CS 306 PROJECT PHASE 4

İlke Kanıl 29271

Sinem Gül Kaya 30298

Göksu Gültekin 30301

In Project Phase 4, we have developed an advanced application interfacing with MongoDB Atlas, specifically tailored for managing and analyzing customer feedback through complaints and item comments. This application offers a range of functionalities including the creation and meticulous management of various collections, sophisticated mechanisms for reading and selectively filtering data, and comprehensive operations such as create, read, update, delete. These functions are dynamically driven by user inputs, fostering a highly interactive database environment. The design prioritizes user centricity, ensuring efficient data management and elevating customer engagement. The application excels in handling structured feedback, thereby streamlining the feedback management process and offering insightful data analytics to better understand customer needs and preferences. This enhanced functionality makes the system not only a tool for data management but also a valuable for strategic customer relationship management.

We have two collections which are customer\_complaints and item\_comments.

Firstly we start with customer\_complaints.

At the beginning, our mongodb Atlas looks like this. There is no any collection.

A screenshot of a computer

Description automatically generated

1-First step, we create a new collection called customer\_complaints with its dummy\_data that we created explicitly in our code.

A screenshot of a computer

Description automatically generated

After the process finished, our collection looks like this:

A screenshot of a computer

Description automatically generated

A close-up of a white background

Description automatically generated

2-In the second part, we “read all data in a collection”.

After we entered the name of the collection we want to read, the data in the collection is read and shown in the terminal:

A black screen with white text

Description automatically generated

3- In this step, we “read some part of the data while filtering”.

First we enter the name of the collection that we want to filter and read from. Then we enter the field and value information. After that the data in the collection which have the related value on that field is read and shown in the terminal:

A black screen with white text

Description automatically generated

4-Now, we insert a data into our collection. After we select collection and enter the values, our new data is added into our current collection.

A screenshot of a computer

Description automatically generated

Our mongoDB Atlas is updated:

A screenshot of a computer

Description automatically generated

A screenshot of a computer code

Description automatically generated

5- Our next step is deleting an item. We select the collection that we want to make changes on, and enter the objectID of the item we want to delete:

A screenshot of a computer

Description automatically generated

As we can see, we deleted the first item in the collection which has the id entered. The number of items is decrement.

A screenshot of a computer

Description automatically generated

A white background with red and green text

Description automatically generated

6- Lastly, we update the data. We enter the collection name, the field we want to update and the value will be used in updating.

A screenshot of a computer

Description automatically generated

Our data is updated:

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Item\_comments;

In the second step we will create a second collection named item\_comments because we have two different data in the dummy data. We also upload the data in item\_comments :

A screenshot of a computer

Description automatically generated

Here is the MongoDb;

A screenshot of a computer

Description automatically generated

A screen shot of a computer

Description automatically generated

Then, we will choose option 2 and read the all data in the item\_comments, and print them in the main.py;

A black screen with white text

Description automatically generated

After that, we will choose 3 option and filter the item\_type (Laptop in this case) and print the other objects in the selected index;

A black screen with white text

Description automatically generated

Then, we will choose option 4 to insert data in the MongoDb, and select item\_comments:

A screenshot of a computer

Description automatically generated

We can see the result in the MongoDb that query results increment by 1 and the inserted document will start with headphone;

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

After that, we will choose the option 5 to delete the data in the collection item\_comments. We will delete by id ( the deleted element is “TV”, item\_name is “LG”);

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Finally we will update the data, with selecting option 6. We will choose the updated element by id then decide which part we want to update and finally terminating the system with selecting “no” for the question “Would you like to do something else?”.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A white background with green text

Description automatically generated

In conclusion, this project represents a solution for managing customer interactions and feedback. It leverages the flexibility and scalability of MongoDB Atlas, making it ideal for handling large volumes of unstructured data. The application's well structured user interface and backend functionalities demonstrate a comprehensive approach to database management, ensuring both efficiency and ease of use. This system is poised to significantly enhance user experience and streamline feedback processing for businesses or service providers.