Big data Management Analytics and Management

In this homework, you will learn how to solve problems using **Apache Spark**. Use Apache Spark to derive some statistics

The dataset files are as follows and columns are separate using '**::**'
business.csv.
review.csv.
user.csv.

Dataset Description.
The dataset comprises of **three** csv files, namely user.csv, business.csv and review.csv.

Business.csv file contain basic information about local businesses.
Business.csv file contains the following columns "business_id"::"full_address"::"categories"

'business_id': (a unique identifier for the business)
'full_address': (localized address),
'categories': [(localized category names)]

review.csv file contains the star rating given by a user to a business. Use user_id to associate this review with others by the same user. Use business_id to associate this review with others of the same business.

review.csv file contains the following columns "review_id"::"user_id"::"business_id"::"stars"
'review_id': (a unique identifier for the review)
'user_id': (the identifier of the reviewed business),
'business_id': (the identifier of the authoring user),
'stars': (star rating, integer 1-5),the rating given by the user to a business

user.csv file contains aggregate information about a single user across all of Yelp
user.csv file contains the following columns "user_id"::"name"::"url"
user_id': (unique user identifier),
'name': (first name, last initial, like 'Matt J.'), this column has been made anonymous to preserve privacy
'url': url of the user on yelp

After being familiar with the data - you are required to **write efficient Spark programs in Java/Scala/Python to find the following information. You can use spark-shell for scala, or pyspark for python if you are using Spark interactive mode.**

NB: :: is Column separator in the files.

Q1
List the 'user id' and 'rating' of users that reviewed businesses located in Stanford
Required files are 'business' and 'review'.

Sample output

User id	**Rating**
0WaCdhr3aXb0G0niwTMGTg	4.0

Q2:
List the business_id , full address and categories of the Top 10 businesses using the average ratings.

This will require you to use **review.csv** and **business.csv files.**

Sample output:

business id	**full address**	**categories**	**avg rating**
xdf12344444444,	CA 91711	List['Local Services', 'Carpet Cleaning']	5.0

Q3:
Given a file that contains weighted edge information of a directed graph. You have to write a Spak program to calculate the sum of weights of all incoming edges for each node in the graph.

Src	**tgt**	**weight**
A	**D**	**1**
A	**F**	**1**
A	**G**	**3**
B	**E**	**51**
B	**F**	**79**
C	**A**	**10**

Load the sample file in HDFS and load it from Spark.

You should output in the following format where nodes with only incoming edges will be visible.

Sample output:

A	**10**
D	**1**

E 51
F 80
G 3