longitute and Latitiute

def \_getDistanceBetweenTwoPoints(self,lat1,long1,lat2,long2):

#Radius of earth=6371 KM

radius\_of\_earth = 6371

difference\_lat = self.\_degtorad(lat2-lat1)

difference\_long = self.\_degtorad(long2-long1)

#using haversrine algorithm to calculate distannce

delta\_attitude=math.pow(math.sin(difference\_lat/2),2)+math.pow(math.sin(difference\_long),2) . \* math.cos(self.\_degtorad(lat1))\*math.cos(self.\_degtorad(lat2))

delta\_theeta = 2 \* math.atan2(math.sqrt(delta\_attitude),math.sqrt(1-delta\_attitude))

distance\_kms = radius\_of\_earth \* delta\_theeta

return distance\_kms

def getCustomersNearby(self,reference\_lat,reference\_long,threshold\_distance\_km,input\_file\_location,output\_file\_directory):

"""

The GPS coordinates for Dublin area is 53.339428, -6.257664

Longitude=53.339428

Latitiuate=-6.257664

Threshold Distance=100 km

"""

reference\_lat = 53.339428

reference\_long = -6.257664

threshold\_distance\_km = 100

# Read the input customers file from desktop

input\_file\_location ='C:\\Users\\Naveen Kumar\\Desktop\\Customers \_Assignment\_Coding Challenge (Upto 6 Years).txt'

# Write the customers output result file at desktop

output\_file\_directory = 'C:\\Users\\Naveen Kumar\\Desktop\\'

# Make an empty customer list

customer\_list = []

#Open the input file as json\_file and read it line by line

with open(input\_file\_location) as json\_file:

for line in json\_file:

line\_data = json.loads(line)

distanceFromReference = self.\_getDistanceBetweenTwoPoints(reference\_lat,reference\_long,float(line\_data['latitude']),float(line\_data['longitude']))

# Compare each line distance with threshold distance and then appends to customer\_list

if (distanceFromReference <= threshold\_distance\_km):

customer = self.\_customer\_details(line\_data['name'],line\_data['user\_id'],line\_data['latitude'],line\_data['longitude'],distanceFromReference)

customer\_list.append(customer)

# Save result file as "biz2credit\_results.txt" at output\_file\_directory location

output\_file\_path = output\_file\_directory + 'biz2credit\_results.txt'

# If same file name already exist, then automattically remove it and save as a new output file

if (os.path.exists(output\_file\_path)):

os.remove(output\_file\_path)

output\_data={}

# Sort the resultant distance based on user\_id and append to output\_data dictionary

for customer in customer\_list:

output\_data[customer.user\_id] = customer.name

output\_sorted\_keys = sorted(output\_data)

output\_print\_data = []

for key in output\_sorted\_keys:

temp = {}

temp["user\_id"]= key

temp["name"] = output\_data[key]

output\_print\_data.append(temp)

print(output\_print\_data)

# Write an output file as json format

with open(output\_file\_path,"w+") as outfile:

json.dump(output\_print\_data,outfile)

**NOTE:** This above file (biz2credit\_code.py) is called by another file (biz2credit\_call.py) as:

# Call customerPortal from biz2credit\_code module

from biz2credit\_code import customerPortal

customerPortal\_object=customerPortal()

customerPortal\_object.getCustomersNearby(53.339428,6.257664,100,'C:\\Users\\Naveen Kumar\\Desktop\\Customers \_Assignment\_Coding Challenge (Upto 6 Years).txt','C:\\Users\\Naveen Kumar\\Desktop\\')