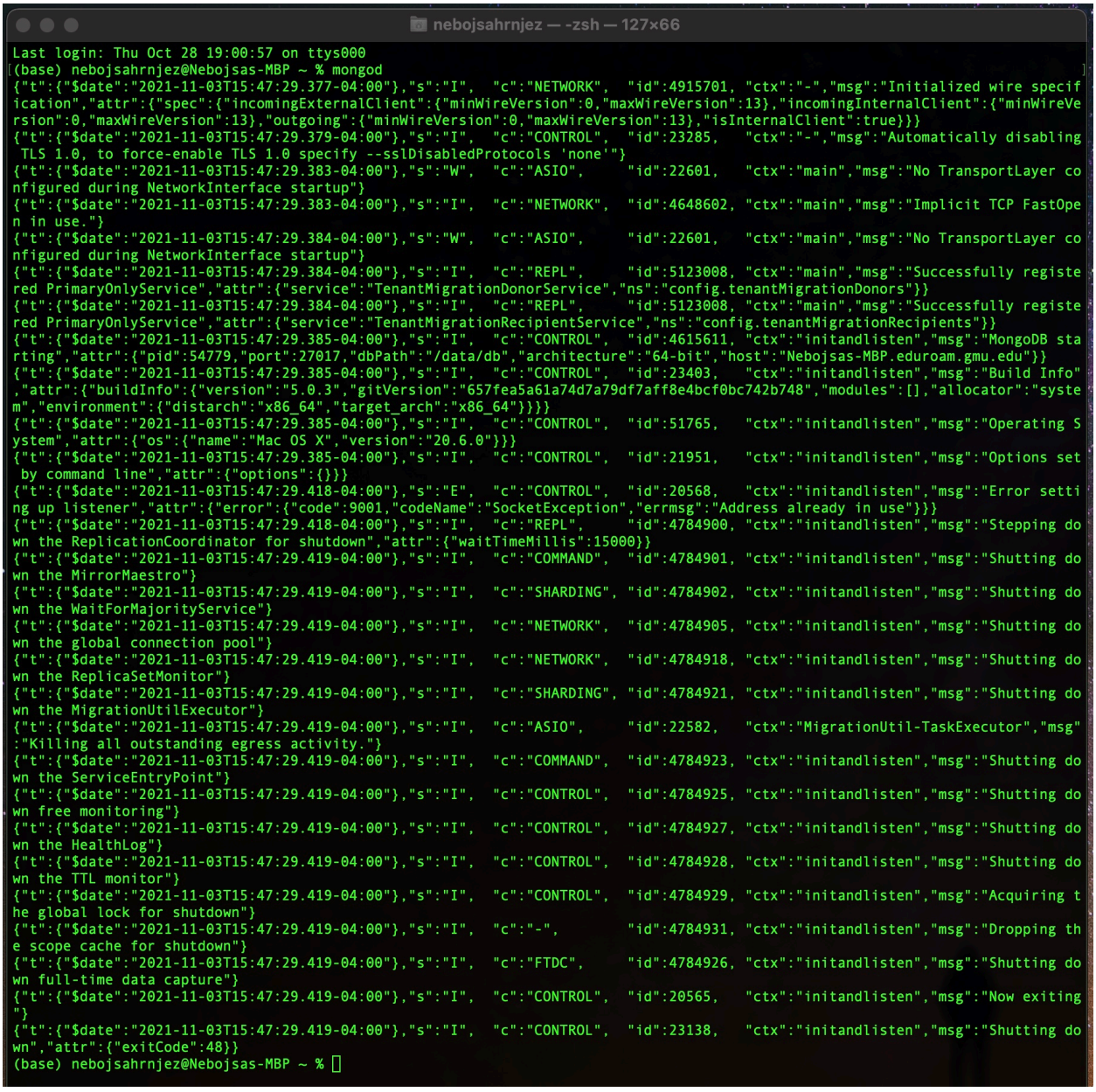
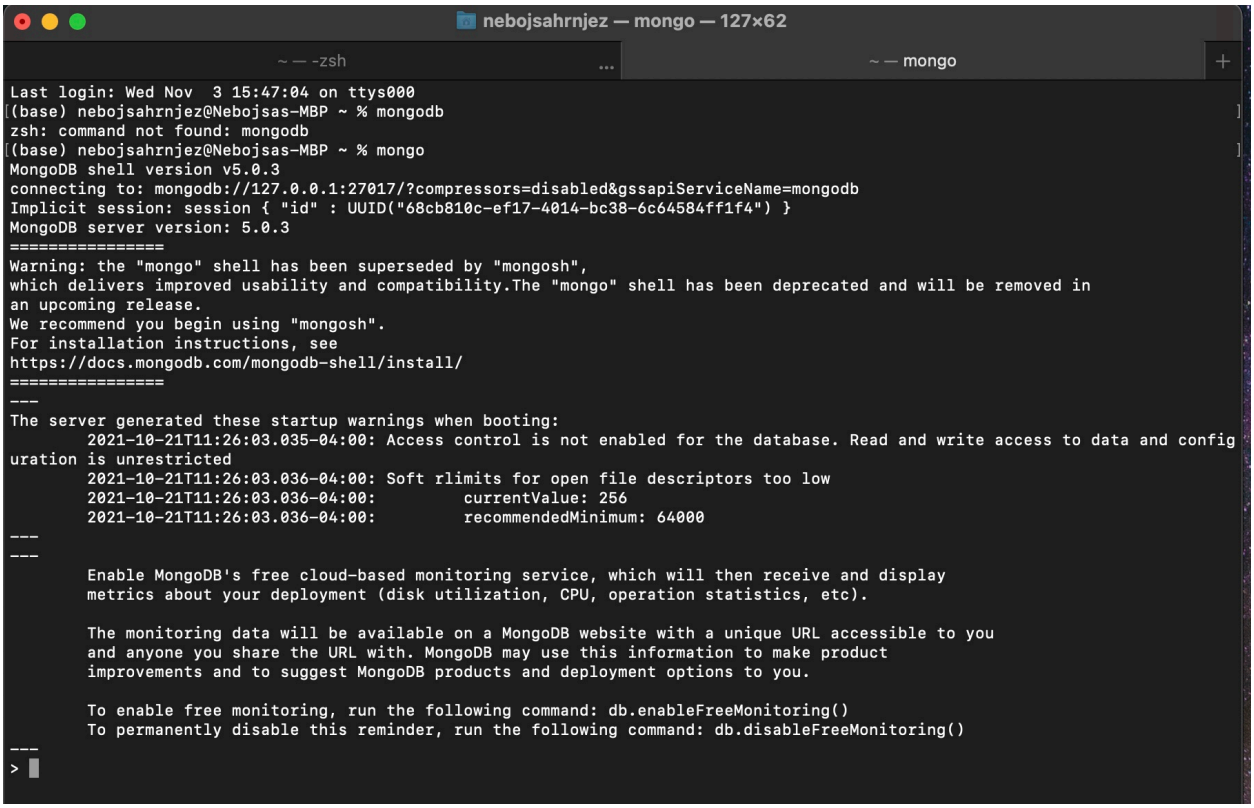
CS504 Project 2 Screenshots

Nebojsa Hrnjez
G01337837

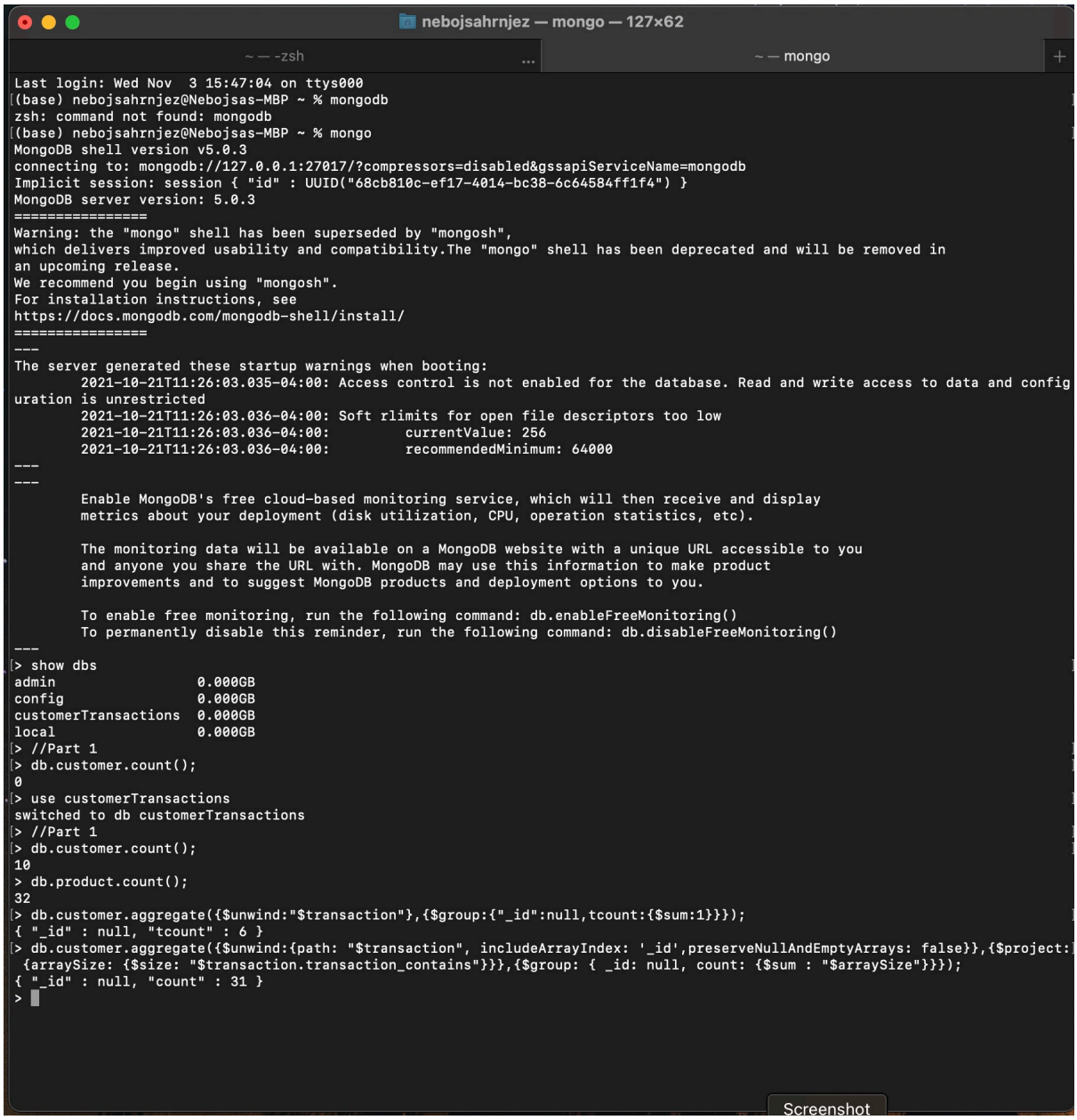


Opened terminal and typed the command “mongod”, this runs the background processes for “mongoDB”



Opened a new terminal window and called the “mongo” command.

Used the “show dbs” command to see what databases were present



Typed the command “use customerTransactions” to select the project 2 database.

Proceeded to complete part 1 of project 2:

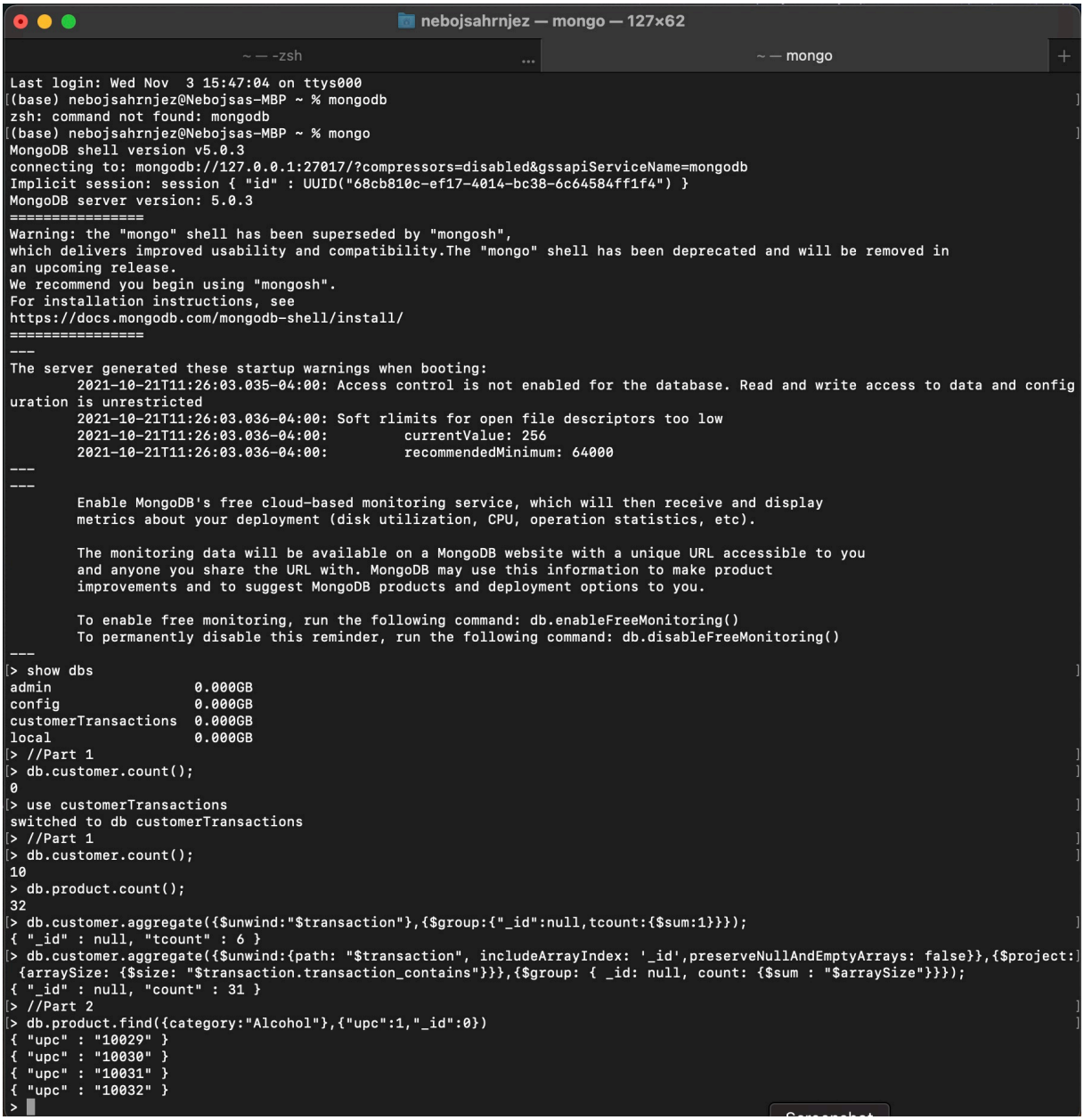
Typed the command “db.customer.count();” , this returns the count of documents in the customer collection

Typed the command “db.product.count();” , this returns the count of documents in the product collection

Typed the command “db.customer.aggregate({$unwind:”$transaction”},{$group: {“_id”:null,tcount:{$sum:1}}});" , this returns the count of “transactions” embedded in the documents in the customer collection

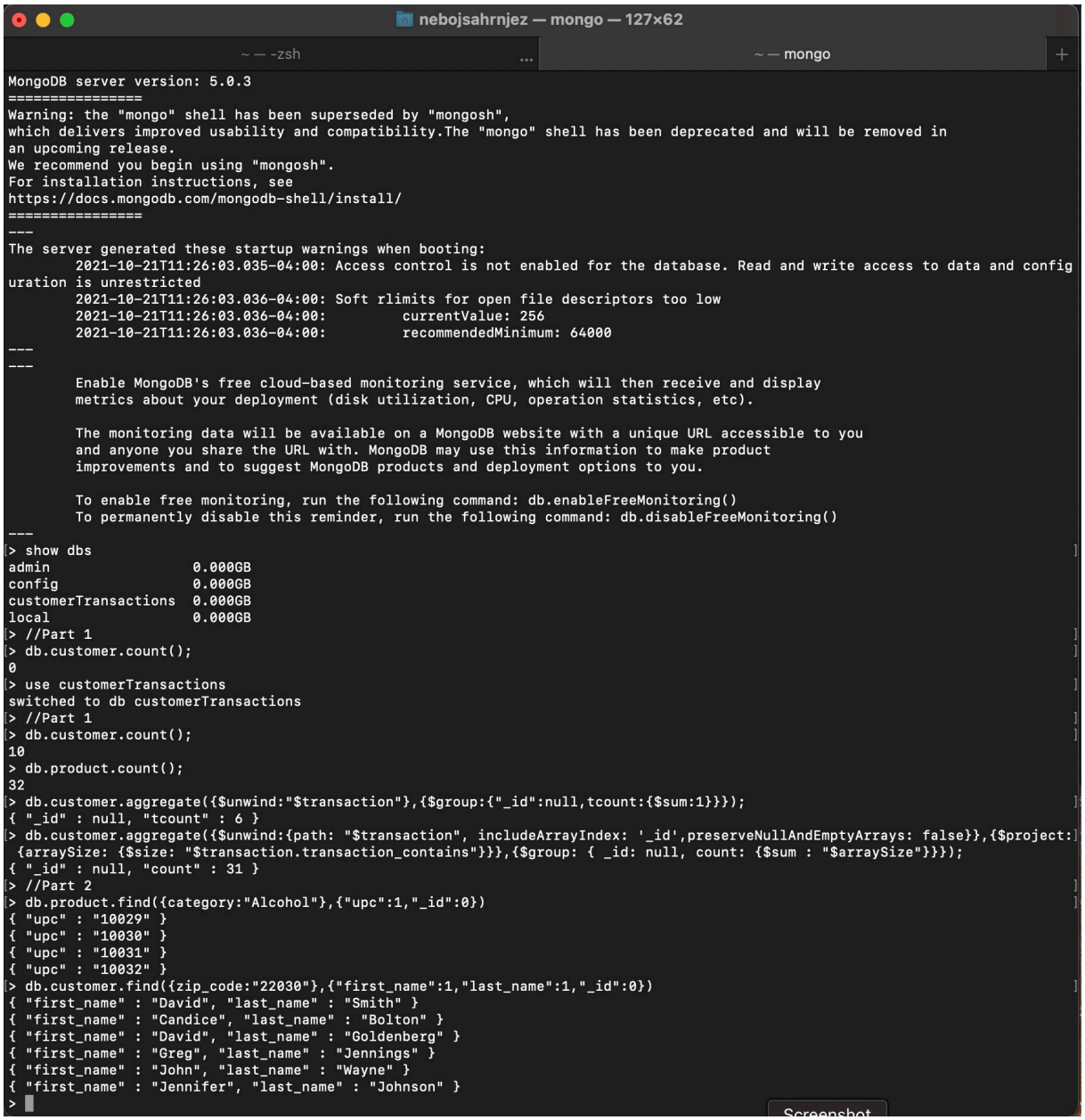
Typed the command “db.customer.aggregate({$unwind:{path: "$transaction", includeArrayIndex: '_id',preserveNullAndEmptyArrays: false}},{$project: {arraySize: {$size: "$transaction.transaction_contains"}}},{$group: { _id: null, count: {$sum : “$arraySize”}}})" , this

returns the count of “transaction_contains” that is embedded in “transactions” that is embedded in the documents in the customer collection

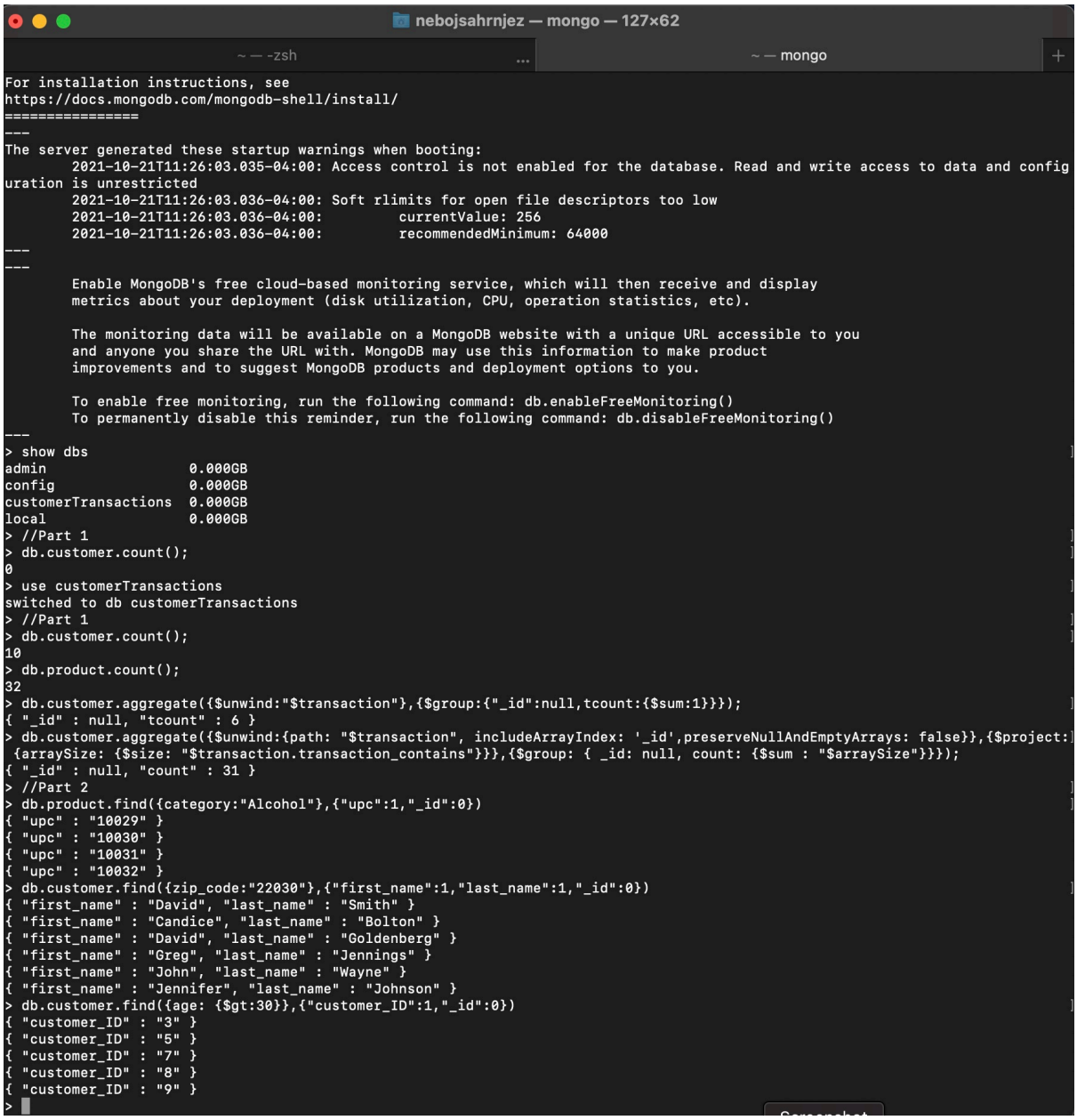


With part 1 complete, I proceeded on part 2 of project 2:

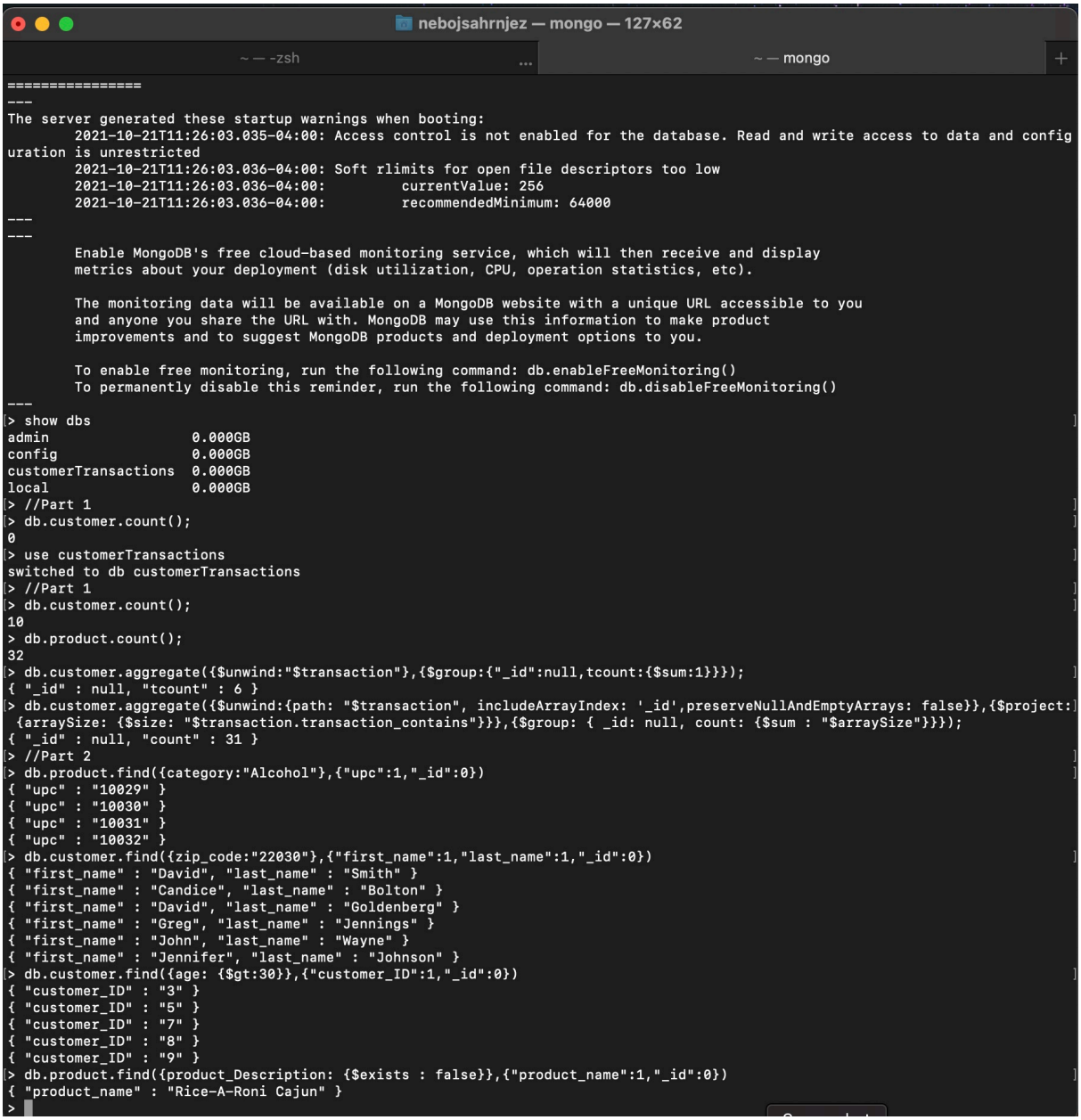
Typed the command “db.product.find({category:”Alcohol”},{"upc":1,"_id":0})" , This finds all the documents in the product collection with the “alcohol” category, I then restricted the return to only show the “upc”



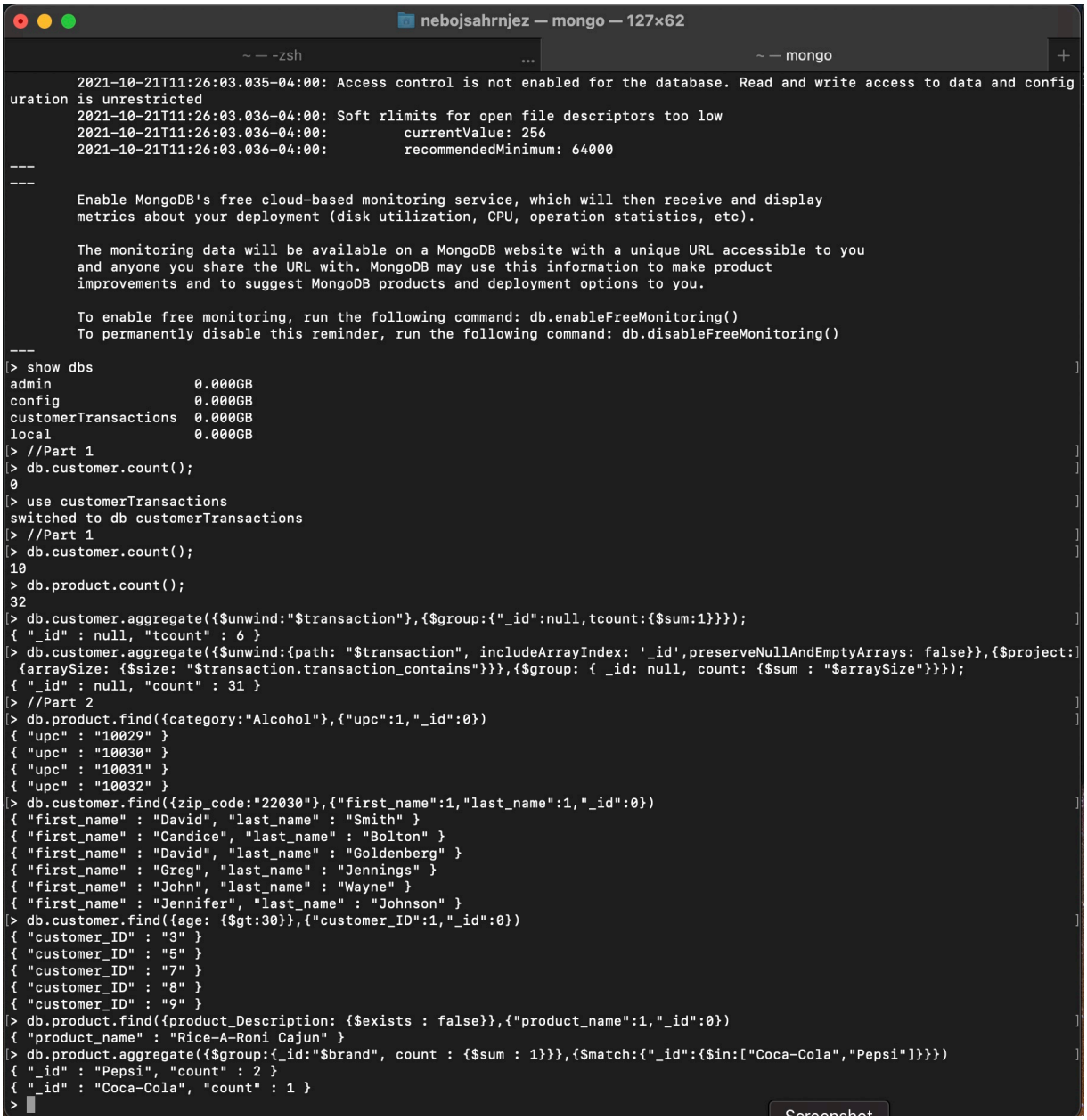
Typed the command “db.customer.find({zip_code:”22030”}, {“first_name":1,"last_name":1,"_id":0})" , This returns all the documents in the customer collection with the zip code “22030” (the zip code for fairfax), I then restricted the return so it only shows the first and last name.



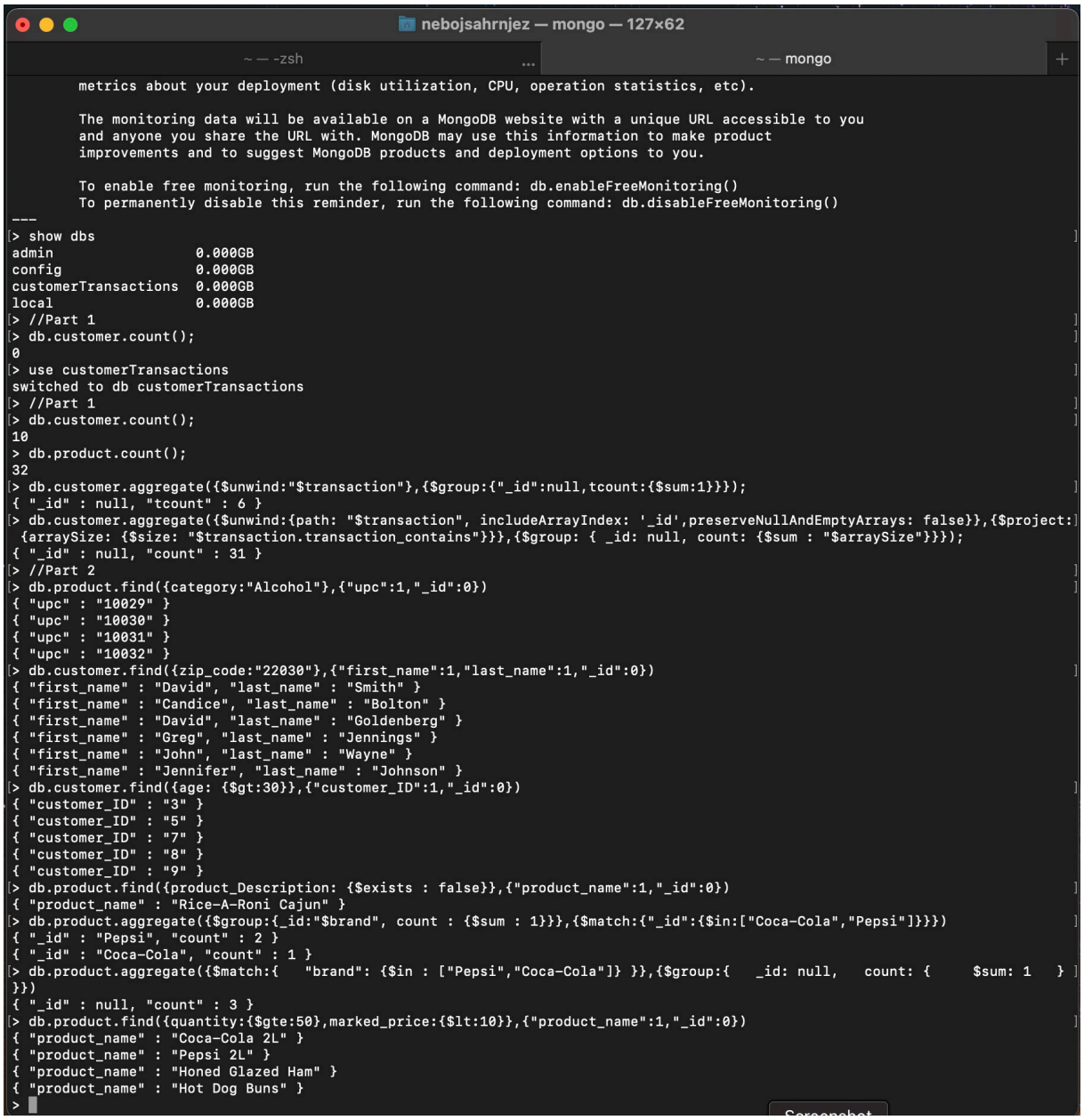
Typed the command “db.customer.find({age: {$gt:30}},{“customer_ID”:1,"_id":0})" , this returns all the documents in the customer collections with an age greater than 30. I then restricted theft to only show “customer_ID”



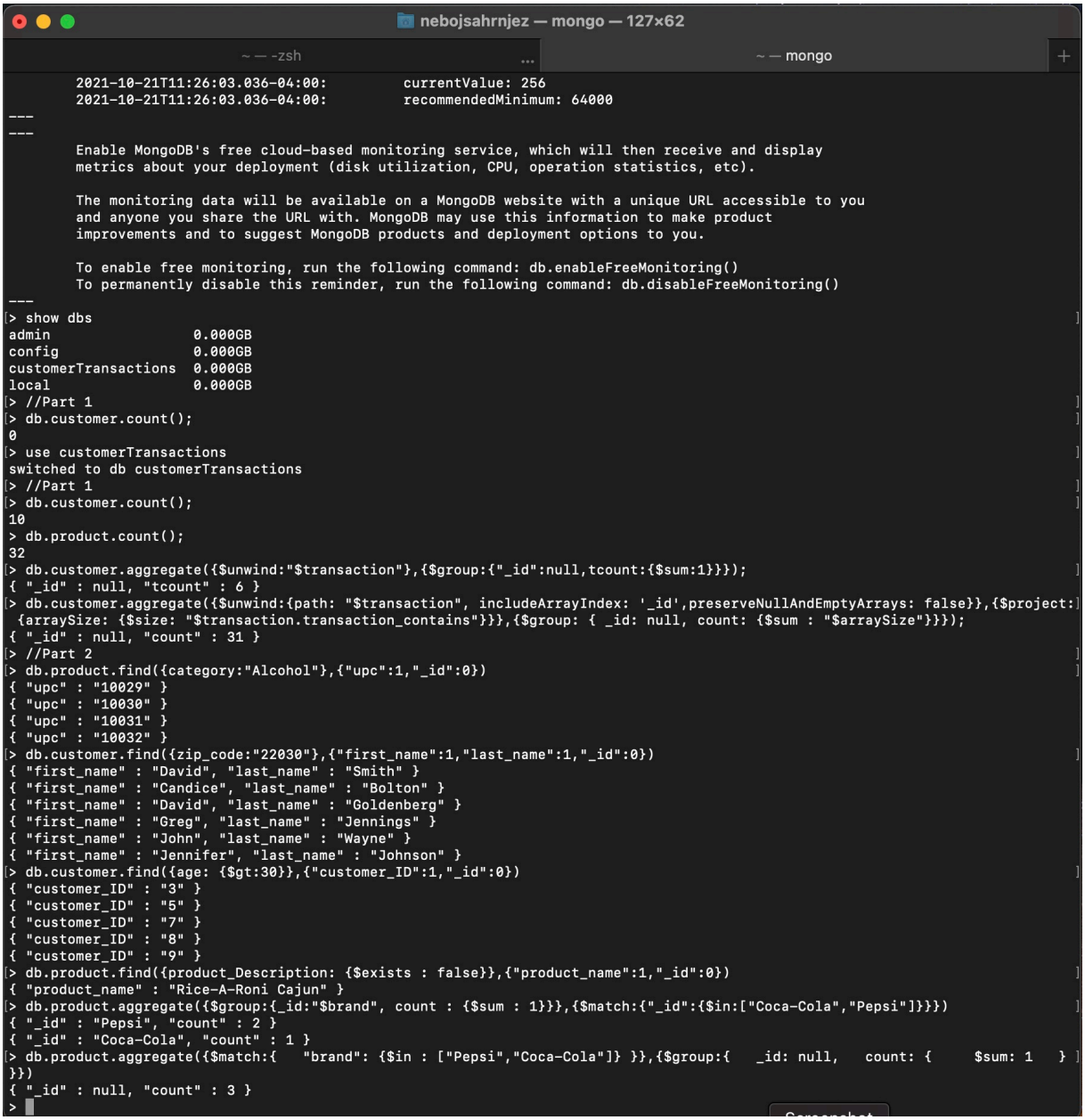
Typed the command “db.product.find({product_Description: {$exists : false}}, {“product_name”:1,"_id":0})" , this returns all the documents from the product collection where “product_Description” exists evaluates to false. I then restricted the output to only showing the “product_name”



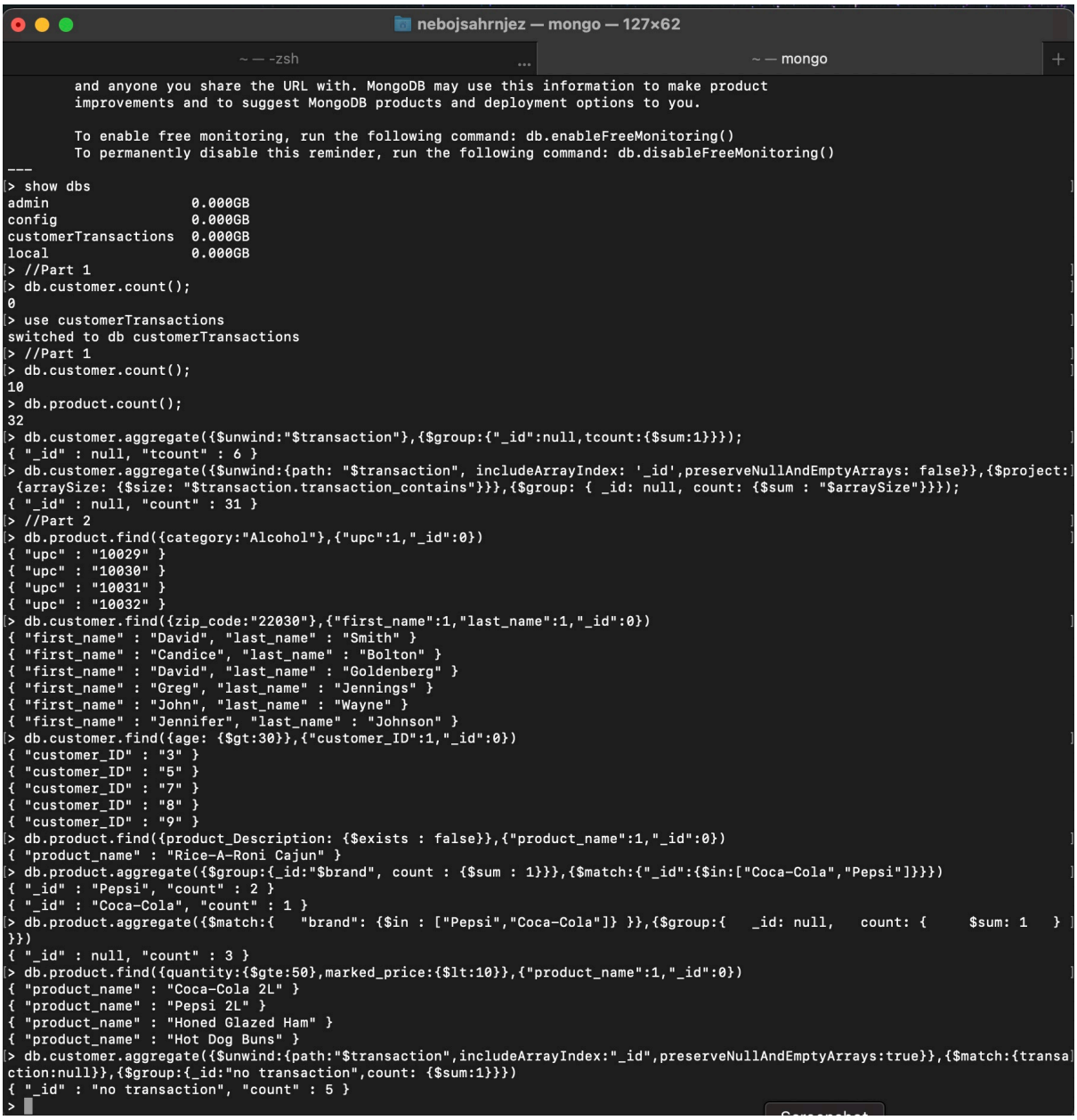
Typed the command “> db.product.aggregate({$group:{_id:"$brand", count : {$sum : 1}}}, {$match:{“_id”:{$in:[“Coca-Cola","Pepsi"]}}})", this command looks at the documents in the product collection, groups them by “brand” and pulls the count for each of these “brands”, it then matches the “_id” to “Coca-Cola” or “Pepsi”. It then returns the two brand names desired and the count of each product with that brand name



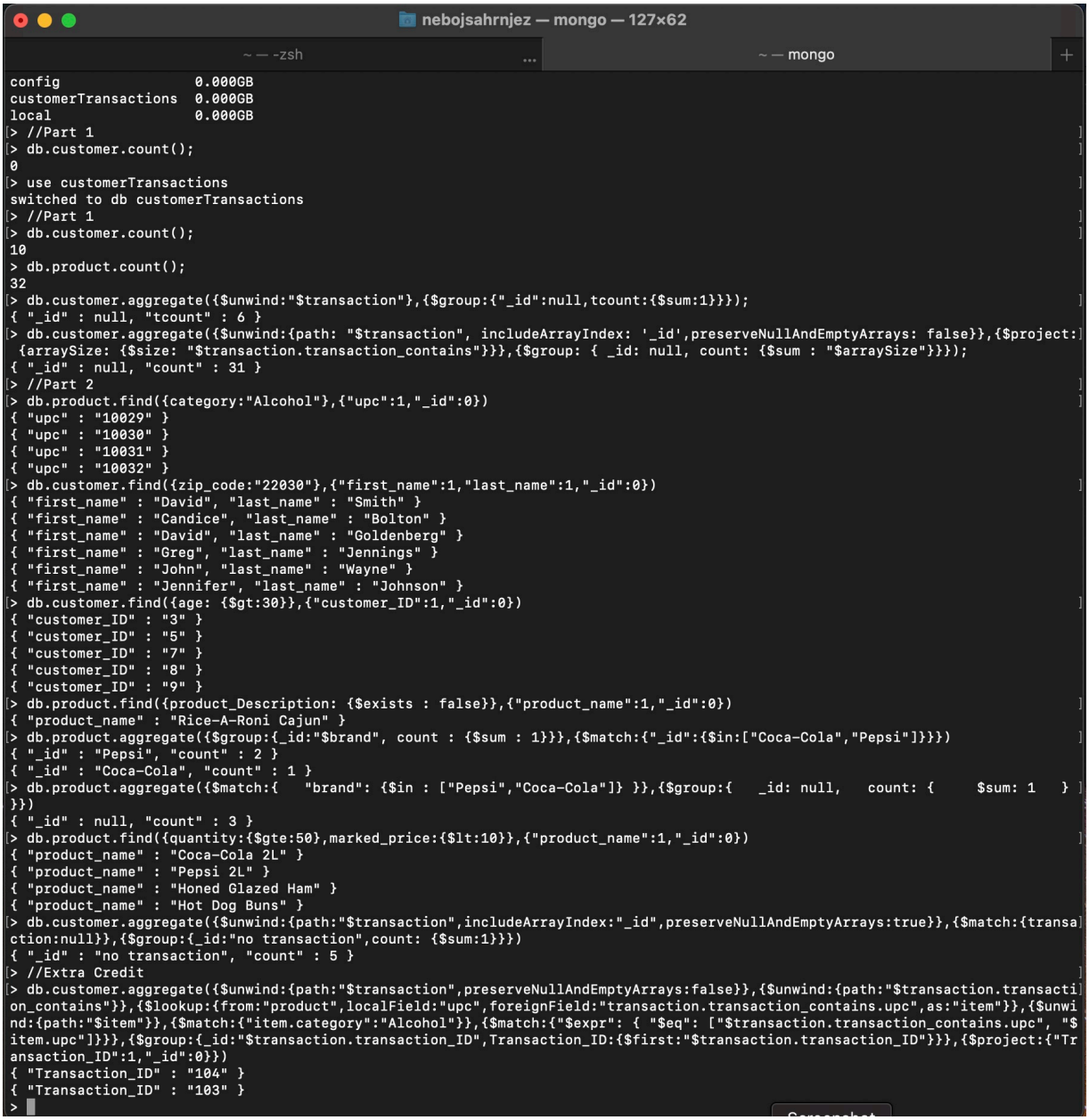
Additionally, if you just want the total count for all products with the brand “Coca-Cola” or “Pepsi” the command is “db.product.aggregate({$match:{ "brand": {$in : ["Pepsi","Coca-Cola"]} }},{$group:{ _id: null, count: { $sum: 1 } }})”



Typed the command “db.product.find({quantity:{$gte:50},marked_price:{$lt:10}}, {“product_name”:1,”_id":0})" , This finds documents in the product collections with a quantity greater than 50 and a price less than 10, I then restricted the output to only show “product_name”



Typed the command “db.customer.aggregate({$unwind: {path:"$transaction",includeArrayIndex:"_id",preserveNullAndEmptyArrays:true}},{$match: {transaction:null}},{$group:{_id:"no transaction",count: {$sum:1}}})” , this command goes into the transactions embedded within the documents in the customer collection, it finds all the transactions that contain “null” or nothing and then counts them



Typed the command “db.customer.aggregate({$unwind: {path:"$transaction",preserveNullAndEmptyArrays:false}},{$unwind: {path:"$transaction.transaction_contains"}},{$lookup: {from:"product",localField:"upc",foreignField:"transaction.transaction_contains.upc",as:"item"}} ,{$unwind:{path:"$item"}},{$match:{"item.category":"Alcohol"}},{$match:{"$expr": { "$eq": ["$transaction.transaction_contains.upc", “$item.upc”]}}},{$group: {_id:"$transaction.transaction_ID",Transaction_ID:{$first:"$transaction.transaction_ID"}}}, {$project:{"Transaction_ID":1,"_id":0}})" , this command creates a new field from the

documents in product called “upc”, this is then used to compare to the “upc” field in the embedded transaction contains.