Middle East Technical University

Department of Computer Engineering

CENG 495
Cloud Computing
Spring 2022–2023
HW - 1

Due date: 2023-04-16 23:59

1 Introduction

For this homework, you will develop an application and deploy it onto the cloud Platform-as-a-Service (PaaS) Render with NoSQL database MongoDB on the Database-as-a-Service (DBaaS) cloud platform MongoDB Atlas. The application will be an e-commerce site that allows regular users to *browse* items for sale with an additional administrator privileges and functionalities that allow privileged users to create, edit and delete items presented on the site.

2 MongoDB Platform: Atlas

- Go over NoSQL and MongoDB concepts. Make sure you understand how dynamic schema differs from what you are used to from relational databases
- Create a free account on MongoDB Atlas `https://www.mongodb.com/atlas/database`
- Create a new project and deploy a new database, you can follow their documentation
- Since you cannot control the IPv4 address of your Render deployment, configure your database to be accessible from any IP address, and setup password authorization for DB access
- Atlas allows fine-grained database access controls under Security section, you can leverage this for your admin user and regular users

3 PaaS Platform: Render

- Sign up to Render or use your existing account.
- Go over the documentation, select the programming language you will use to implement the homework from any language supported by Render
- While reading the documentation, you might see that Render supports MongoDB as well, we are using Atlas instead because deploying MongoDB on Render cannot be done with their free tier
- *Also* while reading the documentation, you might realize that Render expects a GitHub repo to pull and deploy your application from. You can use a private repo until the deadline (and the late submission days), after which you can make your homework public if you'd like

4 E-Commerce Application

When it comes to e-commerce, I'm sure there are some websites that come to your mind, ones that maybe you use regularly to buy groceries, clothing or appliances. Still, if you would like to draw inspiration for your design, sites that use the Shopify platform are common. If you would like to draw architectural inspirations instead, this example can be a good start. In any case, the functionalities we expect from your application are *much* less compared to real e-commerce websites or the example provided.

You are expected to develop an e-commerce application that has the following pages and functionalities;

4.1 Home Page

This is the start page of your application, it is accessed through `index.html`. The items available for sale should be listed on the homepage with filtering options for the category of the items. For the purposes of this homework, *at minimum*, the following categories of items should be for sale;

- Clothing
- Computer Components
- Monitors
- Snacks

If the current user is not authorized, they are only allowed to browse the items. You should offer a way for user authentication on your home page. You are not required but free to use passwords for regular users, or they can authenticate by entering their username. However, the admin user should have proper authentication.

4.2 Admin Capabilities

When the current user authenticates themselves as the admin user, they should be able to do the following;

Add Item Create & insert a new item into the database, see 4.4 for the required attributes of items.

Remove Item Remove the item from the application, deleting ratings and reviews of the item as well. Any user effected by this removal should have their relevant fields updated as well.

Add User Create a new user, refer to 4.5 for user attributes.

Remove User Remove the user and update relevant items that are effected by this change.

4.3 Regular User Capabilities

When a user logs in, they should be able to do the following on each item's page;

Rate Item Rates an item on a scale of 1 to 5. The average rating of the item is updated as a result. A user can rate an item multiple times, the latter overwriting the prior rating.

Review Item Allows a user to review the item. Reviews can be updated, and the newer review overwrites the older one.

4.4 Item Attributes

The following is the list of attributes for items. When implementing the item details view (either a new page or a frame), ensure that every relevant detail of the item is presented to the user. Note that not every field is required for every item.

Name The name of the item

Description The description of the item

Price The price of the item, choice of currency is left to you

Seller The seller of the item

Image The image showing the item. You do not need to implement image upload, a hyperlink to an image file on the Internet is fine

Size The size of the item, only relevant if the item is an article of clothing

Colour The colour of the item, only relevant if the item is an article of clothing

Spec The amount of RAM the item has or the dimensions of the screen, only relevant for computer components and monitors

Rating Average rating this item has received

Reviews List of the reviews on the item

You are free to add additional attributes as you see fit. For instance, to keep track of the average rating, you can also have a **number of reviewers**.

4.5 User Attributes

Create a dedicated page presenting the following details of the currently authenticated regular user. If the current user is not logged-in, they should not be able to see this page or the way of accessing this page.

Username The username of the user

Average Rating The average of every rating given by this user

Reviews Every review written by this user

5 Before Submission

Populate your application by adding at least 8 items and 3 users. Have your users rate and review every item. Make sure that at least one item from every possible category is present in the application. You are free & encouraged to use frameworks and libraries while building your application. However, the overall project should be your own work.

Bear in mind that we are using NoSQL based MongoDB which uses dynamic schema. You are expected to use the flexibility it provides and, for instance, design your database with the fewest collections possible.

6 Submission

- Deploy your application to Render and submit the source code of your application to our ODTUClass page.
- Archive your project as a `.tar.gz` file and name it as "firstname_lastname.tar.gz". Your submission must include a `README` file that includes your design decisions and the URL of your Render deployment. Without a publicly accessible deployment URL I cannot test & grade your submission.
- Your README can include: how to login as a regular user and the admin user, why you chose the programming language you did, which frameworks you have chosen and why, a user guide about how to use your application, as well as other points you would like to mention.
- This is an individual assignment. You can discuss your ideas with your peers but using implementation specific code that is not your own is strictly forbidden and constitutes as cheating. This includes but not limited to material from other students taking this course, previous homework, large language models or the Internet in general. The violators will get no grade from this assignment and will be punished according to the department regulations.

7 Grading

- 50% of your grade is reserved for the correctness of your MongoDB usage
- 40% will be graded on the usability of your web application and the implementation of the functionalities
- Your `README` file will make up the 10% of your grade.

Black-box testing will not (cannot) be employed for this homework, so I will use each application and read every submission in its entirety for grading.