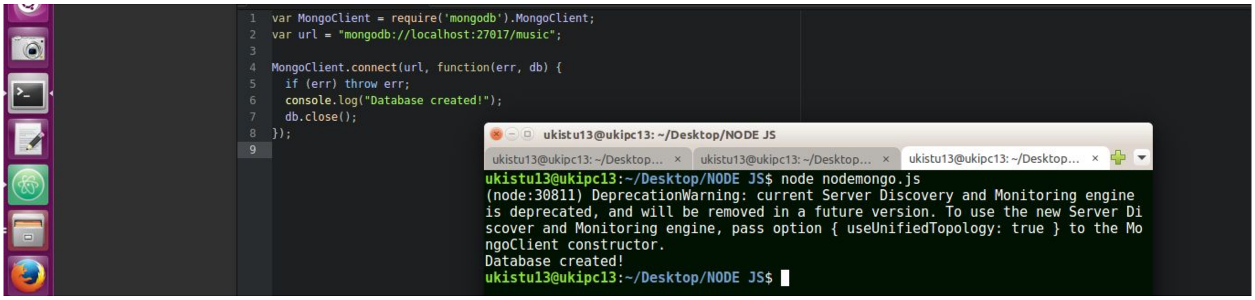
# Node.js Exercise 2
# Node.js MongoDB

01. Complete mongodbex1 with the mongoclient object you just created in Node.js

1) Create a Database called music.



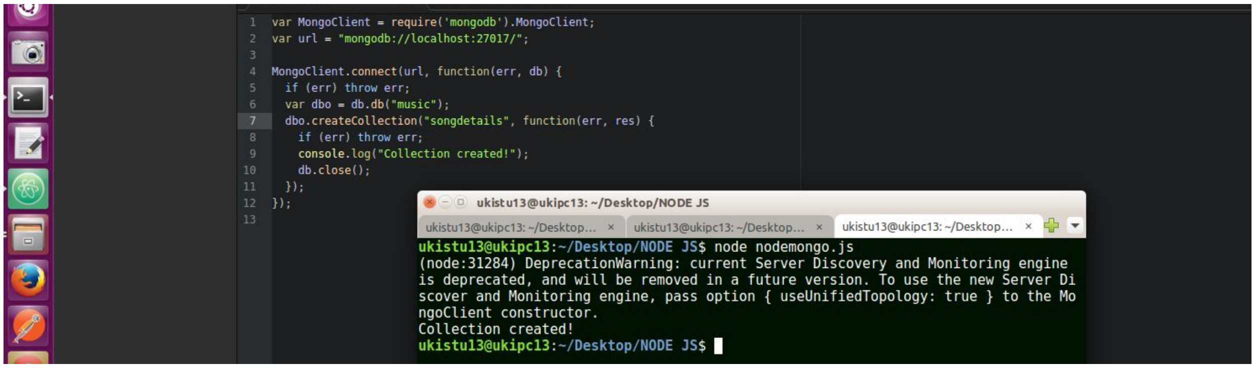
```
var MongoClient = require('mongodb').MongoClient;
var url = "mongodb://localhost:27017/music";

MongoClient.connect(url, function(err, db) {
  if (err) throw err;
  console.log("Database created!");
  db.close();
});
```

ukistu13@ukipc13: ~/Desktop/NODE JS
ukistu13@ukipc13: ~/Desktop...
ukistu13@ukipc13: ~/Desktop...
ukistu13@ukipc13: ~/Desktop...

```
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.



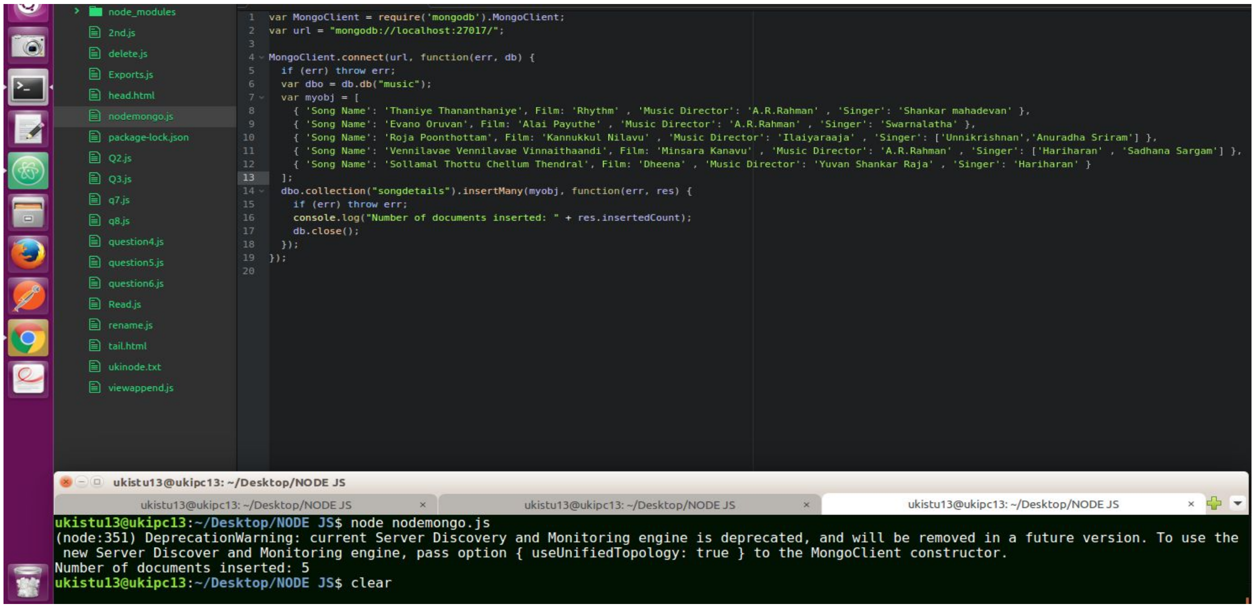
```
var MongoClient = require('mongodb').MongoClient;
var url = "mongodb://localhost:27017/";

MongoClient.connect(url, function(err, db) {
  if (err) throw err;
  var dbo = db.db("music");
  dbo.createCollection("songdetails", function(err, res) {
    if (err) throw err;
    console.log("Collection created!");
    db.close();
  });
});
```

ukistu13@ukipc13: ~/Desktop/NODE JS
ukistu13@ukipc13: ~/Desktop...
ukistu13@ukipc13: ~/Desktop...
ukistu13@ukipc13: ~/Desktop...

```
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.

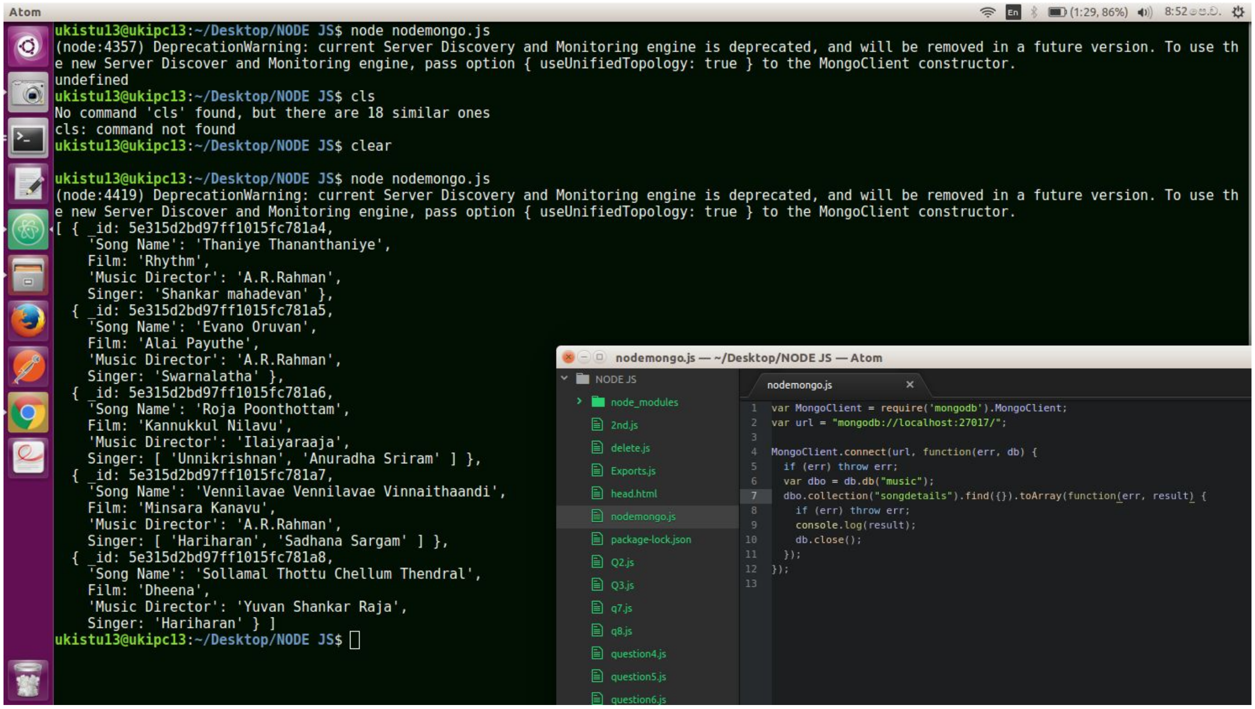


```
var MongoClient = require('mongodb').MongoClient;
var url = "mongodb://localhost:27017/";

MongoClient.connect(url, function(err, db) {
  if (err) throw err;
  var dbo = db.db("music");
  var myobj = [
    { 'Song Name': 'Thaniye Thananthaniye', Film: 'Rhythm' , 'Music Director': 'A.R.Rahman' , 'Singer': 'Shankar mahadevan' },
    { 'Song Name': 'Evano Oruvan', Film: 'Alai Payuthe' , 'Music Director': 'A.R.Rahman' , 'Singer': 'Swarnalatha' },
    { 'Song Name': 'Roja Poonthottam', Film: 'Kannukkul Nilavu' , 'Music Director': 'Ilaiyaraaja' , 'Singer': ['Unnikrishnan','Anuradha Sriram'] },
    { 'Song Name': 'Vennilavae Vennilavae Vinnaithaandi', Film: 'Minsara Kanavu' , 'Music Director': 'A.R.Rahman' , 'Singer': ['Hariharan' , 'Sadhana Sargam'] },
    { 'Song Name': 'Sollamal Thottu Chellum Thendral', Film: 'Dheena' , 'Music Director': 'Yuvan Shankar Raja' , 'Singer': 'Hariharan' }
  ];
  dbo.collection("songdetails").insertMany(myobj, function(err, res) {
    if (err) throw err;
    console.log("Number of documents inserted: " + res.insertedCount);
    db.close();
  });
});
```

```
ukistu13@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
ukistu13@ukipc13:~/Desktop/NODE JS$ clear
```

4) List all documents created.



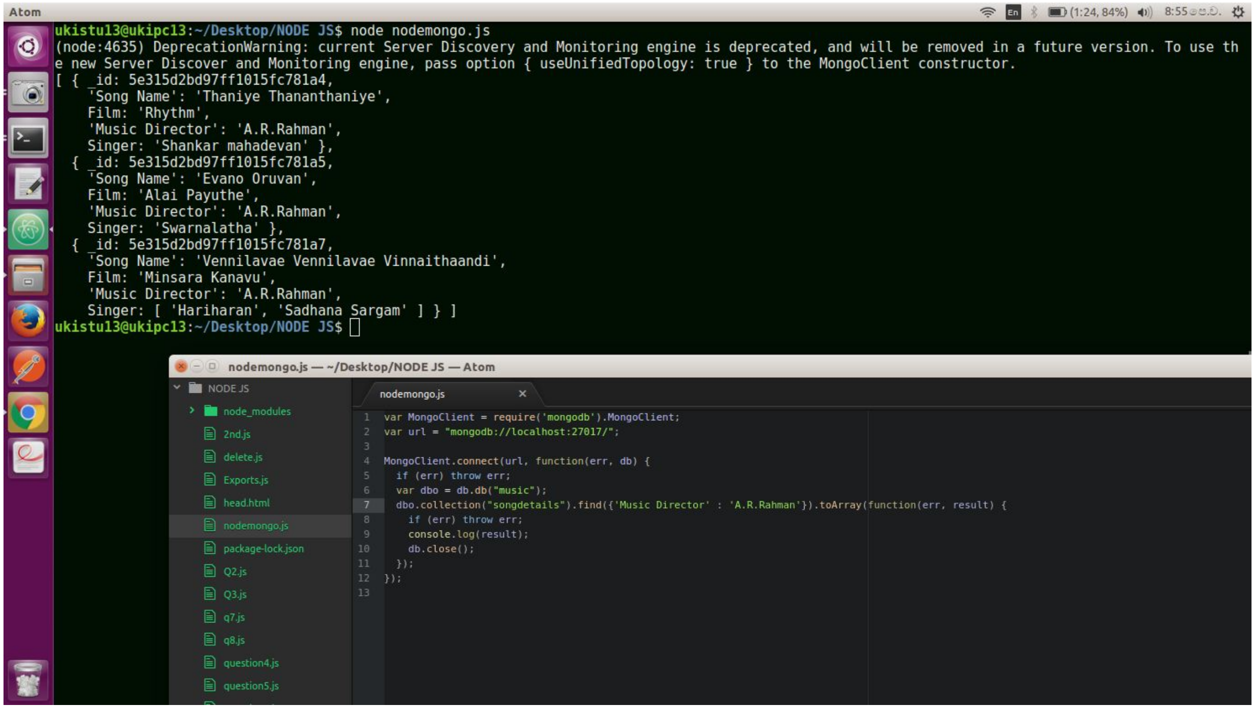
```
ukistu13@ukipc13:~/Desktop/NODE JS$ node nodemongo.js
(node:4357) DeprecationWarning: current Server Discovery and Monitoring engine is deprecated, and will be removed in a future version. To use th
e new Server Discover and Monitoring engine, pass option { useUnifiedTopology: true } to the MongoClient constructor.
undefined
ukistu13@ukipc13:~/Desktop/NODE JS$ cls
No command 'cls' found, but there are 18 similar ones
cls: command not found
ukistu13@ukipc13:~/Desktop/NODE JS$ clear

ukistu13@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 th
e 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': 'Roja Poonthottam',
    Film: 'Kannukkul Nilavu',
    'Music Director': 'Ilaiyaraaja',
    Singer: [ 'Unnikrishnan', 'Anuradha Sriram' ] },
  { _id: 5e315d2bd97ff1015fc781a7,
    'Song Name': 'Vennilavae Vennilavae Vinnaithaandi',
    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' } ]
ukistu13@ukipc13:~/Desktop/NODE JS$
```

```
var MongoClient = require('mongodb').MongoClient;
var url = "mongodb://localhost:27017/";

MongoClient.connect(url, function(err, db) {
  if (err) throw err;
  var dbo = db.db("music");
  dbo.collection("songdetails").find({}).toArray(function(err, result) {
    if (err) throw err;
    console.log(result);
    db.close();
  });
});
```

5) List A.R.Rahman’s songs.



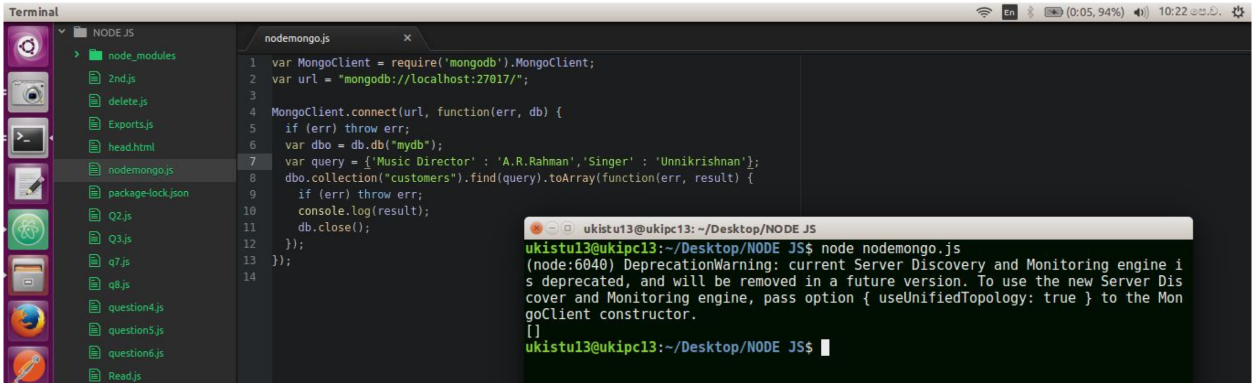
```
ukistu13@ukipc13:~/Desktop/NODE JS$ node nodemongo.js
(node:4635) DeprecationWarning: current Server Discovery and Monitoring engine is deprecated, and will be removed in a future version. To use th
e new Server Discover and Monitoring engine, pass option { useUnifiedTopology: true } to the MongoClient 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: 5e315d2bd97ff1015fc781a7,
    'Song Name': 'Vennilavae Vennilavae Vinnaithaandi',
    Film: 'Minsara Kanavu',
    'Music Director': 'A.R.Rahman',
    Singer: [ 'Hariharan', 'Sadhana Sargam' ] } ]
ukistu13@ukipc13:~/Desktop/NODE JS$
```

```
var MongoClient = require('mongodb').MongoClient;
var url = "mongodb://localhost:27017/";

MongoClient.connect(url, function(err, db) {
  if (err) throw err;
  var dbo = db.db("music");
  dbo.collection("songdetails").find({'Music Director' : 'A.R.Rahman'}).toArray(function(err, result) {
    if (err) throw err;
    console.log(result);
    db.close();
  });
});

```

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



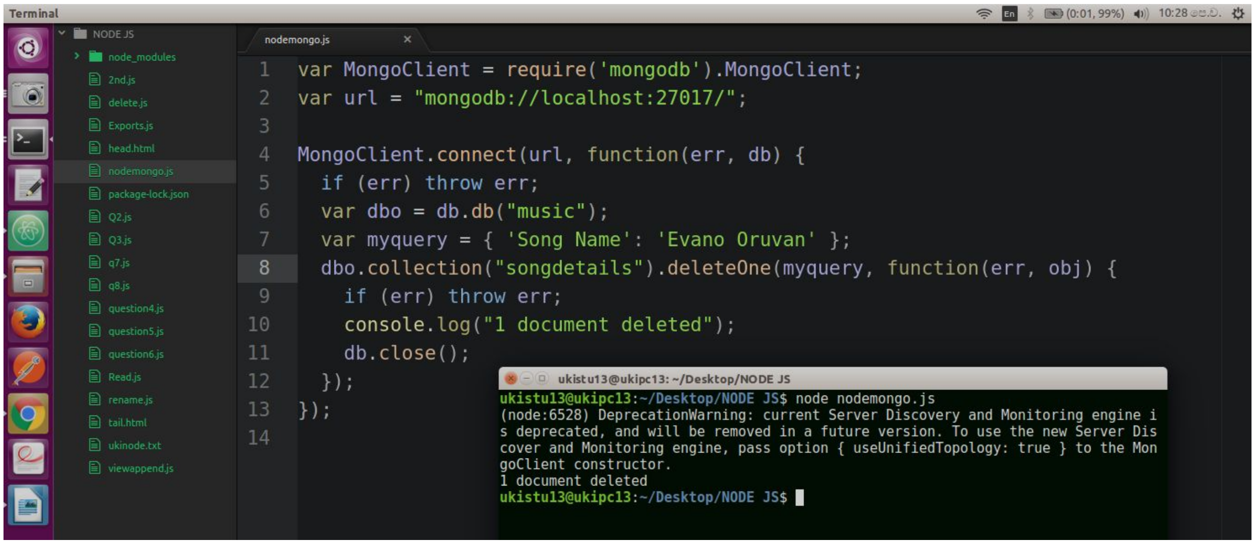
```
var MongoClient = require('mongodb').MongoClient;
var url = "mongodb://localhost:27017/";

MongoClient.connect(url, function(err, db) {
  if (err) throw err;
  var dbo = db.db("mydb");
  var query = {'Music Director' : 'A.R.Rahman','Singer' : 'Unnikrishnan'};
  dbo.collection("customers").find(query).toArray(function(err, result) {
    if (err) throw err;
    console.log(result);
    db.close();
  });
});

```

```
ukistu13@ukipc13:~/Desktop/NODE JS$ node nodemongo.js
(node:6040) DeprecationWarning: current Server Discovery and Monitoring engine i
s deprecated, and will be removed in a future version. To use the new Server Dis
cover and Monitoring engine, pass option { useUnifiedTopology: true } to the Mon
goClient constructor.
[]
ukistu13@ukipc13:~/Desktop/NODE JS$
```

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

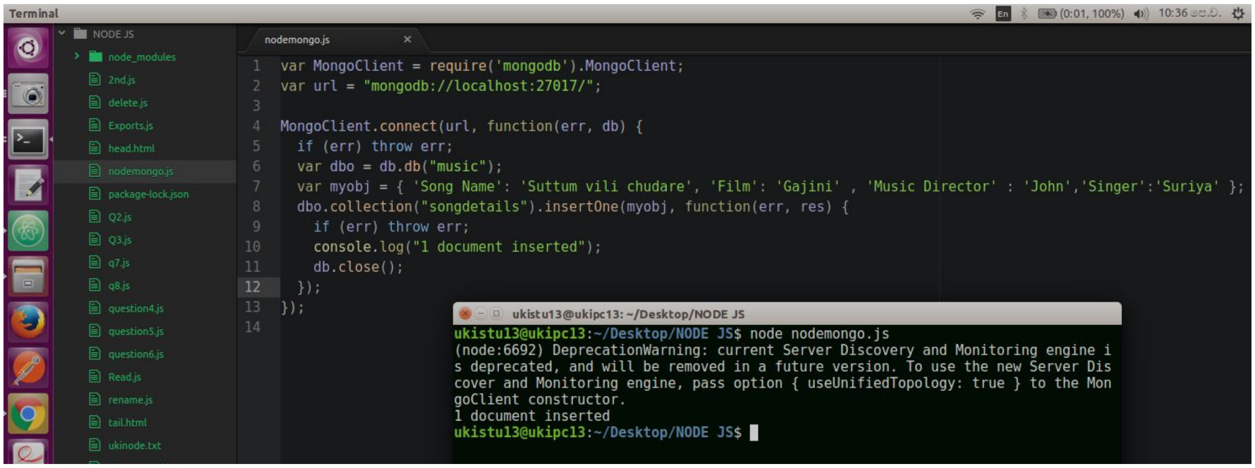


```
var MongoClient = require('mongodb').MongoClient;
var url = "mongodb://localhost:27017/";

MongoClient.connect(url, function(err, db) {
  if (err) throw err;
  var dbo = db.db("music");
  var myquery = { 'Song Name': 'Evano Oruvan' };
  dbo.collection("songdetails").deleteOne(myquery, function(err, obj) {
    if (err) throw err;
    console.log("1 document deleted");
    db.close();
  });
});
```

```
ukistu13@ukipc13:~/Desktop/NODE JS$ node nodemongo.js
(node:6528) DeprecationWarning: current Server Discovery and Monitoring engine i
s deprecated, and will be removed in a future version. To use the new Server Dis
cover and Monitoring engine, pass option { useUnifiedTopology: true } to the Mon
goClient constructor.
1 document deleted
ukistu13@ukipc13:~/Desktop/NODE JS$
```

8) Add new song which is your favourite.

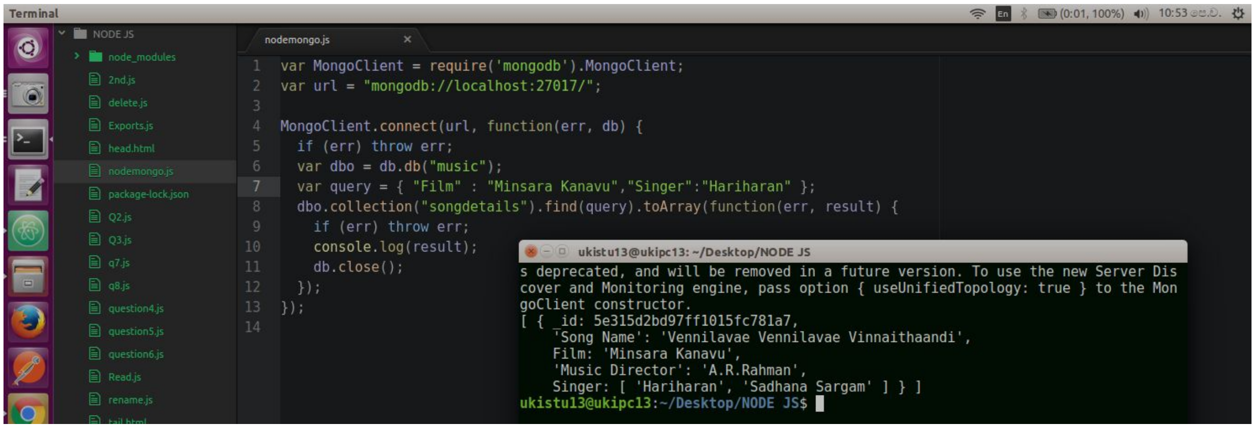


```
var MongoClient = require('mongodb').MongoClient;
var url = "mongodb://localhost:27017/";

MongoClient.connect(url, function(err, db) {
  if (err) throw err;
  var dbo = db.db("music");
  var myobj = { 'Song Name': 'Suttum vili chudare', 'Film': 'Gajini' , 'Music Director' : 'John','Singer':'Suriya' };
  dbo.collection("songdetails").insertOne(myobj, function(err, res) {
    if (err) throw err;
    console.log("1 document inserted");
    db.close();
  });
});
```

```
ukistu13@ukipc13:~/Desktop/NODE JS$ node nodemongo.js
(node:6692) DeprecationWarning: current Server Discovery and Monitoring engine i
s deprecated, and will be removed in a future version. To use the new Server Dis
cover and Monitoring engine, pass option { useUnifiedTopology: true } to the Mon
goClient constructor.
1 document inserted
ukistu13@ukipc13:~/Desktop/NODE JS$
```

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

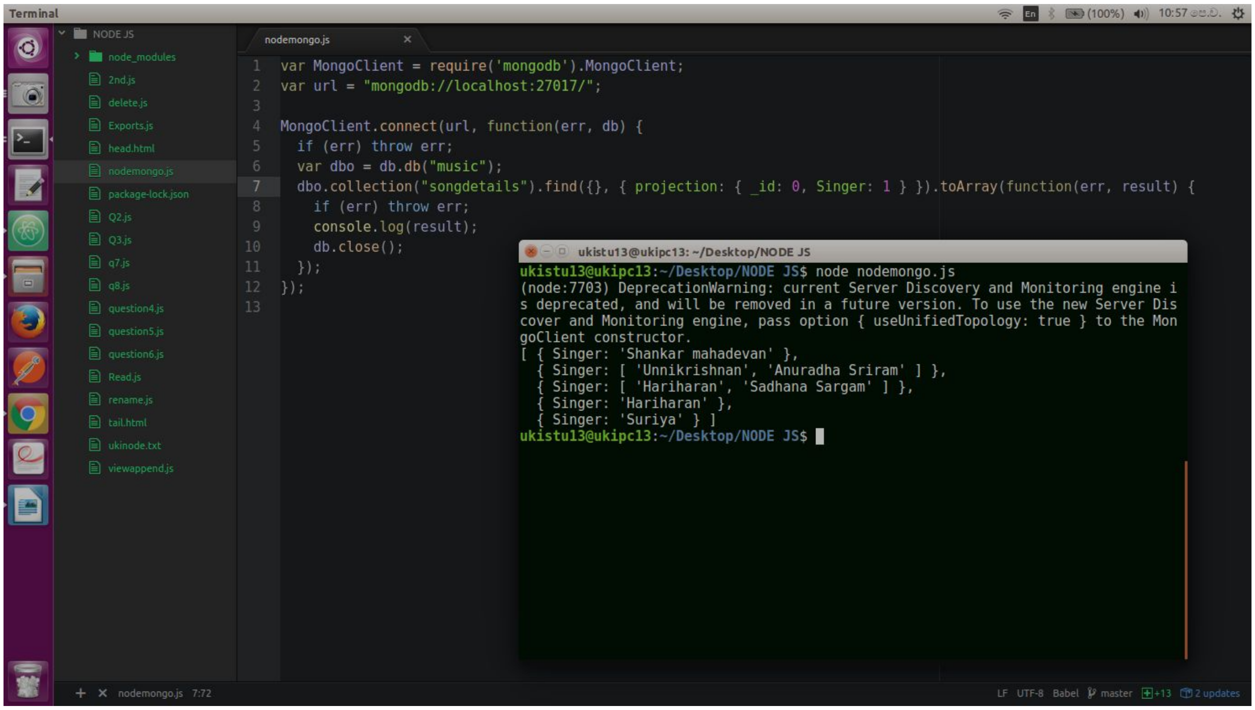


```
var MongoClient = require('mongodb').MongoClient;
var url = "mongodb://localhost:27017/";

MongoClient.connect(url, function(err, db) {
  if (err) throw err;
  var dbo = db.db("music");
  var query = { "Film" : "Minsara Kanavu","Singer":"Hariharan" };
  dbo.collection("songdetails").find(query).toArray(function(err, result) {
    if (err) throw err;
    console.log(result);
    db.close();
  });
});
```

```
s deprecated, and will be removed in a future version. To use the new Server Dis
cover and Monitoring engine, pass option { useUnifiedTopology: true } to the Mon
goClient constructor.
[ { _id: 5e315d2bd97ff1015fc781a7,
    'Song Name': 'Vennilavae Vennilavae Vinnaithaandi',
    Film: 'Minsara Kanavu',
    'Music Director': 'A.R.Rahman',
    Singer: [ 'Hariharan', 'Sadhana Sargam' ] } ]
ukistu13@ukipc13:~/Desktop/NODE JS$
```

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



```
var MongoClient = require('mongodb').MongoClient;
var url = "mongodb://localhost:27017/";

MongoClient.connect(url, function(err, db) {
  if (err) throw err;
  var dbo = db.db("music");
  dbo.collection("songdetails").find({}, { projection: { _id: 0, Singer: 1 } }).toArray(function(err, result) {
    if (err) throw err;
    console.log(result);
    db.close();
  });
});
```

```
ukistu13@ukipc13:~/Desktop/NODE JS$ node nodemongo.js
(node:7703) DeprecationWarning: current Server Discovery and Monitoring engine i
s deprecated, and will be removed in a future version. To use the new Server Dis
cover and Monitoring engine, pass option { useUnifiedTopology: true } to the Mon
goClient constructor.
[ { Singer: 'Shankar mahadevan' },
  { Singer: [ 'Unnikrishnan', 'Anuradha Sriram' ] },
  { Singer: [ 'Hariharan', 'Sadhana Sargam' ] },
  { Singer: 'Hariharan' },
  { Singer: 'Suriya' } ]
ukistu13@ukipc13:~/Desktop/NODE JS$
```