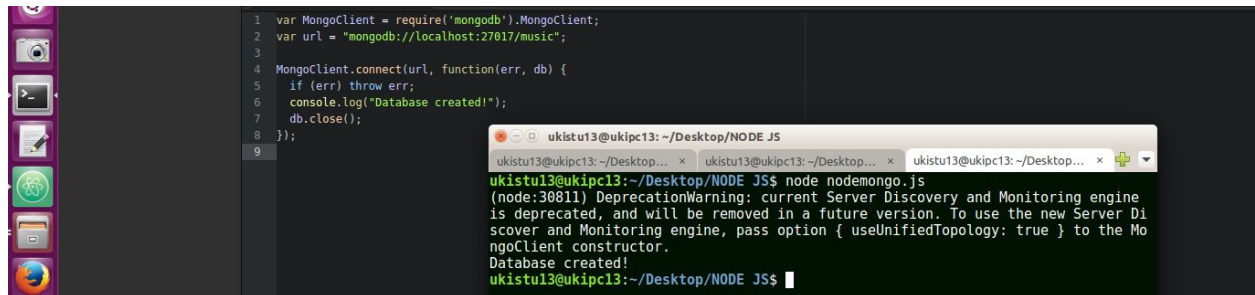


## Node.js Exercise 2

### Node.js MongoDB

01. Complete mongodbx1 with the MongoClient object you just created in Node.js

1) Create a Database called music.

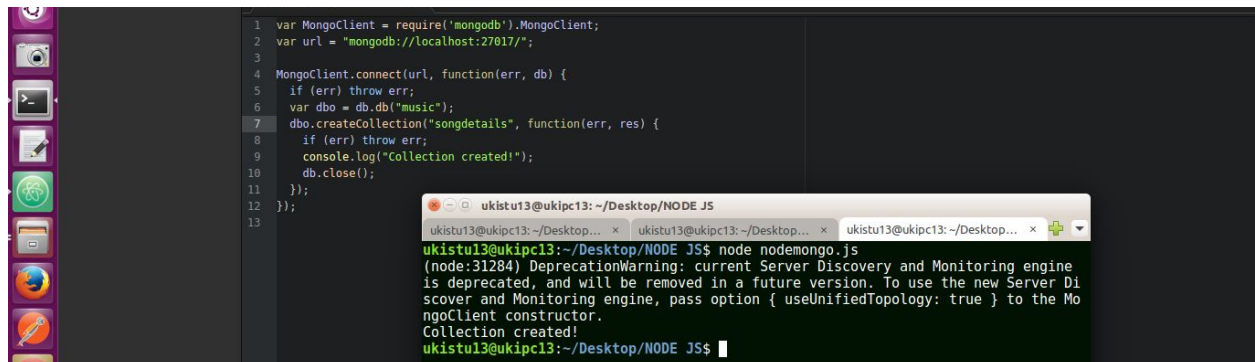


```
1 var MongoClient = require('mongodb').MongoClient;
2 var url = "mongodb://localhost:27017/music";
3
4 MongoClient.connect(url, function(err, db) {
5   if (err) throw err;
6   console.log("Database created!");
7   db.close();
8 });
9
```

Terminal output:

```
ukistu13@ukipc13: ~/Desktop/NODE JS
ukistu13@ukipc13:~/Desktop/NODE JS$ node nodemongo.js
(node:30811) DeprecationWarning: current Server Discovery and Monitoring engine
is deprecated, and will be removed in a future version. To use the new Server Di
scover and Monitoring engine, pass option { useUnifiedTopology: true } to the Mo
ngoClient constructor.
Database created!
ukistu13@ukipc13:~/Desktop/NODE JS$
```

2) Create a collection called songdetails.

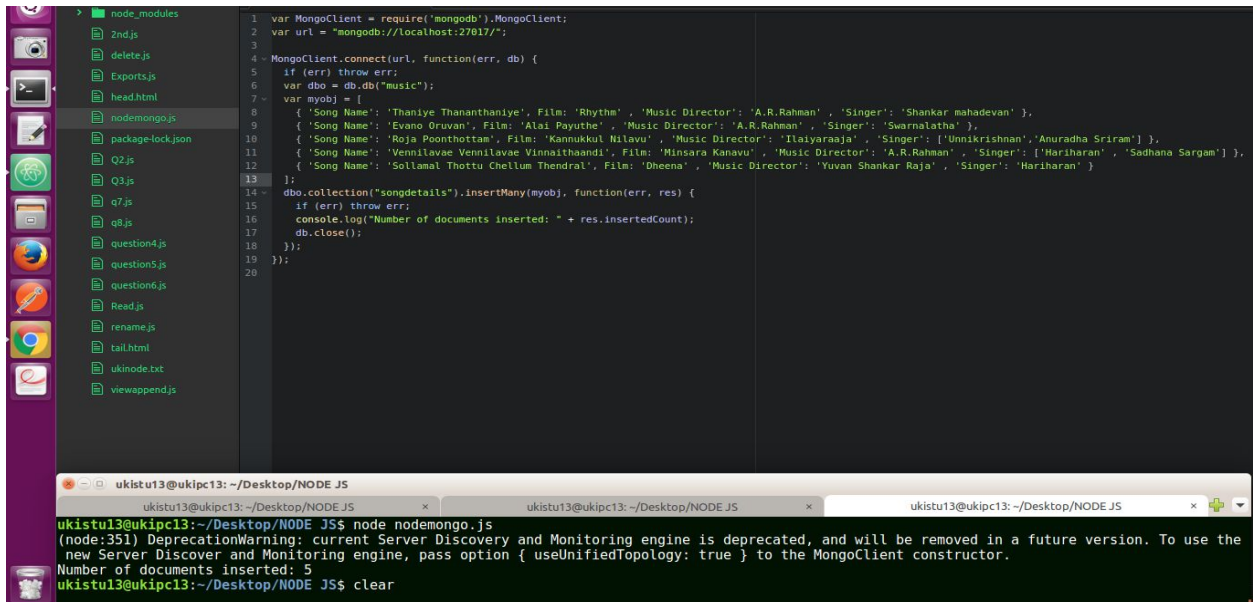


```
1 var MongoClient = require('mongodb').MongoClient;
2 var url = "mongodb://localhost:27017/";
3
4 MongoClient.connect(url, function(err, db) {
5   if (err) throw err;
6   var dbo = db.db("music");
7   dbo.createCollection("songdetails", function(err, res) {
8     if (err) throw err;
9     console.log("Collection created!");
10    db.close();
11  });
12 });
13
```

Terminal output:

```
ukistu13@ukipc13: ~/Desktop/NODE JS
ukistu13@ukipc13:~/Desktop/NODE JS$ node nodemongo.js
(node:31284) DeprecationWarning: current Server Discovery and Monitoring engine
is deprecated, and will be removed in a future version. To use the new Server Di
scover and Monitoring engine, pass option { useUnifiedTopology: true } to the Mo
ngoClient constructor.
Collection created!
ukistu13@ukipc13:~/Desktop/NODE JS$
```

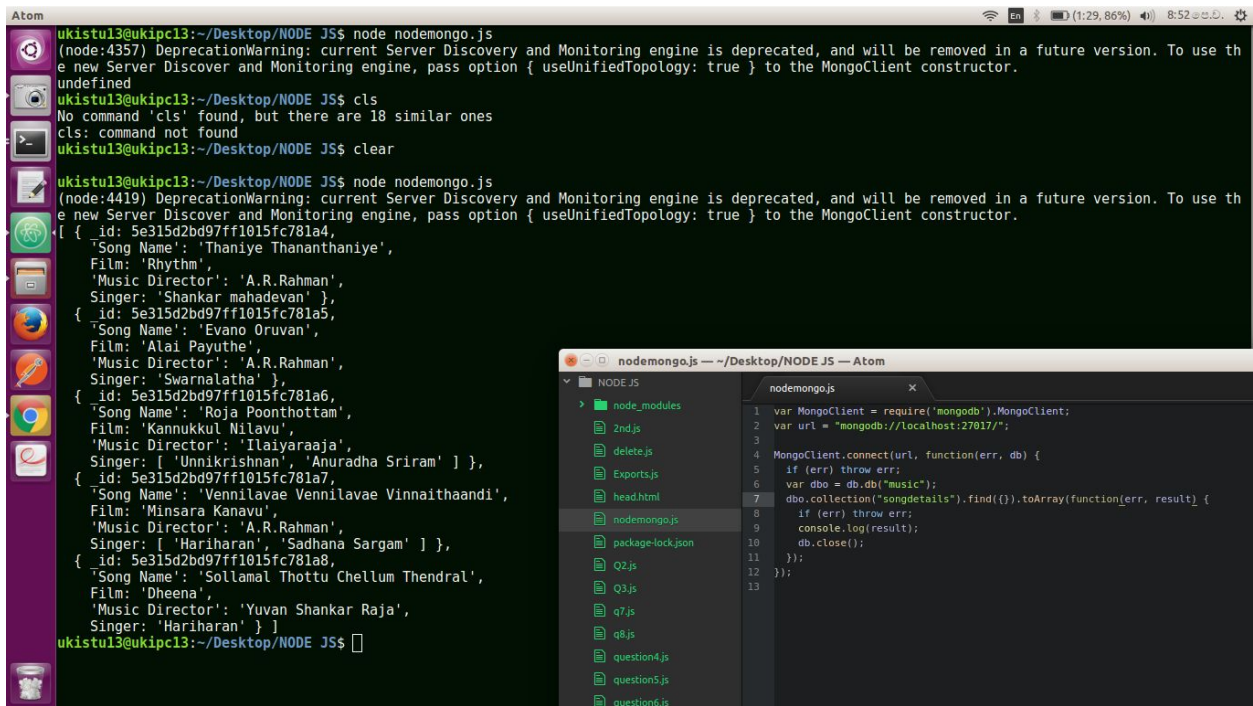
3) Create the above 5 song documents.



```
1 var MongoClient = require('mongodb').MongoClient;
2 var url = "mongodb://localhost:27017/";
3
4 MongoClient.connect(url, function(err, db) {
5   if (err) throw err;
6   var dbo = db.db("music");
7   var myobj = [
8     { 'Song Name': 'Thaniye Thananthaniye', 'Film': 'Rhythm', 'Music Director': 'A.R.Rahman', 'Singer': 'Shankar mahadevan' },
9     { 'Song Name': 'Evano Oruvan', 'Film': 'Alai Payuthe', 'Music Director': 'A.R.Rahman', 'Singer': 'Swarnalatha' },
10    { 'Song Name': 'Raja Poonthottam', 'Film': 'Kannukkul Nilavu', 'Music Director': 'Ilaiyaraaja', 'Singer': ['Unnikrishnan', 'Anuradha Sriram'] },
11    { 'Song Name': 'Vennilavae Vennilavae Vinnaihaandi', 'Film': 'Minsara Kanavu', 'Music Director': 'A.R.Rahman', 'Singer': ['Hariharan', 'Sadhana Sargam'] },
12    { 'Song Name': 'Sollamal Thottu Chellum Thendral', 'Film': 'Dheena', 'Music Director': 'Yuvan Shankar Raja', 'Singer': 'Hariharan' }
13  ];
14  dbo.collection("songdetails").insertMany(myobj, function(err, res) {
15    if (err) throw err;
16    console.log("Number of documents inserted: " + res.insertedCount);
17    db.close();
18  });
19 });
20
```

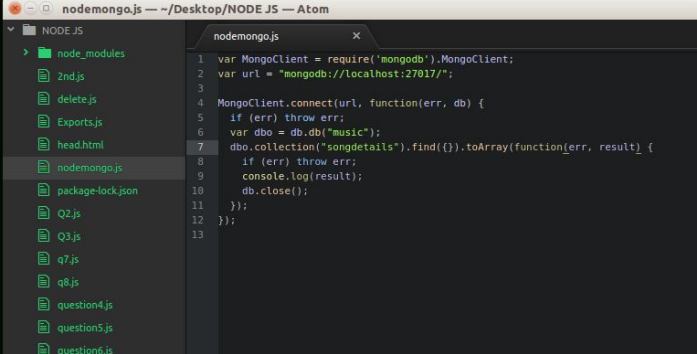
ukistul3@ukipc13: ~/Desktop/NODE JS  
ukistul3@ukipc13:~/Desktop/NODE JS\$ node nodemongo.js  
(node:351) DeprecationWarning: current Server Discovery and Monitoring engine is deprecated, and will be removed in a future version. To use the new Server Discover and Monitoring engine, pass option { useUnifiedTopology: true } to the MongoClient constructor.  
Number of documents inserted: 5  
ukistul3@ukipc13:~/Desktop/NODE JS\$ clear

4) List all documents created.



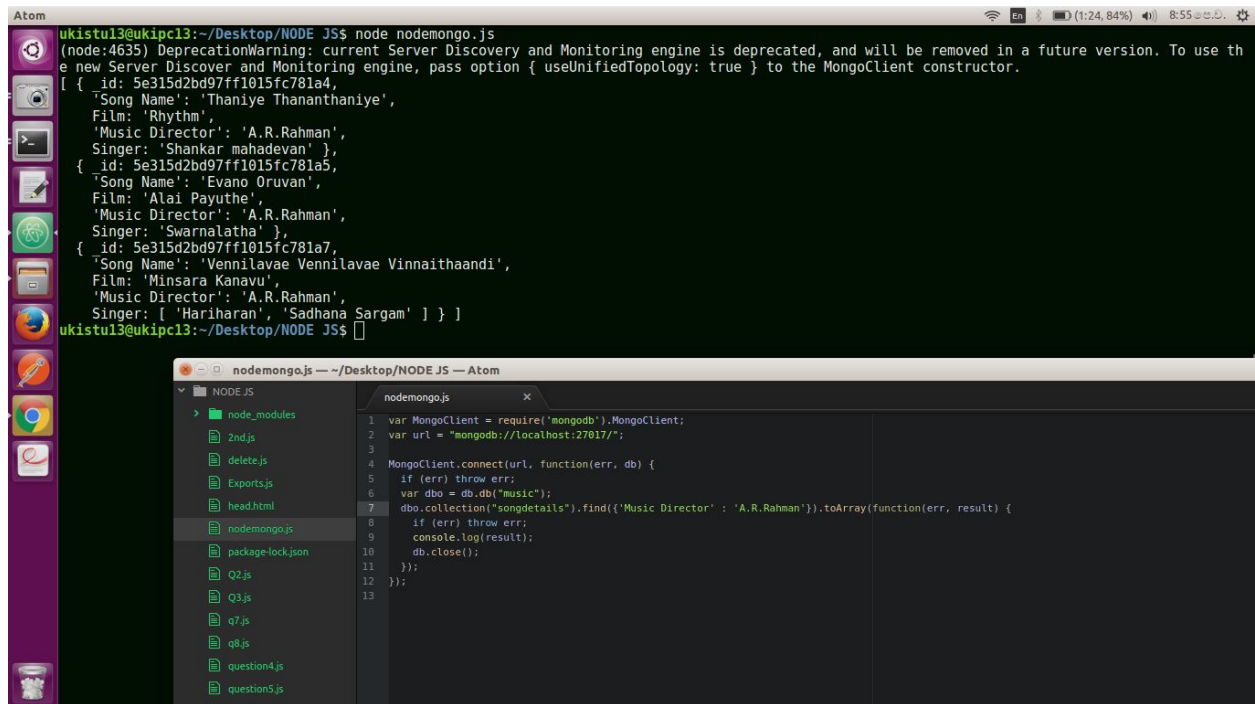
```
ukistul3@ukipc13:~/Desktop/NODE JS$ node nodemongo.js
(node:4357) DeprecationWarning: current Server Discovery and Monitoring engine is deprecated, and will be removed in a future version. To use the new Server Discover and Monitoring engine, pass option { useUnifiedTopology: true } to the MongoClient constructor.
undefined
ukistul3@ukipc13:~/Desktop/NODE JS$ cls
No command 'cls' found, but there are 18 similar ones
cls: command not found
ukistul3@ukipc13:~/Desktop/NODE JS$ clear

ukistul3@ukipc13:~/Desktop/NODE JS$ node nodemongo.js
(node:4419) DeprecationWarning: current Server Discovery and Monitoring engine is deprecated, and will be removed in a future version. To use the new Server Discover and Monitoring engine, pass option { useUnifiedTopology: true } to the MongoClient constructor.
[ { _id: 5e315d2bd97ff1015fc781a4,
  'Song Name': 'Thaniye Thananthaniye',
  Film: 'Rhythm',
  'Music Director': 'A.R.Rahman',
  Singer: 'Shankar mahadevan' },
  { _id: 5e315d2bd97ff1015fc781a5,
    'Song Name': 'Evano Oruvan',
    Film: 'Alai Payuthe',
    'Music Director': 'A.R.Rahman',
    Singer: 'Swarnalatha' },
  { _id: 5e315d2bd97ff1015fc781a6,
    'Song Name': 'Raja Poonthottam',
    Film: 'Kannukkul Nilavu',
    'Music Director': 'Ilaiyaraaja',
    Singer: [ 'Unnikrishnan', 'Anuradha Sriram' ] },
  { _id: 5e315d2bd97ff1015fc781a7,
    'Song Name': 'Vennilavae Vennilavae Vinnaihaandi',
    Film: 'Minsara Kanavu',
    'Music Director': 'A.R.Rahman',
    Singer: [ 'Hariharan', 'Sadhana Sargam' ] },
  { _id: 5e315d2bd97ff1015fc781a8,
    'Song Name': 'Sollamal Thottu Chellum Thendral',
    Film: 'Dheena',
    'Music Director': 'Yuvan Shankar Raja',
    Singer: 'Hariharan' } ]
ukistul3@ukipc13:~/Desktop/NODE JS$
```



```
1 var MongoClient = require('mongodb').MongoClient;
2 var url = "mongodb://localhost:27017/";
3
4 MongoClient.connect(url, function(err, db) {
5   if (err) throw err;
6   var dbo = db.db("music");
7   dbo.collection("songdetails").find().toArray(function(err, result) {
8     if (err) throw err;
9     console.log(result);
10    db.close();
11  });
12 });
13
```

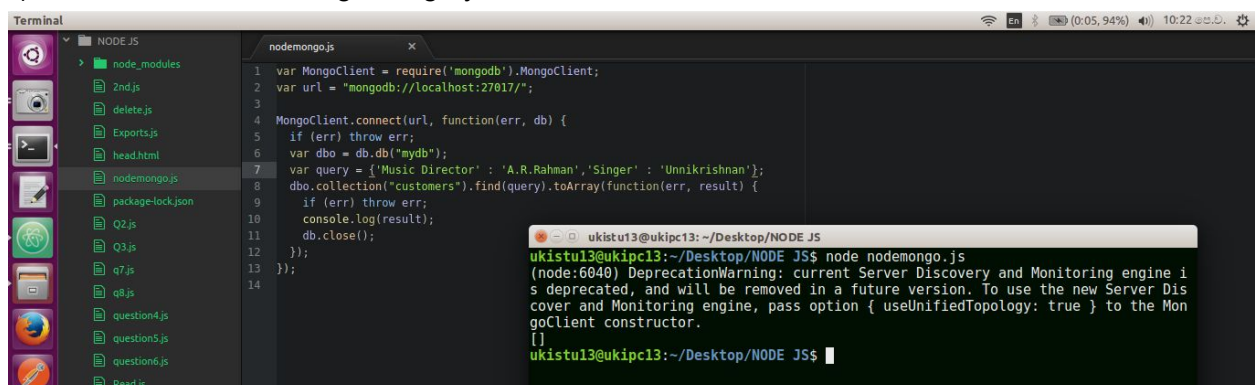
5) List A.R.Rahman's songs.



```
ukistul3@ukipc13:~/Desktop/NODE_JS$ node nodemongo.js
(node:4635) DeprecationWarning: current Server Discovery and Monitoring engine is deprecated, and will be removed in a future version. To use the new Server Discovery and Monitoring engine, pass option { useUnifiedTopology: true } to the MongoClient constructor.
[ { id: 5e315d2bd97ff1015fc781a4,
  'Song Name': 'Thaniye Thananthaniye',
  Film: 'Rhythm',
  'Music Director': 'A.R.Rahman',
  Singer: 'Shankar mahadevan' },
  { id: 5e315d2bd97ff1015fc781a5,
  'Song Name': 'Evano Oruvan',
  Film: 'Alai Payuthe',
  'Music Director': 'A.R.Rahman',
  Singer: 'Swarnalatha' },
  { id: 5e315d2bd97ff1015fc781a7,
  'Song Name': 'Vennilavae Vennilavae Vinnaithaandi',
  Film: 'Minsara Kanavu',
  'Music Director': 'A.R.Rahman',
  Singer: [ 'Hariharan', 'Sadhana Sargam' ] } ]
ukistul3@ukipc13:~/Desktop/NODE_JS$
```

```
nodemongo.js
1 var MongoClient = require('mongodb').MongoClient;
2 var url = "mongodb://localhost:27017/";
3
4 MongoClient.connect(url, function(err, db) {
5   if (err) throw err;
6   var dbo = db.db("music");
7   dbo.collection("songdetails").find({'Music Director' : 'A.R.Rahman'}).toArray(function(err, result) {
8     if (err) throw err;
9     console.log(result);
10    db.close();
11  });
12 });
13
```

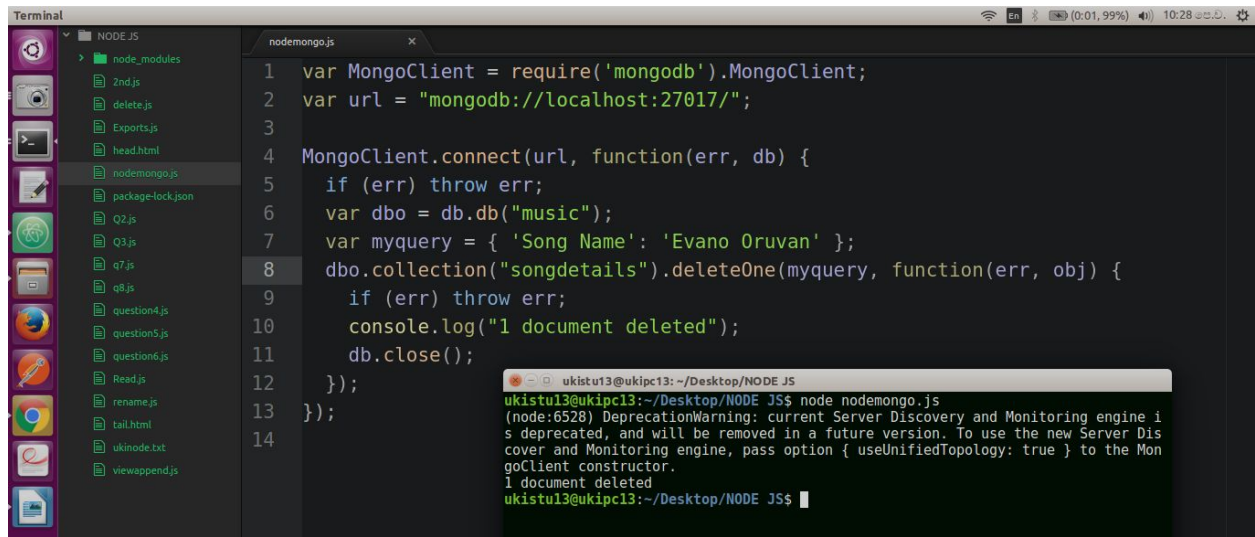
6) List A.R.Rahman's songs sung by Unnikrishnan.



```
Terminal
ukistul3@ukipc13:~/Desktop/NODE_JS$ node nodemongo.js
(node:6040) DeprecationWarning: current Server Discovery and Monitoring engine is deprecated, and will be removed in a future version. To use the new Server Discovery and Monitoring engine, pass option { useUnifiedTopology: true } to the MongoClient constructor.
[]
ukistul3@ukipc13:~/Desktop/NODE_JS$
```

```
nodemongo.js
1 var MongoClient = require('mongodb').MongoClient;
2 var url = "mongodb://localhost:27017/";
3
4 MongoClient.connect(url, function(err, db) {
5   if (err) throw err;
6   var dbo = db.db("mydb");
7   var query = {'Music Director' : 'A.R.Rahman', 'Singer' : 'Unnikrishnan'};
8   dbo.collection("customers").find(query).toArray(function(err, result) {
9     if (err) throw err;
10    console.log(result);
11    db.close();
12  });
13 });
14
```

7) Delete the song which you don't like.

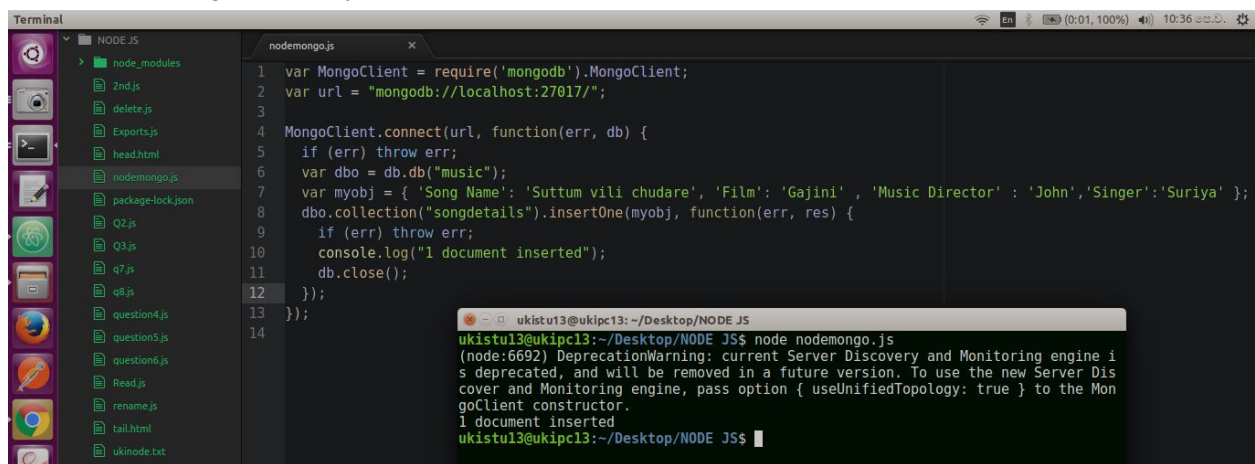


The terminal window shows a Node.js script named `nodemongo.js` that connects to a MongoDB instance at `localhost:27017`. It uses the `mongoose` library to connect and then deletes a document from the `music` database collection named `songdetails` where the `Song Name` is `'Evano Oruvan'`. The script logs `"1 document deleted"` to the console. The output shows the command being executed and the result.

```
1 var MongoClient = require('mongodb').MongoClient;
2 var url = "mongodb://localhost:27017/";
3
4 MongoClient.connect(url, function(err, db) {
5   if (err) throw err;
6   var dbo = db.db("music");
7   var myquery = { 'Song Name': 'Evano Oruvan' };
8   dbo.collection("songdetails").deleteOne(myquery, function(err, obj) {
9     if (err) throw err;
10    console.log("1 document deleted");
11    db.close();
12  });
13 });
14
```

ukistul3@ukipc13: ~/Desktop/NODE JS  
ukistul3@ukipc13:~/Desktop/NODE JS\$ node nodemongo.js  
(node:6528) DeprecationWarning: current Server Discovery and Monitoring engine is deprecated, and will be removed in a future version. To use the new Server Discovery and Monitoring engine, pass option { useUnifiedTopology: true } to the MongoClient constructor.  
1 document deleted  
ukistul3@ukipc13:~/Desktop/NODE JS\$

8) Add new song which is your favourite.

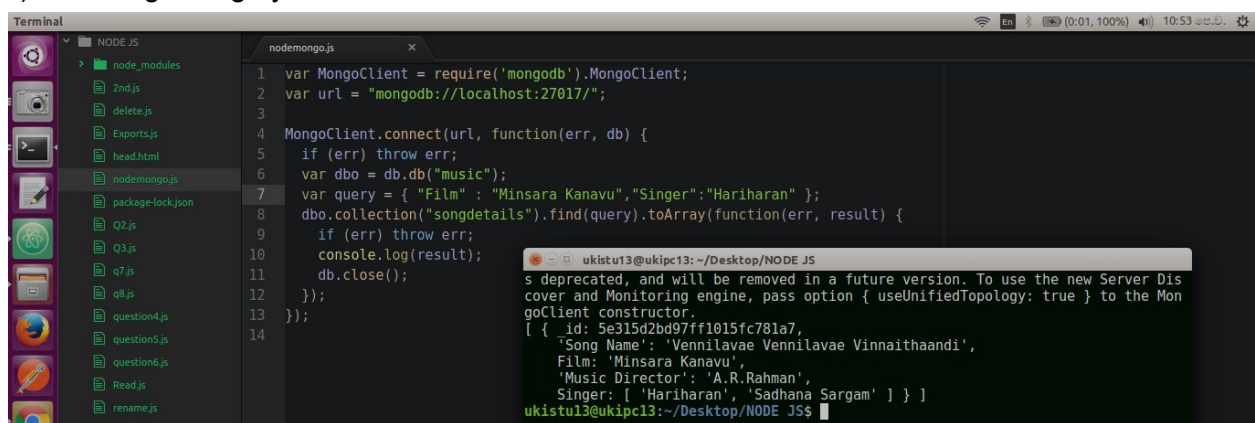


The terminal window shows a Node.js script named `nodemongo.js` that connects to a MongoDB instance at `localhost:27017`. It uses the `mongoose` library to connect and then inserts a new document into the `music` database collection named `songdetails`. The document contains the `Song Name` `'Suttum vili chudare'`, `Film` `'Gajini'`, `Music Director` `'John'`, and `Singer` `'Suriya'`. The script logs `"1 document inserted"` to the console. The output shows the command being executed and the result.

```
1 var MongoClient = require('mongodb').MongoClient;
2 var url = "mongodb://localhost:27017/";
3
4 MongoClient.connect(url, function(err, db) {
5   if (err) throw err;
6   var dbo = db.db("music");
7   var myobj = { 'Song Name': 'Suttum vili chudare', 'Film': 'Gajini', 'Music Director': 'John', 'Singer': 'Suriya' };
8   dbo.collection("songdetails").insertOne(myobj, function(err, res) {
9     if (err) throw err;
10    console.log("1 document inserted");
11    db.close();
12  });
13 });
14
```

ukistul3@ukipc13: ~/Desktop/NODE JS  
ukistul3@ukipc13:~/Desktop/NODE JS\$ node nodemongo.js  
(node:6692) DeprecationWarning: current Server Discovery and Monitoring engine is deprecated, and will be removed in a future version. To use the new Server Discovery and Monitoring engine, pass option { useUnifiedTopology: true } to the MongoClient constructor.  
1 document inserted  
ukistul3@ukipc13:~/Desktop/NODE JS\$

9) List Songs sung by Hariharan from Minsara kanavu film.



The terminal window shows a Node.js script named `nodemongo.js` that connects to a MongoDB instance at `localhost:27017`. It uses the `mongoose` library to connect and then queries the `music` database collection named `songdetails` for documents where the `Film` is `'Minsara Kanavu'` and the `Singer` is `'Hariharan'`. The script logs the result to the console. The output shows the command being executed and the result.

```
1 var MongoClient = require('mongodb').MongoClient;
2 var url = "mongodb://localhost:27017/";
3
4 MongoClient.connect(url, function(err, db) {
5   if (err) throw err;
6   var dbo = db.db("music");
7   var query = { "Film" : "Minsara Kanavu", "Singer": "Hariharan" };
8   dbo.collection("songdetails").find(query).toArray(function(err, result) {
9     if (err) throw err;
10    console.log(result);
11    db.close();
12  });
13 });
14
```

ukistul3@ukipc13: ~/Desktop/NODE JS  
s deprecated, and will be removed in a future version. To use the new Server Discovery and Monitoring engine, pass option { useUnifiedTopology: true } to the MongoClient constructor.  
[ { id: 5e315d2bd97ff1015fc781a7, 'Song Name': 'Vennilavae Vennilavae Vinnaitaandi', Film: 'Minsara Kanavu', 'Music Director': 'A.R.Rahman', Singer: [ 'Hariharan', 'Sadhana Sargam' ] } ]  
ukistul3@ukipc13:~/Desktop/NODE JS\$

10) List out the singers' names in your document.



Terminal

NODE JS

node\_modules

2nd.js

delete.js

Exports.js

head.html

nodemongo.js

package-lock.json

Q2.js

Q3.js

q7.js

q8.js

question4.js

question5.js

question6.js

Read.js

rename.js

tail.html

ukinode.txt

viewappend.js

nodemongo.js

```
1 var MongoClient = require('mongodb').MongoClient;
2 var url = "mongodb://localhost:27017/";
3
4 MongoClient.connect(url, function(err, db) {
5   if (err) throw err;
6   var dbo = db.db("music");
7   dbo.collection("songdetails").find({}, { projection: { _id: 0, Singer: 1 } }).toArray(function(err, result) {
8     if (err) throw err;
9     console.log(result);
10    db.close();
11  });
12 });
13
```

ukistu13@ukipc13: ~/Desktop/NODE JS

ukistu13@ukipc13:~/Desktop/NODE JS\$ node nodemongo.js  
(node:7783) DeprecationWarning: current Server Discovery and Monitoring engine is deprecated, and will be removed in a future version. To use the new Server Discovery and Monitoring engine, pass option { useUnifiedTopology: true } to the MongoClient constructor.  
[ { Singer: 'Shankar mahadevan' },  
 { Singer: [ 'Unnikrishnan', 'Anuradha Sriram' ] },  
 { Singer: [ 'Hariharan', 'Sadhana Sargam' ] },  
 { Singer: 'Hariharan' },  
 { Singer: 'Suriya' } ]  
ukistu13@ukipc13:~/Desktop/NODE JS\$

+ x nodemongo.js 7:72

LF UTF-8 Babel master +13 2 updates