Middle East Technical University
Department of Computer Engineering

CENG 495
Cloud Computing
Spring 2018-2019 Homework 2

Due Date: 19.04.2019, 23:55

This homework aims to get you familiar with the NoSQL databases and Database as a Service (DBaaS) platforms. MongoDB is classified as a NoSQL database program, which is open-source. As a document store database, it uses JSON like queries to handle the database operations. Atlas is the cloud platform of MongoDB. You are going to develop and deploy a simple eBay clone using MongoDB database on Atlas.

Keywords: PaaS, DBaaS, BaaS, MongoDB, JSON, Atlas, Stitch, Azure, Cloud Computing, eBay

1. MongoDB Atlas

- Create an Atlas account as a free user. (mongodb.com/cloud/Atlas)
- Create a database with MongoDB and deploy it into Atlas.
- Use the Backend as a Service Platform (BaaS) of Atlas (Stitch) to communicate with your database through an html code. Alternatively, you can use Azure to reach your MongoDB database.

2. eBay Clone

- You will implement an html code that communicates with your MongoDB on Atlas through Stitch. Your database will be tested through this code. Or you can just deploy an application on Microsoft Azure that communicates with MongoDB.
- The homepage of your code should have these options: **Add User**, **Delete User**, **Login as a User** which navigates through the homepage of the selected user as him or her. Note that you do not have to implement passwords or any kind of protection.
- User homepage shows 5 different kinds of elements: **User Name**, **wallet**, **rating**, **store**, **order history**. Wallet is the active balance of the user in your application.

Rating is the average point that user got from his previous sales (between 1 and 5). Store shows the products that the user put on sale (no need to delete after being sold out). Order history shows the products the user bought before.

- Product page shows 6 main elements: **Product Name**, **photo**, **price**, **quantity, sellers name, sellers rating**. It is necessary for the photo to be visually seen on the product page. (Note that there is no need to really upload the photos, you can just use links)
- A user can apply these 6 operations: **Deposit** money on the wallet, **withdraw** money from the wallet, **buy** product(s) (the product quantity should be sufficient, a user can buy more than one at once, both sellers and users wallets should be updated after trade, buyers wallet should also be checked to see if the purchase is possible), **sell** product(s) (more than one product can be put on sale, no need to check if there is a product with the same name from same user, update wallet if another user buys the product), **rate** the user after purchase (user cannot rate the same user for the same product more than once), **drop** to stop selling a product (ratings from the previous sales of this product, and all history should remain, remove it only from the store).
- You can use just one page with different elements or different pages as you wish. However, these elements and operations should work exactly as described.
- Before submission, your database should have at least 10 different users, and at least 2 of them having more than one product on sale.

3. Useful Links

- mongodb.com/cloud/Atlas
- mongodb.com/cloud/stitch
- docs.mongodb.com/
- tutorialspoint.com/mongodb/

4. Submission

- In this assignment, you are expected to submit your html code(s) to ODTÜClass. The main file should be named as “index.html”. For submission on ODTÜClass, a tar.gz archive file (named hw2.tar.gz) that contains all your source code files and a README file that includes the design choices you made and a user’s guide. If you want to use Microsoft Azure instead of html code, just include your Azure source codes and give your app link on the README file.

- Do not forget to change the privacy settings since your database will be reached from an IP address other than yours for grading.
- The work you submit should be implemented by only you and genuine. However, you can use external libraries for graphical user interface if any. If you do so, you need to state your references for these codes in your README file.
- We have zero tolerance policy for cheating. There is no teaming up! People involved in cheating will be punished according to the university regulations and will get 0. You can discuss design choices or language preferences, but sharing code between each other or submitting third party code as a whole is strictly forbidden. In case a match is found, this will be considered as cheating.