Bonus Assignment #2

Due Date: 11/13/20 by 11:59pm

Deliverable:

- Use the object-oriented design principles and utilize the MVC architecture discussed in the class to produce an object-oriented web-based enterprise application that is reusable, flexible, and extensible.
- Use **Servlets** to implement the functionalities listed below.
- Record 10 minutes demo of your assignment's run using screencast. The tool can be downloaded from this URL http://screencast-o-matic.com/home
- Capture most important 10 screen-shots of your output and save them in a file called output.pdf
- All source code and byte code shall be submitted.
- Readme text file that illustrates how to compile/install/run your application
- Post your homework as a single zipped file on Blackboard with the name "Bonus_HW2_YourLastName,FirstName"

Important Notes:

- NO IDE to be used in any shape/form in the implementation of this assignment
- Do NOT communicate or share your assignment with others
- For this Assignment you must enter at least 100 transactions

High-Level Requirements:

Extend Assignment #4 to implement the following requirements.

The bonus assignment can be considered for grading ONLY if the functionalities listed in Assignment #4 are completely implemented.

Make sure you review the references and material for Neo4j tutorial before you work on this assignment.

Detailed Requirements:

- Add the following relationship types to the transactions (use random values generated)
    1. FRIENDS
        - Example: Customer c1 is friend of Customer c2
    2. SHARED
        - Example: Customer c1 shared product p5 Customer c3
    3. LIKES
        - Example: Customer c1 likes product p5
- Which product categories are often reviewed by the same customers? (review co-occurence graph : https://neo4j.com/docs/graph-algorithms/current/yelp-example/)
- Find similar groups of customers by projecting a review similarity network between them
- Use PageRank to present customers with recommendations of other products (https://neo4j.com/docs/graph-algorithms/current/algorithms/page-rank/)
- An influencer customer might write many reviews, likes/dislikes many products, has many friends, purchased many products but this may not always lead to an increased spread of liked and purchased products. A customer, who has high betweenness centrality, whose likes/reviews/friends, spread widely outside the immediate neighbors in the network, is generally considered an influential customer as well. Discuss and demonstrate how to find customers who have high betweenness centrality.
- Use PageRank to find potential influencers of the customers
- Use ArticleRank to find potential influencers of the customers (https://neo4j.com/docs/graph-algorithms/current/labs-algorithms/article-rank/)
- Based on the results you produce above, which one is better to find potential influencers? PageRank or ArticleRank?