**Name:-Suraj Namdev Bhilare**

**Seat No:-976**

MONGODB

1. Create a database with name ‘Company’.

2. An ‘Employee’ is a collection of documents with the following fields:

a. Employee ID

b. First Name

c. Last Name

d. Email

e. Phone No.

f. Address (House No, Street, City, State, Country, Pin-code)

g. Salary

h. Designation

i. Experience

j. Date of Joining

k. Birthdate

3. A ‘Transaction’ is a collection of documents with the following fields:

a. Transaction Id,

b. Transaction Date

c. Name (First Name of employee who processed the transaction)

d. Transaction Details (Item Id, Item Name, Quantity, Price)

e. Payment (Type of Payment (Debit/Credit/Cash), Total amount paid, Payment

Successful)

f. Remark (Remark field can be empty.)

Queries:

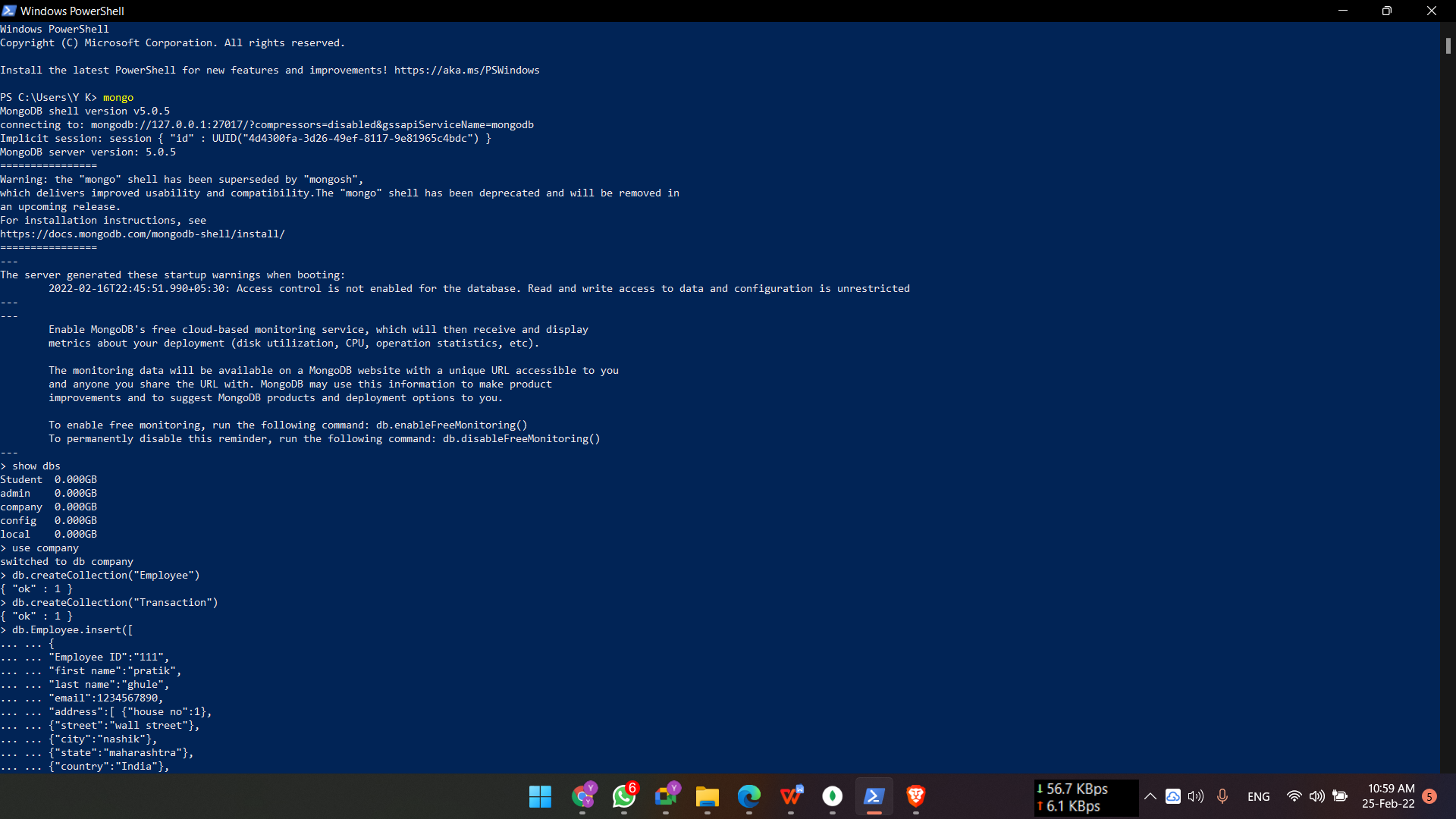
1. Insert at least 5 documents in ‘Employee’ collection.

2. Insert multiple documents (at least 10) into the ‘Transaction’ collection by passing

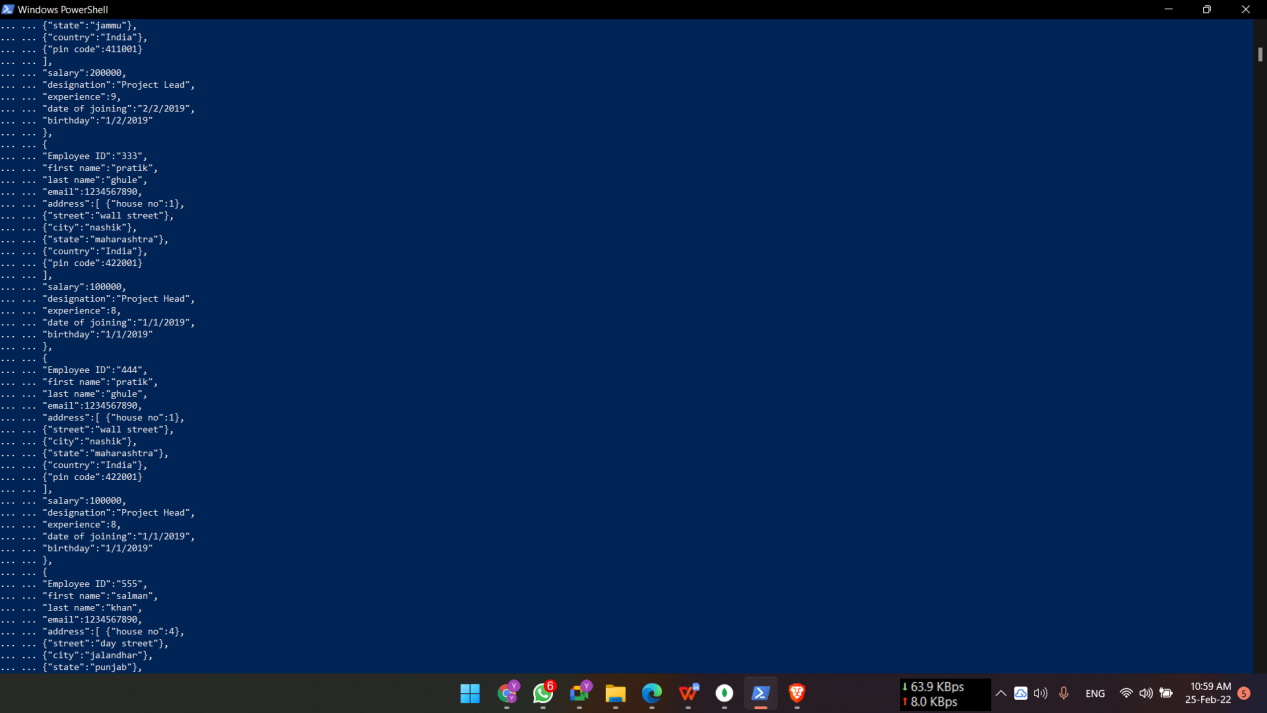
an array of documents to the db.collection.insert () method.

3. Display all the documents of both the collections in a formatted manner.

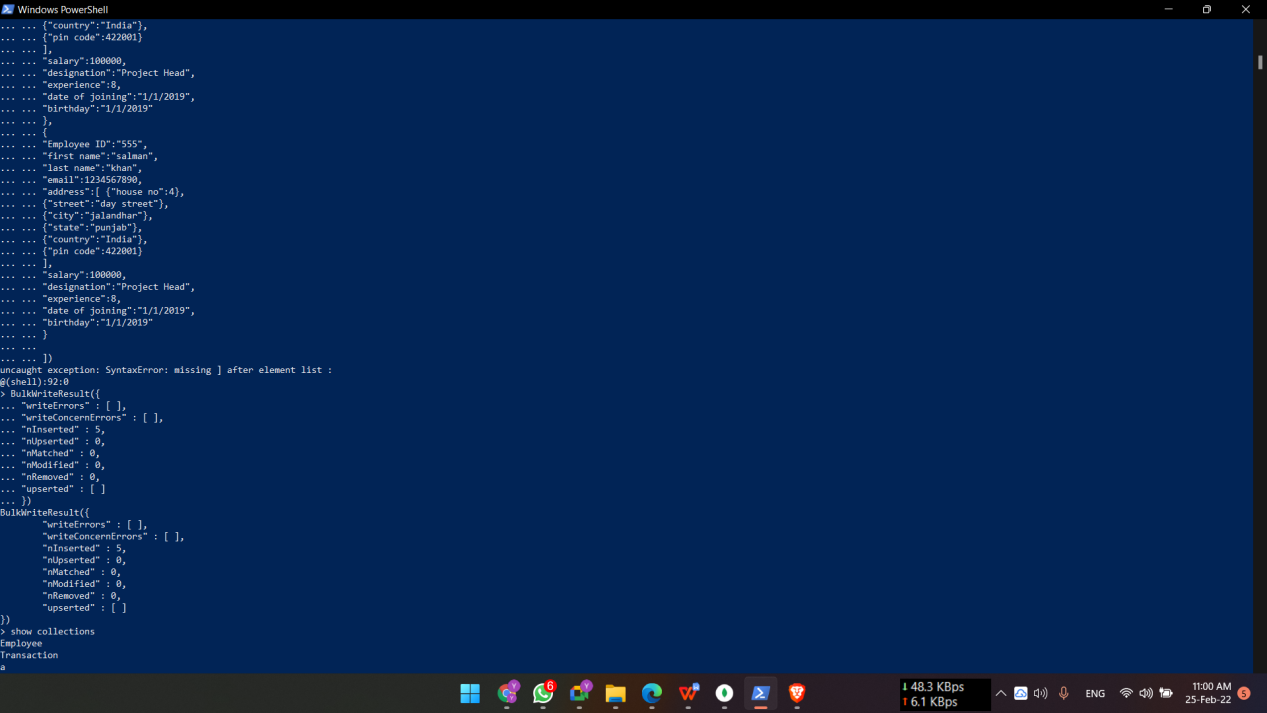
4. Update salary of all employees by giving an increment of Rs. 4000.



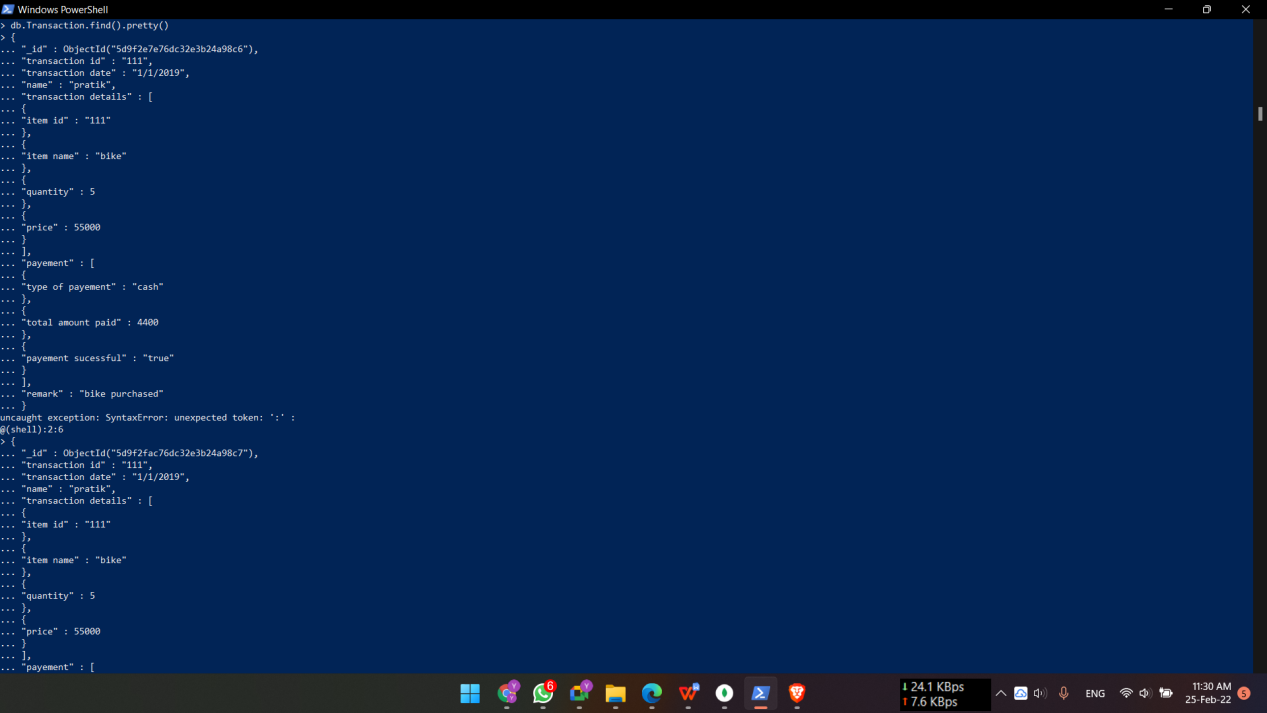
Page1

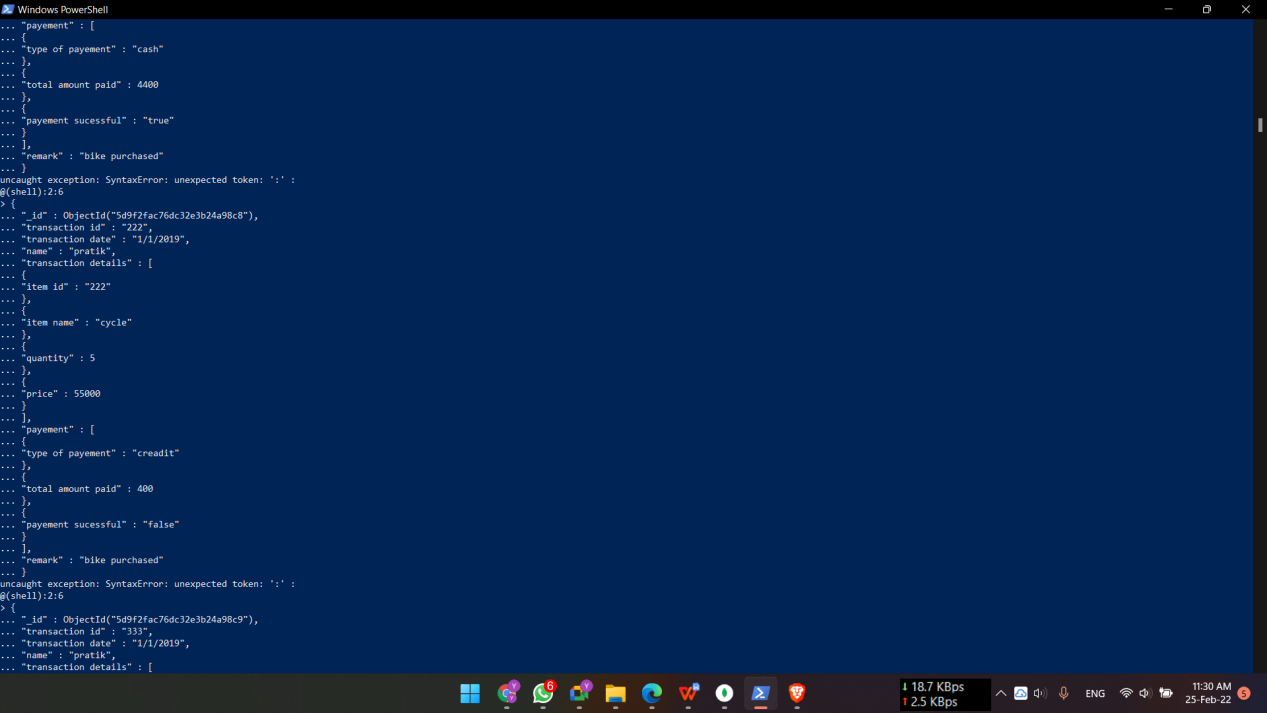


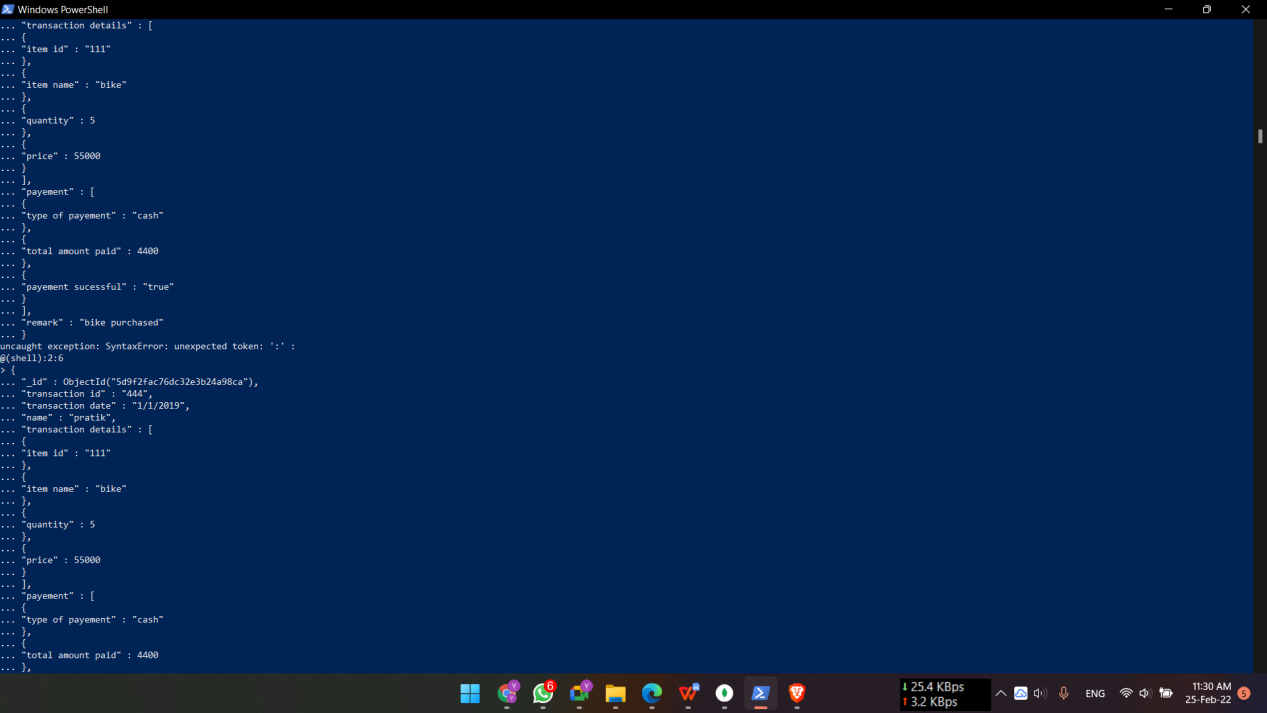
Page2

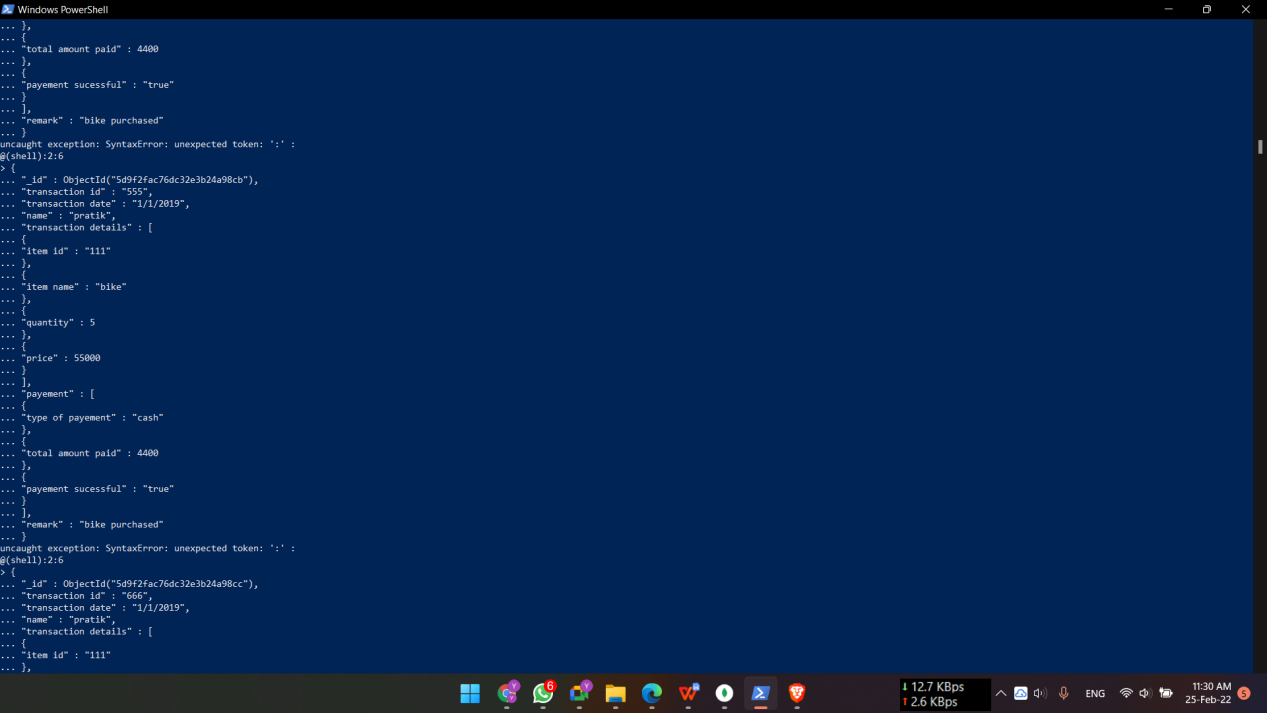


Page3

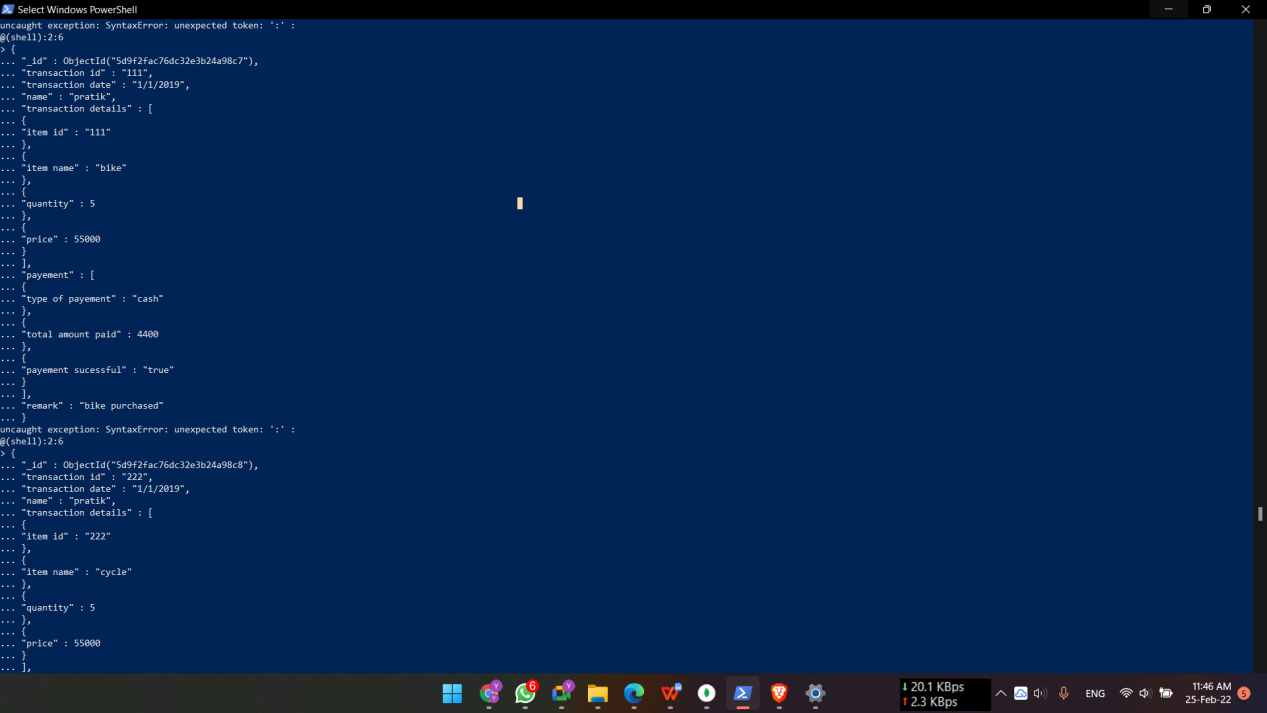






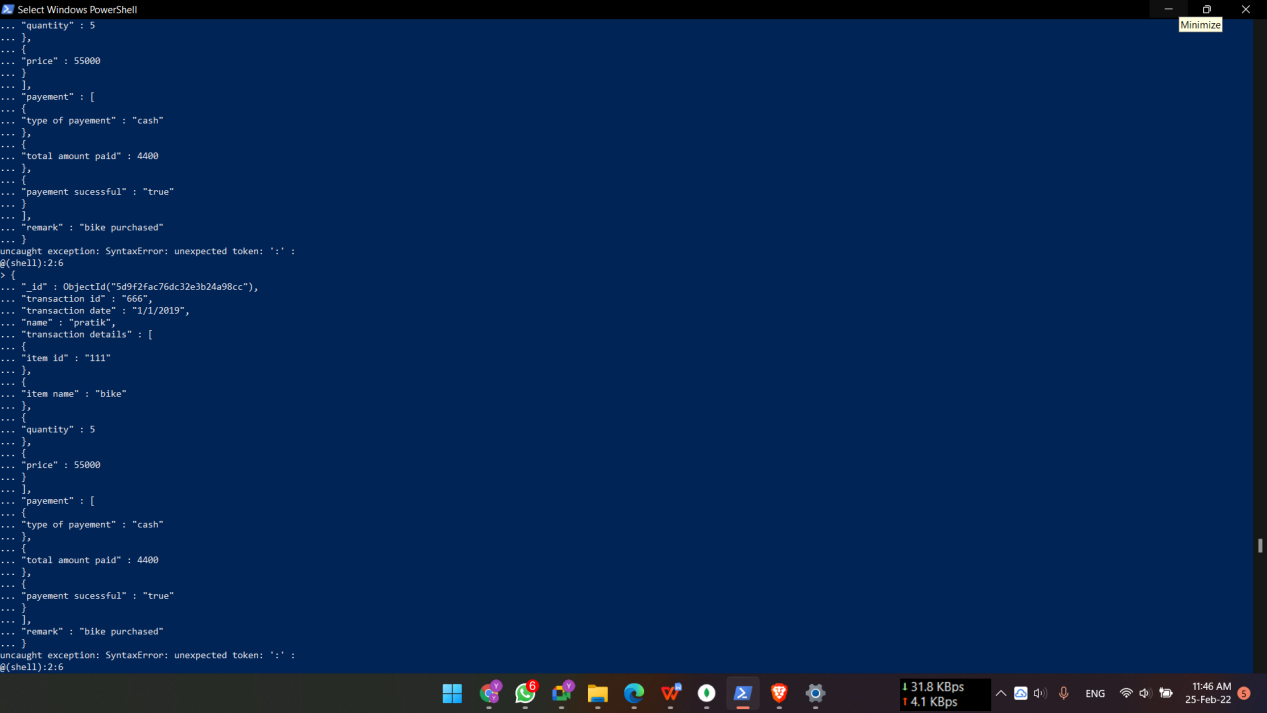


1. Update designation of an employee named “\_ ” from supervisor to manager.









4) Update salary of all employees by giving an increment of Rs. 4000.

> db.Employee.updateMany({},{$inc:{"salary":4000}})

{ "acknowledged" : true, "matchedCount" : 6, "modifiedCount" : 6 }

>

--------------------------------------------------------------------------------------------------------------------------

5) Update the remark for transaction id 201.

> db.Transaction.update({"transaction id":"201"},{$set:{"remarks":"All cool!"}})

WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })

>

------------------------------------------------------------------------------------------------------------------------

6) Update designation of an employee named “\_ ” from supervisor to manager.

> db.Employee.update({"first name":"kabir","last name":"singh"},{$set:{"designation":"manager"}})

WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })

>

------------------------------------------------------------------------------------------------------------------------

7) Update designation of an employee having Employee Id as “101”

> db.Employee.update({"Employee ID":"555"},{$set:{"designation":"manager"}})

WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })

>

--------------------------------------------------------------------------------------------------------------------------

8) Change the address of an employee having Employee Id as “101”

> db.Employee.update({"Employee ID":"555"},{$set:{"address":[{"house no":23},{"street":"franklin street"},{"city":"sydney"},{"state":"sydney"},{"country":"australia"},{"pin code":111111}]}})

WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })

>

--------------------------------------------------------------------------------------------------------------------------

9) Delete transaction made by “ raj” employee on the given date.

> db.Transaction.remove({"name":"raj"},{"transaction date":"2/1/2019"})

WriteResult({ "nRemoved" : 1 })

>

----------------------------------------------------------------------------------------------------------------------

10) Delete all the employees whose first name starts with ‘K’.

(a) using only pattern (without regex)

> db.Employee.remove({"first name":/^k/})

WriteResult({ "nRemoved" : 4 })

>

(b) using regex

> db.Employee.remove( { "first name": { $regex: /^k.\*/ } } )

WriteResult({ "nRemoved" : 3 })

>

