1.Create a Database named 'MYWall'

2.Create two collections named 'user and ''post'

3.Insert some post related information like post_id, title, description , posted_by, comments, created_at etc…

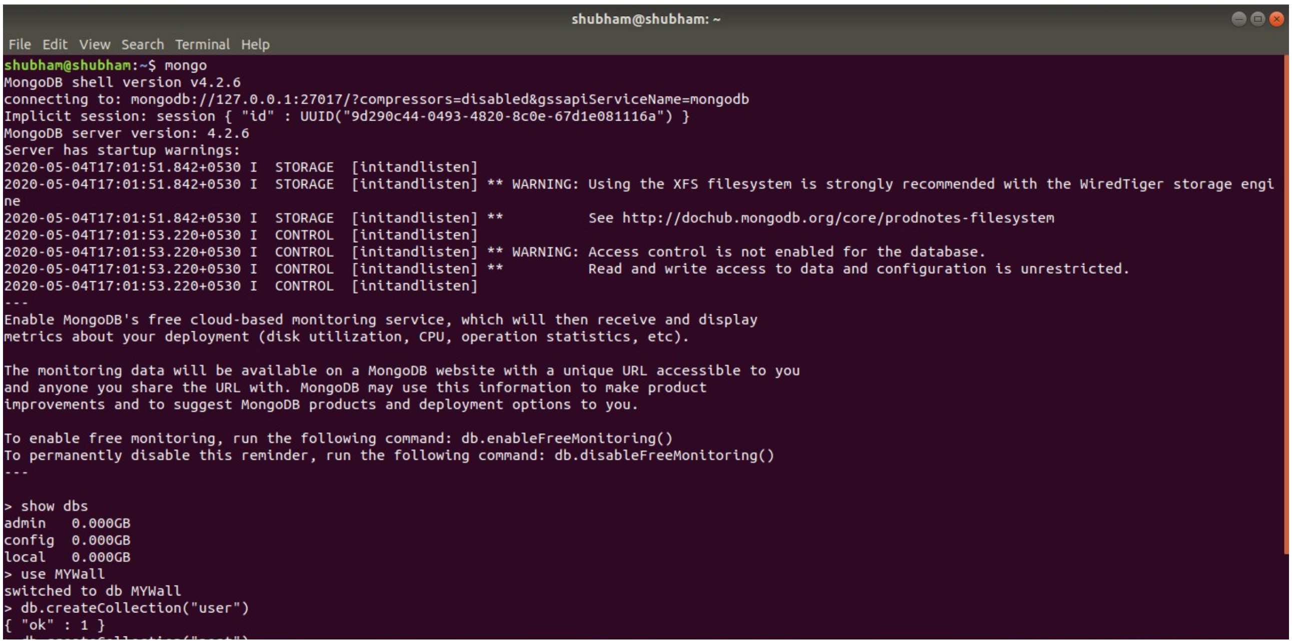
4.Comments should be an array of objects which has comment_id, message, timestamp

5.Write query to add a new comment for a post

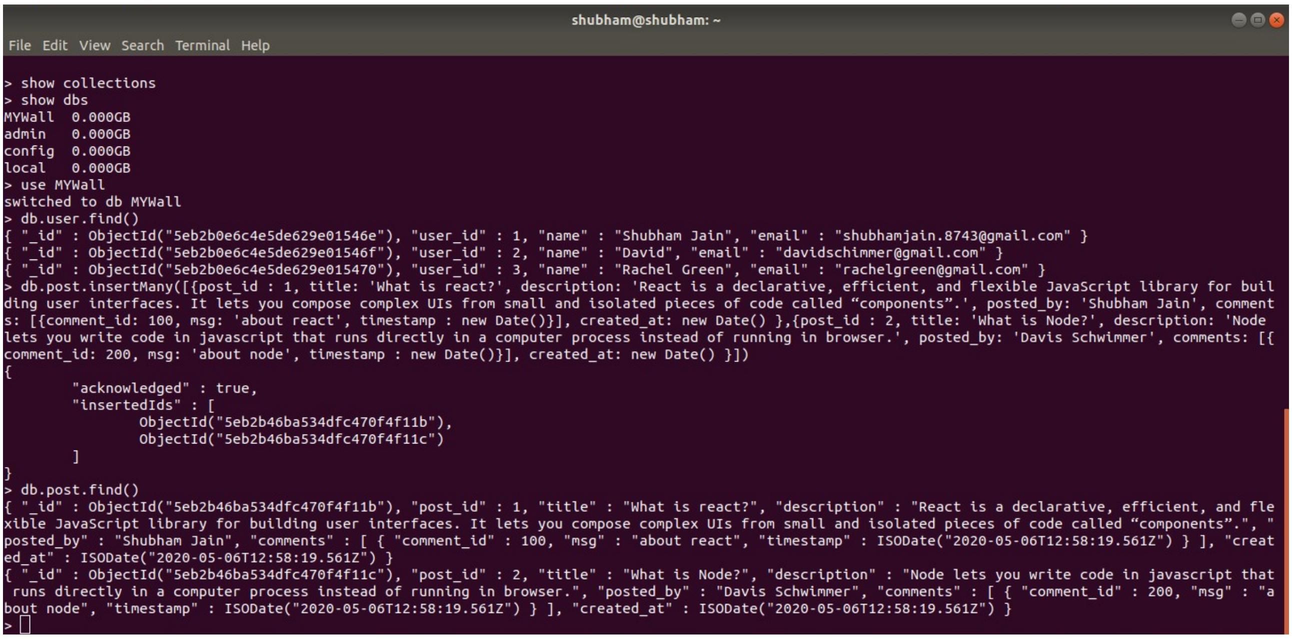
6.Write query to update description of a post

7.Write query to update comment message

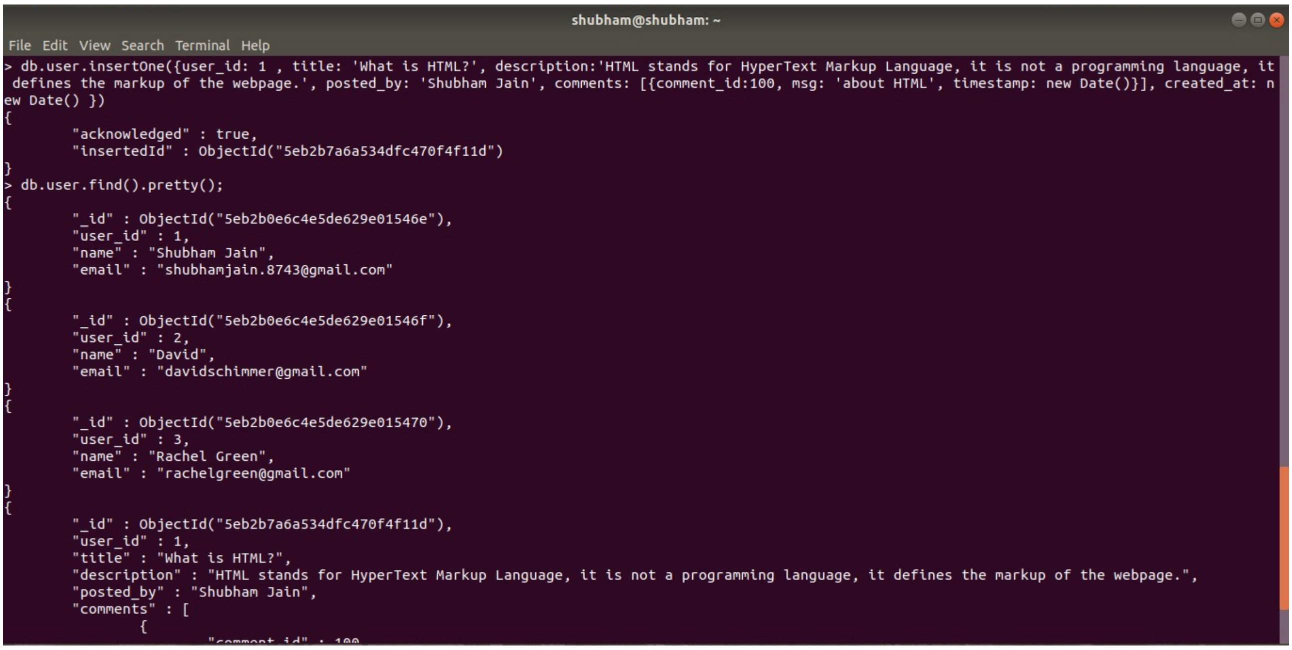
8.Write query to remove a comment



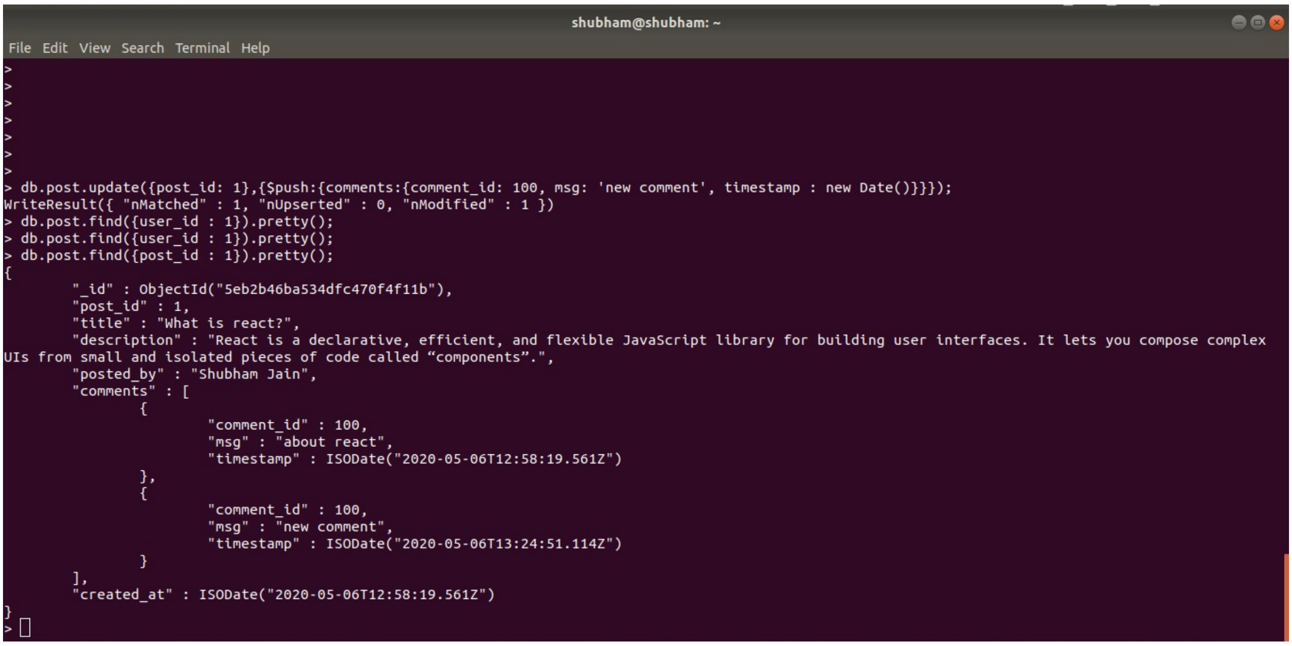
```
shubham@shubham:~$ mongo
MongoDB shell version v4.2.6
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("9d290c44-0493-4820-8c0e-67d1e081116a") }
MongoDB server version: 4.2.6
Server has startup warnings:
2020-05-04T17:01:51.842+0530 I  STORAGE  [initandlisten]
2020-05-04T17:01:51.842+0530 I  STORAGE  [initandlisten] ** WARNING: Using the XFS filesystem is strongly recommended with the WiredTiger storage engi
ne
2020-05-04T17:01:51.842+0530 I  STORAGE  [initandlisten] **          See http://dochub.mongodb.org/core/prodnotes-filesystem
2020-05-04T17:01:53.220+0530 I  CONTROL  [initandlisten]
2020-05-04T17:01:53.220+0530 I  CONTROL  [initandlisten] ** WARNING: Access control is not enabled for the database.
2020-05-04T17:01:53.220+0530 I  CONTROL  [initandlisten] **          Read and write access to data and configuration is unrestricted.
2020-05-04T17:01:53.220+0530 I  CONTROL  [initandlisten]
---
Enable MongoDB's free cloud-based monitoring service, which will then receive and display
metrics about your deployment (disk utilization, CPU, operation statistics, etc).

The monitoring data will be available on a MongoDB website with a unique URL accessible to you
and anyone you share the URL with. MongoDB may use this information to make product
improvements and to suggest MongoDB products and deployment options to you.

To enable free monitoring, run the following command: db.enableFreeMonitoring()
To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
---

> show dbs
admin   0.000GB
config  0.000GB
local   0.000GB
> use MYWall
switched to db MYWall
> db.createCollection("user")
{ "ok" : 1 }
```



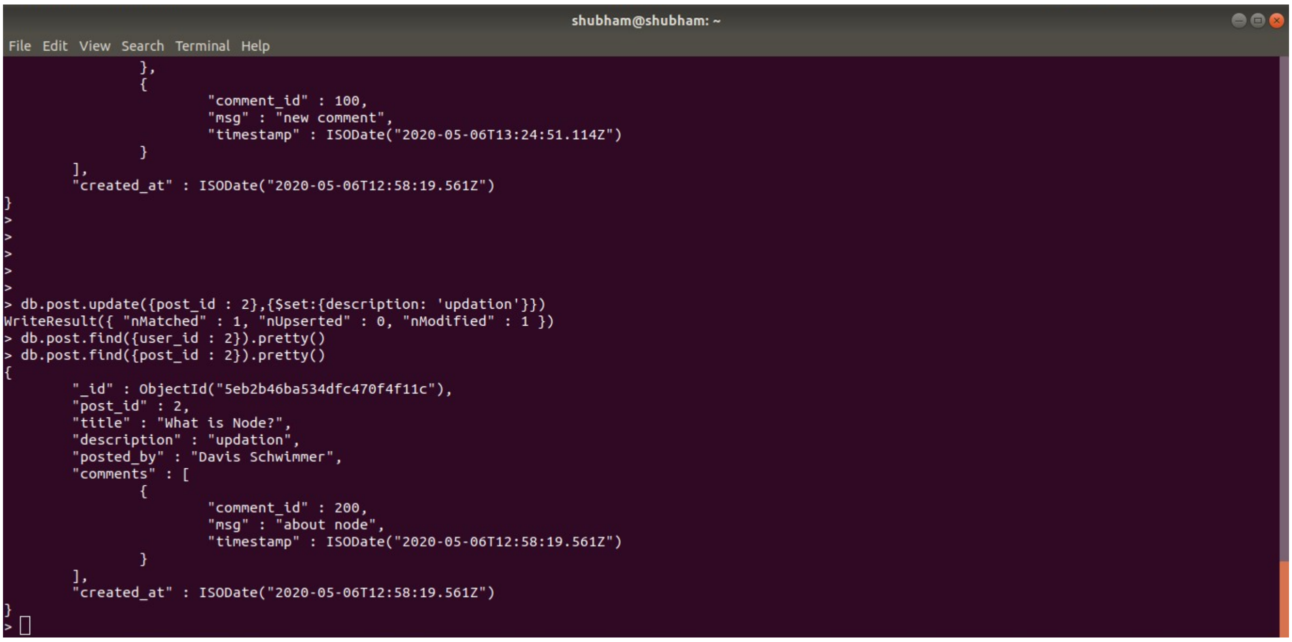
```
> show collections
> show dbs
MYWall  0.000GB
admin   0.000GB
config  0.000GB
local   0.000GB
> use MYWall
switched to db MYWall
> db.user.find()
{ "_id" : ObjectId("5eb2b0e6c4e5de629e01546e"), "user_id" : 1, "name" : "Shubham Jain", "email" : "shubhamjain.8743@gmail.com" }
{ "_id" : ObjectId("5eb2b0e6c4e5de629e01546f"), "user_id" : 2, "name" : "David", "email" : "davidschimmer@gmail.com" }
{ "_id" : ObjectId("5eb2b0e6c4e5de629e015470"), "user_id" : 3, "name" : "Rachel Green", "email" : "rachelgreen@gmail.com" }
> db.post.insertMany([{post_id : 1, title: 'What is react?', description: 'React is a declarative, efficient, and flexible JavaScript library for buil
ding user interfaces. It lets you compose complex UIs from small and isolated pieces of code called “components”.', posted_by: 'Shubham Jain', comment
s: [{comment_id: 100, msg: 'about react', timestamp : new Date()}], created_at: new Date() },{post_id : 2, title: 'What is Node?', description: 'Node
lets you write code in javascript that runs directly in a computer process instead of running in browser.', posted_by: 'Davis Schwimmer', comments: [{
comment_id: 200, msg: 'about node', timestamp : new Date()}], created_at: new Date() }])
{
        "acknowledged" : true,
        "insertedIds" : [
                ObjectId("5eb2b46ba534dfc470f4f11b"),
                ObjectId("5eb2b46ba534dfc470f4f11c")
        ]
}
> db.post.find()
{ "_id" : ObjectId("5eb2b46ba534dfc470f4f11b"), "post_id" : 1, "title" : "What is react?", "description" : "React is a declarative, efficient, and fle
xible JavaScript library for building user interfaces. It lets you compose complex UIs from small and isolated pieces of code called “components”.", "
posted_by" : "Shubham Jain", "comments" : [ { "comment_id" : 100, "msg" : "about react", "timestamp" : ISODate("2020-05-06T12:58:19.561Z") } ], "creat
ed_at" : ISODate("2020-05-06T12:58:19.561Z") }
{ "_id" : ObjectId("5eb2b46ba534dfc470f4f11c"), "post_id" : 2, "title" : "What is Node?", "description" : "Node lets you write code in javascript that
 runs directly in a computer process instead of running in browser.", "posted_by" : "Davis Schwimmer", "comments" : [ { "comment_id" : 200, "msg" : "a
bout node", "timestamp" : ISODate("2020-05-06T12:58:19.561Z") } ], "created_at" : ISODate("2020-05-06T12:58:19.561Z") }
>
```



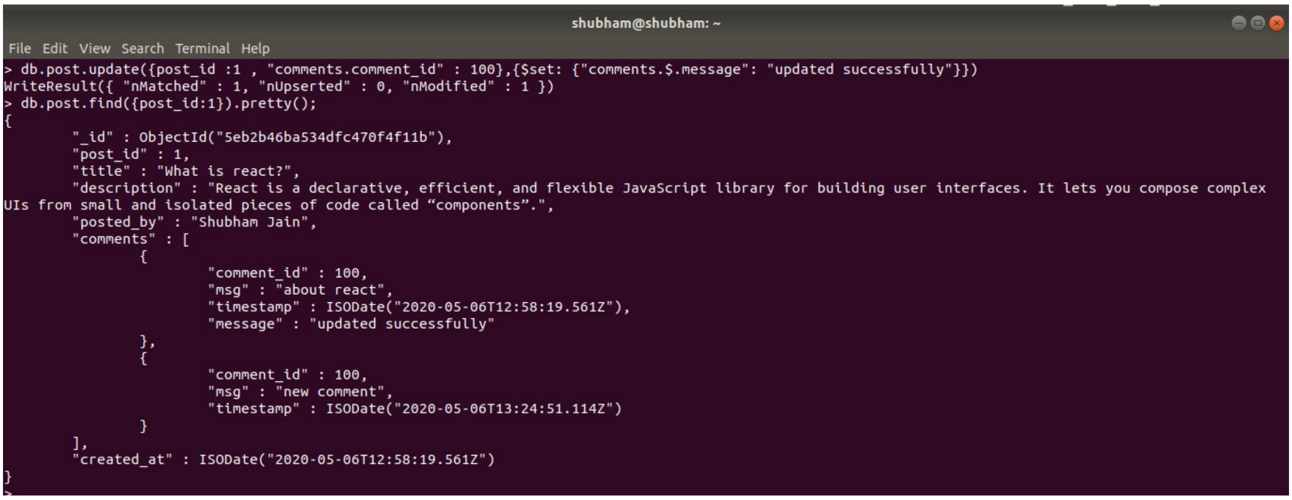
```
> db.user.insertOne({user_id: 1 , title: 'What is HTML?', description:'HTML stands for HyperText Markup Language, it is not a programming language, it
 defines the markup of the webpage.', posted_by: 'Shubham Jain', comments: [{comment_id:100, msg: 'about HTML', timestamp: new Date()}], created_at: n
ew Date() })
{
        "acknowledged" : true,
        "insertedId" : ObjectId("5eb2b7a6a534dfc470f4f11d")
}
> db.user.find().pretty();
{
        "_id" : ObjectId("5eb2b0e6c4e5de629e01546e"),
        "user_id" : 1,
        "name" : "Shubham Jain",
        "email" : "shubhamjain.8743@gmail.com"
}
{
        "_id" : ObjectId("5eb2b0e6c4e5de629e01546f"),
        "user_id" : 2,
        "name" : "David",
        "email" : "davidschimmer@gmail.com"
}
{
        "_id" : ObjectId("5eb2b0e6c4e5de629e015470"),
        "user_id" : 3,
        "name" : "Rachel Green",
        "email" : "rachelgreen@gmail.com"
}
{
        "_id" : ObjectId("5eb2b7a6a534dfc470f4f11d"),
        "user_id" : 1,
        "title" : "What is HTML?",
        "description" : "HTML stands for HyperText Markup Language, it is not a programming language, it defines the markup of the webpage.",
        "posted_by" : "Shubham Jain",
        "comments" : [
                {
                        "comment_id" : 100
```



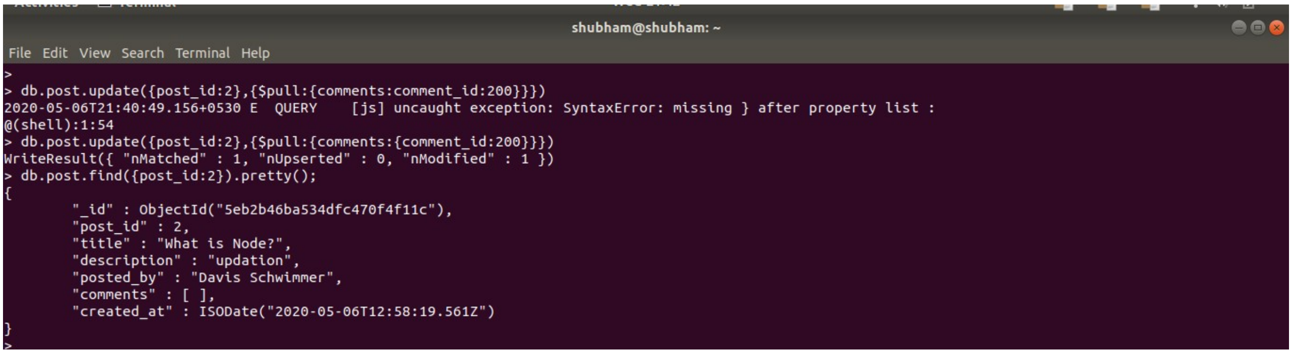
```
>
>
>
>
>
>
>
> db.post.update({post_id: 1},{$push:{comments:{comment_id: 100, msg: 'new comment', timestamp : new Date()}}});
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.post.find({user_id : 1}).pretty();
> db.post.find({user_id : 1}).pretty();
> db.post.find({post_id : 1}).pretty();
{
        "_id" : ObjectId("5eb2b46ba534dfc470f4f11b"),
        "post_id" : 1,
        "title" : "What is react?",
        "description" : "React is a declarative, efficient, and flexible JavaScript library for building user interfaces. It lets you compose complex
UIs from small and isolated pieces of code called “components”.",
        "posted_by" : "Shubham Jain",
        "comments" : [
                {
                        "comment_id" : 100,
                        "msg" : "about react",
                        "timestamp" : ISODate("2020-05-06T12:58:19.561Z")
                },
                {
                        "comment_id" : 100,
                        "msg" : "new comment",
                        "timestamp" : ISODate("2020-05-06T13:24:51.114Z")
                }
        ],
        "created_at" : ISODate("2020-05-06T12:58:19.561Z")
}
> 
```



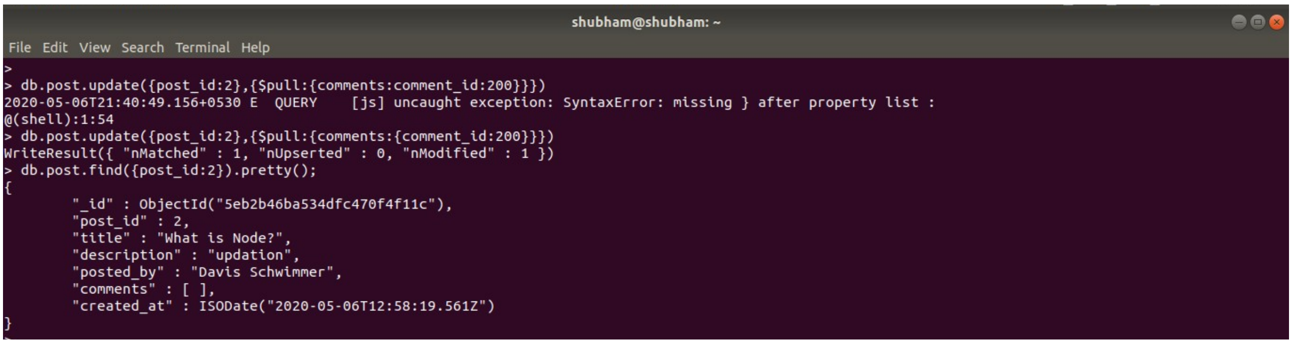
```
                },
                {
                        "comment_id" : 100,
                        "msg" : "new comment",
                        "timestamp" : ISODate("2020-05-06T13:24:51.114Z")
                }
        ],
        "created_at" : ISODate("2020-05-06T12:58:19.561Z")
}
>
>
>
>
>
> db.post.update({post_id : 2},{$set:{description: 'updation'}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.post.find({user_id : 2}).pretty()
> db.post.find({post_id : 2}).pretty()
{
        "_id" : ObjectId("5eb2b46ba534dfc470f4f11c"),
        "post_id" : 2,
        "title" : "What is Node?",
        "description" : "updation",
        "posted_by" : "Davis Schwimmer",
        "comments" : [
                {
                        "comment_id" : 200,
                        "msg" : "about node",
                        "timestamp" : ISODate("2020-05-06T12:58:19.561Z")
                }
        ],
        "created_at" : ISODate("2020-05-06T12:58:19.561Z")
}
> 
```



```
> db.post.update({post_id :1 , "comments.comment_id" : 100},{$set: {"comments.$.message": "updated successfully"}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.post.find({post_id:1}).pretty();
{
        "_id" : ObjectId("5eb2b46ba534dfc470f4f11b"),
        "post_id" : 1,
        "title" : "What is react?",
        "description" : "React is a declarative, efficient, and flexible JavaScript library for building user interfaces. It lets you compose complex
UIs from small and isolated pieces of code called “components”.",
        "posted_by" : "Shubham Jain",
        "comments" : [
                {
                        "comment_id" : 100,
                        "msg" : "about react",
                        "timestamp" : ISODate("2020-05-06T12:58:19.561Z"),
                        "message" : "updated successfully"
                },
                {
                        "comment_id" : 100,
                        "msg" : "new comment",
                        "timestamp" : ISODate("2020-05-06T13:24:51.114Z")
                }
        ],
        "created_at" : ISODate("2020-05-06T12:58:19.561Z")
}
>
```



```
>
> db.post.update({post_id:2},{$pull:{comments:comment_id:200}}})
2020-05-06T21:40:49.156+0530 E  QUERY    [js] uncaught exception: SyntaxError: missing } after property list :
@(shell):1:54
> db.post.update({post_id:2},{$pull:{comments:{comment_id:200}}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.post.find({post_id:2}).pretty();
{
        "_id" : ObjectId("5eb2b46ba534dfc470f4f11c"),
        "post_id" : 2,
        "title" : "What is Node?",
        "description" : "updation",
        "posted_by" : "Davis Schwimmer",
        "comments" : [ ],
        "created_at" : ISODate("2020-05-06T12:58:19.561Z")
}
>
```



```
>
> db.post.update({post_id:2},{$pull:{comments:comment_id:200}}})
2020-05-06T21:40:49.156+0530 E  QUERY    [js] uncaught exception: SyntaxError: missing } after property list :
@(shell):1:54
> db.post.update({post_id:2},{$pull:{comments:{comment_id:200}}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.post.find({post_id:2}).pretty();
{
        "_id" : ObjectId("5eb2b46ba534dfc470f4f11c"),
        "post_id" : 2,
        "title" : "What is Node?",
        "description" : "updation",
        "posted_by" : "Davis Schwimmer",
        "comments" : [ ],
        "created_at" : ISODate("2020-05-06T12:58:19.561Z")
}
```