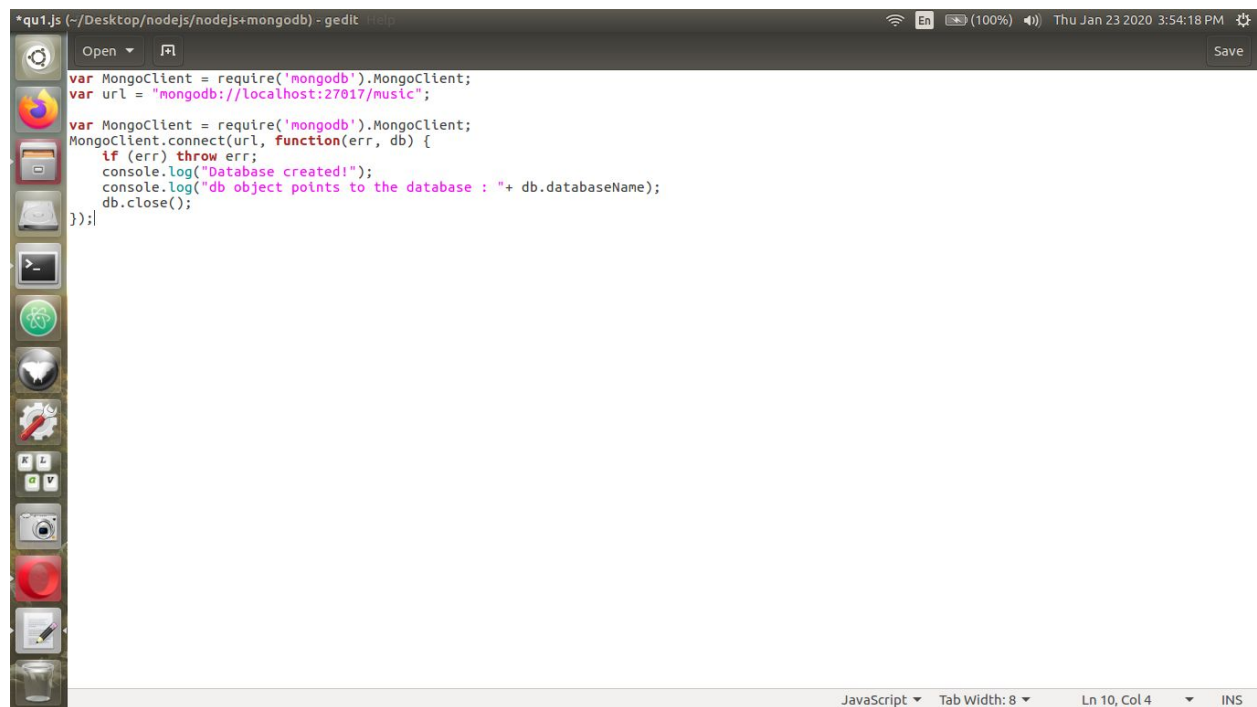
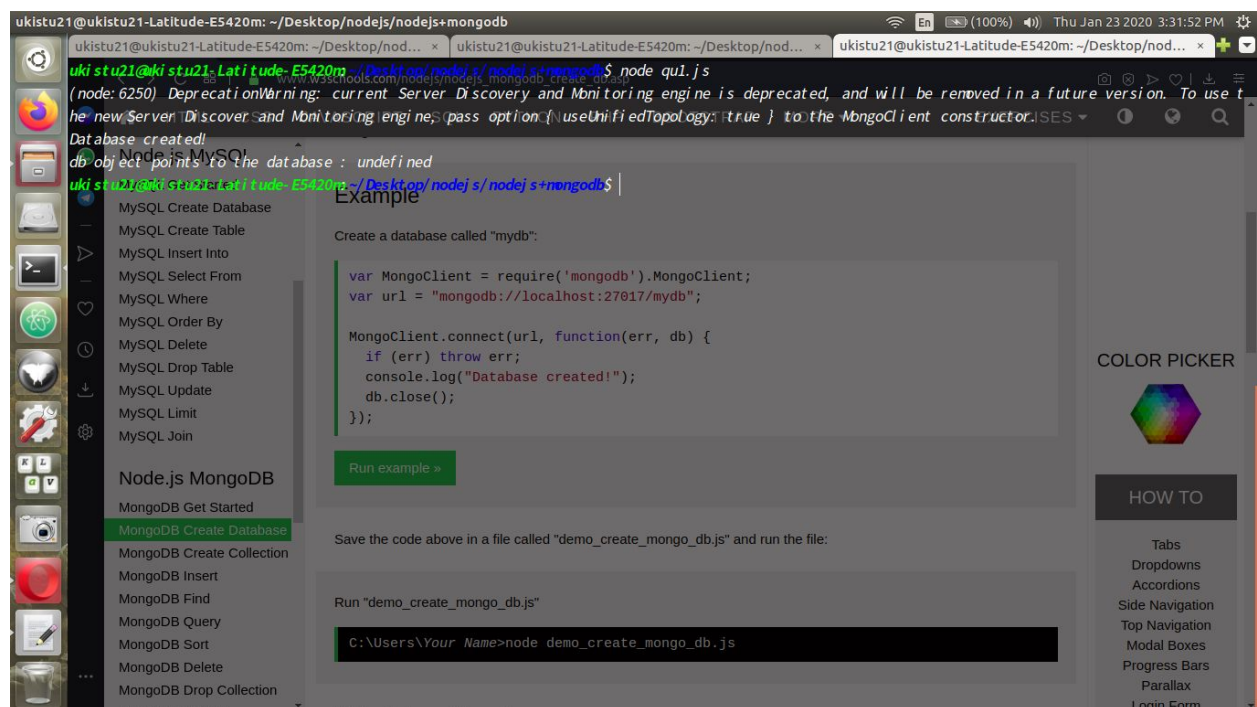


1) Create a Database called music.



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

var MongoClient = require('mongodb').MongoClient;
MongoClient.connect(url, function(err, db) {
  if (err) throw err;
  console.log("Database created!");
  console.log("db object points to the database : " + db.databaseName);
  db.close();
});
```



ukistu21@ukistu21-Latitude-E5420m: ~/Desktop/nodejs/nodejs+mongodb

ukistu21@ukistu21-Latitude-E5420m: ~/Desktop/nodejs/nodejs+mongodb\$ node qu1.js

(node:6250) 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.

Database created!

db object points to the database : undefined

ukistu21@ukistu21-Latitude-E5420m: ~/Desktop/nodejs/nodejs+mongodb\$

### Example

Create a database called "mydb":

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

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

Run example »

Save the code above in a file called "demo\_create\_mongo\_db.js" and run the file:

Run "demo\_create\_mongo\_db.js"

```
C:\Users\Your Name>node demo_create_mongo_db.js
```

### Node.js MongoDB

- MongoDB Get Started
- MongoDB Create Database
- MongoDB Create Collection
- MongoDB Insert
- MongoDB Find
- MongoDB Query
- MongoDB Sort
- MongoDB Delete
- MongoDB Drop Collection

### HOW TO

- Tabs
- Dropdowns
- Accordions
- Side Navigation
- Top Navigation
- Modal Boxes
- Progress Bars
- Parallax
- Login Form

2) Create a collection called songdetails.

```
qu1.js (~/.Desktop/nodejs/nodejs+mongodb) - gedit
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();
  });
});
```

```
ukistu21@ukistu21-Latitude-E5420m: ~/Desktop/nodejs/nodejs+mongodb
ukistu21@ukistu21-Latitude-E5420m: ~/Desktop/nodejs/nodejs+mongodb$ nano qu1.js
ukistu21@ukistu21-Latitude-E5420m: ~/Desktop/nodejs/nodejs+mongodb$ node qu1.js
(node:3445) 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.
Collection created!
ukistu21@ukistu21-Latitude-E5420m: ~/Desktop/nodejs/nodejs+mongodb$ show dbs
The program 'show' can be found in the following packages:
  mailutils-nh
  nrmixtures
Try: sudo apt install <selected package>
ukistu21@ukistu21-Latitude-E5420m: ~/Desktop/nodejs/nodejs+mongodb$
```

```
ukistu21@ukistu21-Latitude-E5420m: ~/Desktop/nodejs/nodejs+mongodb
2020-01-23T15:52:54.182+0530 I CONTROL [initandlisten] ** WARNING: Access control is not enabled for the database.
2020-01-23T15:52:54.182+0530 I CONTROL [initandlisten] ** Read and write access to data and configuration is unrestricted.
2020-01-23T15:52:54.205+0530 I CONTROL [initandlisten]
quit.js

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
music 0.000GB
person 0.000GB
test 0.000GB
uki6 0.000GB
> use music
switched to db music
> show collections
songdetails
> |
```

### 3) Create the above 5 song documents.

```
qu1.js (~/.Desktop/nodes/nodes+mongodb) - gedit
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 Payuthey', Music_Director: 'A.R.Rahman', Singer: 'Swarnalatha' },
    { Song_Name: 'Roja Poonthottam', Film: 'Kannukkul Nilavu', Music_Director: 'Kannukkul', Singer: ['Unnikrishnan', 'Anuradha Srilam'] },
    { 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();
  });
});
```

```
ukistu21@ukistu21-Latitude-E5420m: ~/.Desktop/nodes/nodes+mongodb
ukistu21@ukistu21-Latitude-E5420m: ~/.Desktop/nodes/nodes+mongodb$ node qu1.js
(node:4165) 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.
Number of documents inserted: 5
ukistu21@ukistu21-Latitude-E5420m: ~/.Desktop/nodes/nodes+mongodb$
```

#### 4) List all documents created.

```
ukistu21@ukistu21-Latitude-E5420m: ~/Desktop/nodejs/nodejs+mongodb
> db.songdetails.find().pretty()
var MongoClient = require('mongodb').MongoClient;
var url = "mongodb://localhost:27017/";
MongoClient.connect(url, function(err, db) {
  if (err) throw err;
  var dbo = db.collection("songdetails");
  var myobj = {
    "Song_Name": "Thaniye Thananthaniye",
    "Music_Director": "A.R.Rahman",
    "Film": "Rhythm",
    "Music_Director": "A.R.Rahman",
    "Singer": "Shankar mahadevan",
    "Song_Singer": "Shankar mahadevan",
    "Film": "Alai Payuthey",
    "Music_Director": "A.R.Rahman",
    "Singer": "Swarnalatha",
    "Song_Name": "Raja Poonthottam",
    "Film": "Kannukkul Nilavu",
    "Music_Director": "Kannukkul",
    "Singer": ["Unnikrishnan", "Anuradha Sritram"]],
    "Song_Name": "Vennilavae Vennilavae Vinnaitaandi",
    "Film": "Minsara Kanavu",
    "Music_Director": "A.R.Rahman",
    "Singer": ["Hariharan", "Sadhana", "argan"]],
    "Song_Name": "Sollalai Thottu Chellum Thendral",
    "Film": "Dheena",
    "Music_Director": "Yuvan Shankar Raja",
    "Singer": "Hariharan",
    "id": "5e29a6214ec33010456b1dbf"
  };
  dbo.insertMany(myobj, function(err, res) {
    if (err) throw err;
    console.log("Number of documents inserted: " + res.insertedCount);
    db.close();
  });
});
{
  "_id": "5e29a6214ec33010456b1dc0",
  "Song_Name": "Raja Poonthottam",
  "Film": "Kannukkul Nilavu",
  "Music_Director": "Kannukkul",
  "Singer": [
    "Unnikrishnan",
    "Anuradha Sritram"
  ]
}
{
  "_id": "5e29a6214ec33010456b1dc1",
  "Song_Name": "Vennilavae Vennilavae Vinnaitaandi",
  "Film": "Minsara Kanavu",
  "Music_Director": "A.R.Rahman",
  "Singer": ["Hariharan", "Sadhana", "argan"]
}
```



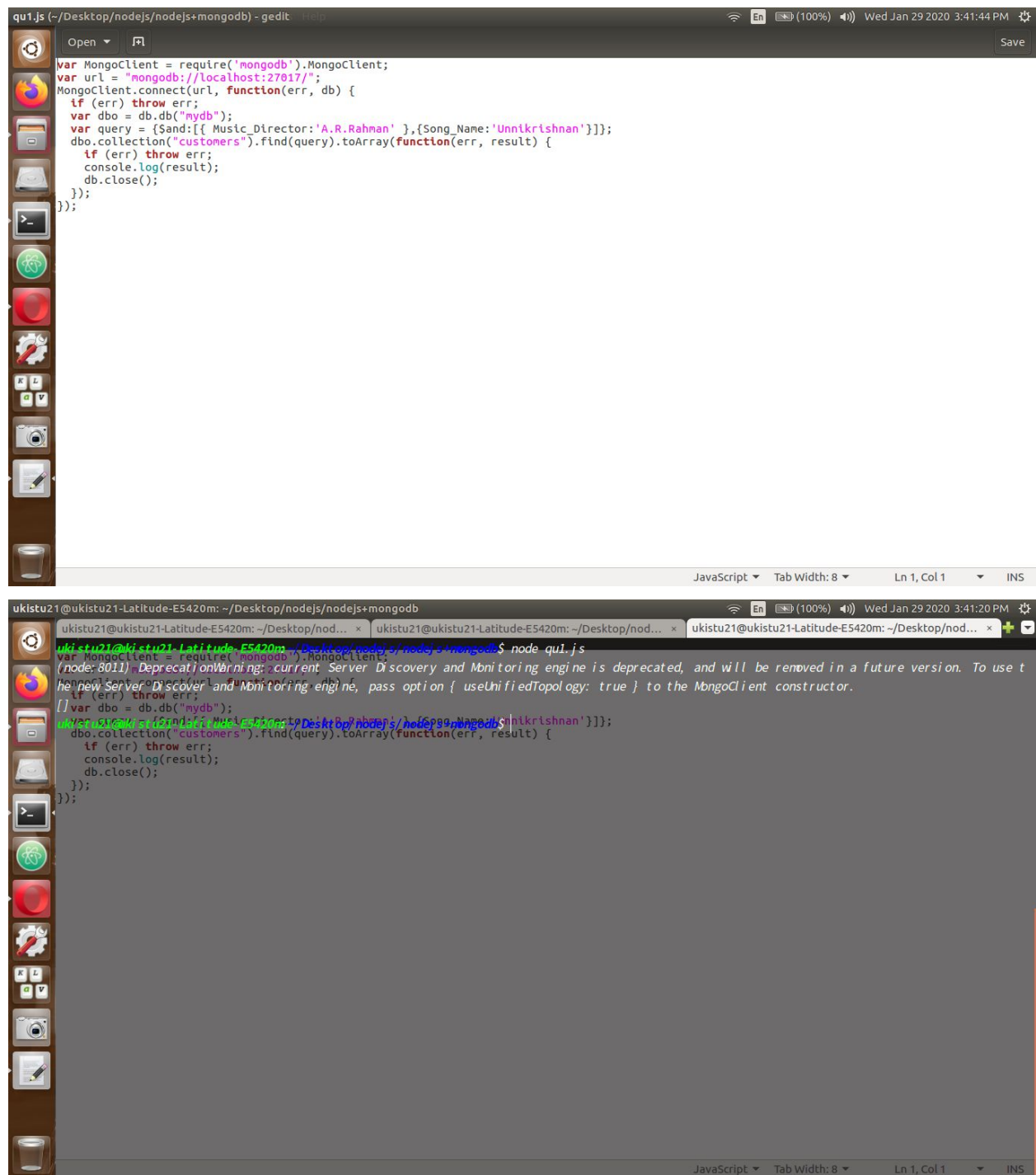
## 5) List A.R.Rahman's songs.

```
qu1.js (~/Desktop/nodejs/nodejs+mongodb) - gedit
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 = { Music_Director: "A.R.Rahman" };
  dbo.collection("songdetails").find(query).toArray(function(err, result) {
    if (err) throw err;
    console.log(result);
    db.close();
  });
});

ukistu21@ukistu21-Latitude-E5420m: ~/Desktop/nodejs/nodejs+mongodb
ukistu21@ukistu21-Latitude-E5420m: ~/Desktop/nodejs/nodejs+mongodb$ node qu1.js
(node:4165) 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.
Number of documents inserted: 5
ukistu21@ukistu21-Latitude-E5420m: ~/Desktop/nodejs/nodejs+mongodb$ node qu1.js
(node:6300) 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: '5e29a6214ec33010456b1dbf',
  Song_Name: 'Thani ye Thanant hani ye',
  Film: 'Rhythm',
  Music_Director: 'A. R. Rahman',
  Singer: 'Shankar mahadevan' },
  { _id: '5e29a6214ec33010456b1dbf',
  Song_Name: 'Evano Oruvan',
  Film: 'Alai Payuthey',
  Music_Director: 'A. R. Rahman',
  Singer: 'Svarnalatha' },
  { _id: '5e29a6214ec33010456b1dc1',
  Song_Name: 'Venni Lavae Venni Lavae Vinnai thaandi',
  Film: 'Mnsara Kanavu',
  Music_Director: 'A. R. Rahman',
  Singer: [ 'Hari haran', 'Sadhana Sargani' ] } ]
ukistu21@ukistu21-Latitude-E5420m: ~/Desktop/nodejs/nodejs+mongodb$
```

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



```
qui1.js (-~/Desktop/nodejs/nodejs+mongodb) - gedit
var MongoClient = require('mongodb').MongoClient;
var url = "mongodb://localhost:27017/";
MongoClient.connect(url, function(err, db) {
  if (err) throw err;
  var dbo = db.db("nydb");
  var query = {$and:[{ Music_Director:'A.R.Rahman' },{Song_Name:'Unnikrishnan'}]};
  dbo.collection("customers").find(query).toArray(function(err, result) {
    if (err) throw err;
    console.log(result);
    db.close();
  });
});

ukistu21@ukistu21-Latitude-E5420m: ~/Desktop/nodejs/nodejs+mongodb
$ node qui1.js
(node:8011) 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.
[[{"Song_Name": "Unnikrishnan", "Music_Director": "A.R.Rahman"}]]
```

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

```
qu1.js (~./Desktop/nodejs/nodejs+mongodb) - gedit
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 = { Film: 'Dheena' };
  dbo.collection("songdetails").deleteOne(myquery, function(err, obj) {
    if (err) throw err;
    console.log("1 document deleted");
    db.close();
  });
});
```

```
ukistu21@ukistu21-Latitude-E5420m: ~/Desktop/nodejs/nodejs+mongodb
ukistu21@ukistu21-Latitude-E5420m: ~/Desktop/nodejs/nodejs+mongodb$ node qu1.js
(node:5374) 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;
ukistu21@ukistu21-Latitude-E5420m: ~/Desktop/nodejs/nodejs+mongodb$ cls
No command 'cls' found, but there are 18 similar ones
cls: command not found
console.log("1 document deleted");
ukistu21@ukistu21-Latitude-E5420m: ~/Desktop/nodejs/nodejs+mongodb$ clear
ukistu21@ukistu21-Latitude-E5420m: ~/Desktop/nodejs/nodejs+mongodb$
```



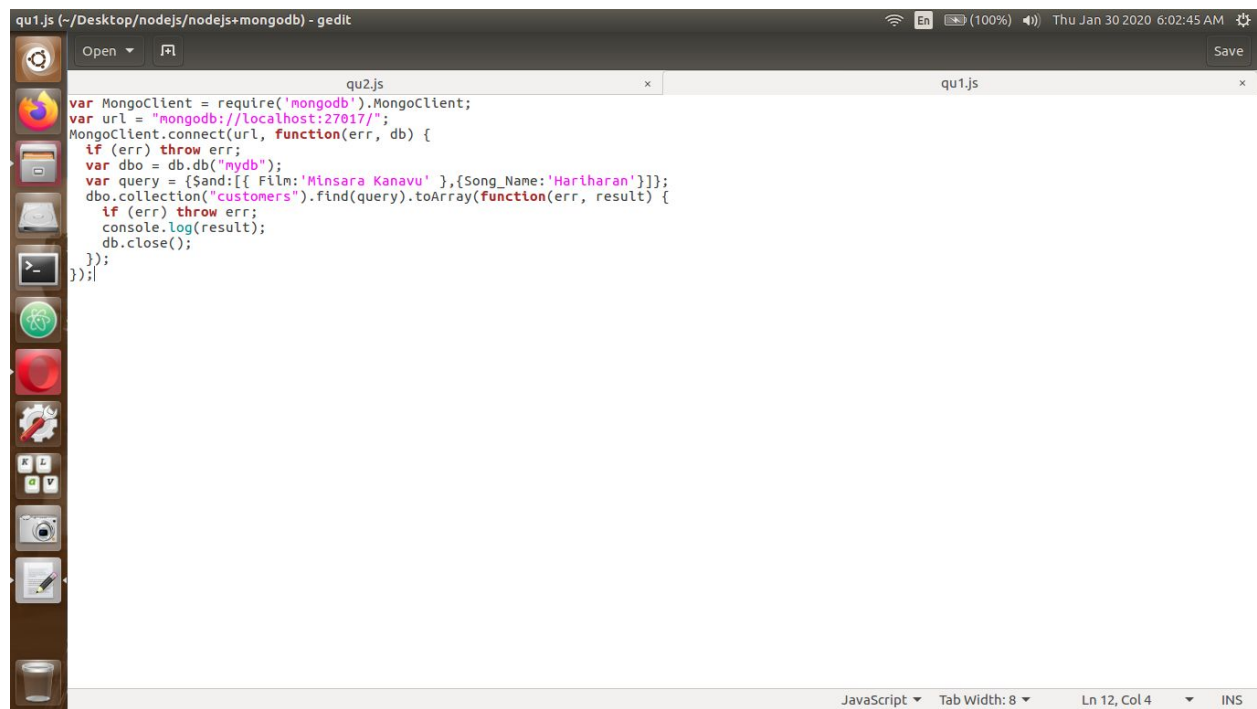
## 8) Add new song which is your favourite.

```
qu1.js (~/Desktop/nodejs/nodejs+mongodb) - gedit
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: 'Thaarane', Film: 'Kadaaram Kondaan', Music_Director: 'Ghibran', Singer: 'Sid Sriraam' };
  dbo.collection("songdetails").insertOne(myobj, function(err, res) {
    if (err) throw err;
    console.log("1 document inserted");
    db.close();
  });
});
```

```
ukistu21@ukistu21-Latitude-E5420m: ~/Desktop/nodejs/nodejs+mongodb
ukistu21@ukistu21-Latitude-E5420m: ~/Desktop/nodejs/nodejs+mongodb$ node qu1.js
(node:6986) 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
ukistu21@ukistu21-Latitude-E5420m: ~/Desktop/nodejs/nodejs+mongodb$
var myobj = { Song_Name: 'Thaarane', Film: 'Kadaaram Kondaan', Music_Director: 'Ghibran', Singer: 'Sid Sriraam' };
dbo.collection("songdetails").insertOne(myobj, function(err, res) {
  if (err) throw err;
  console.log("1 document inserted");
  db.close();
});
});
```

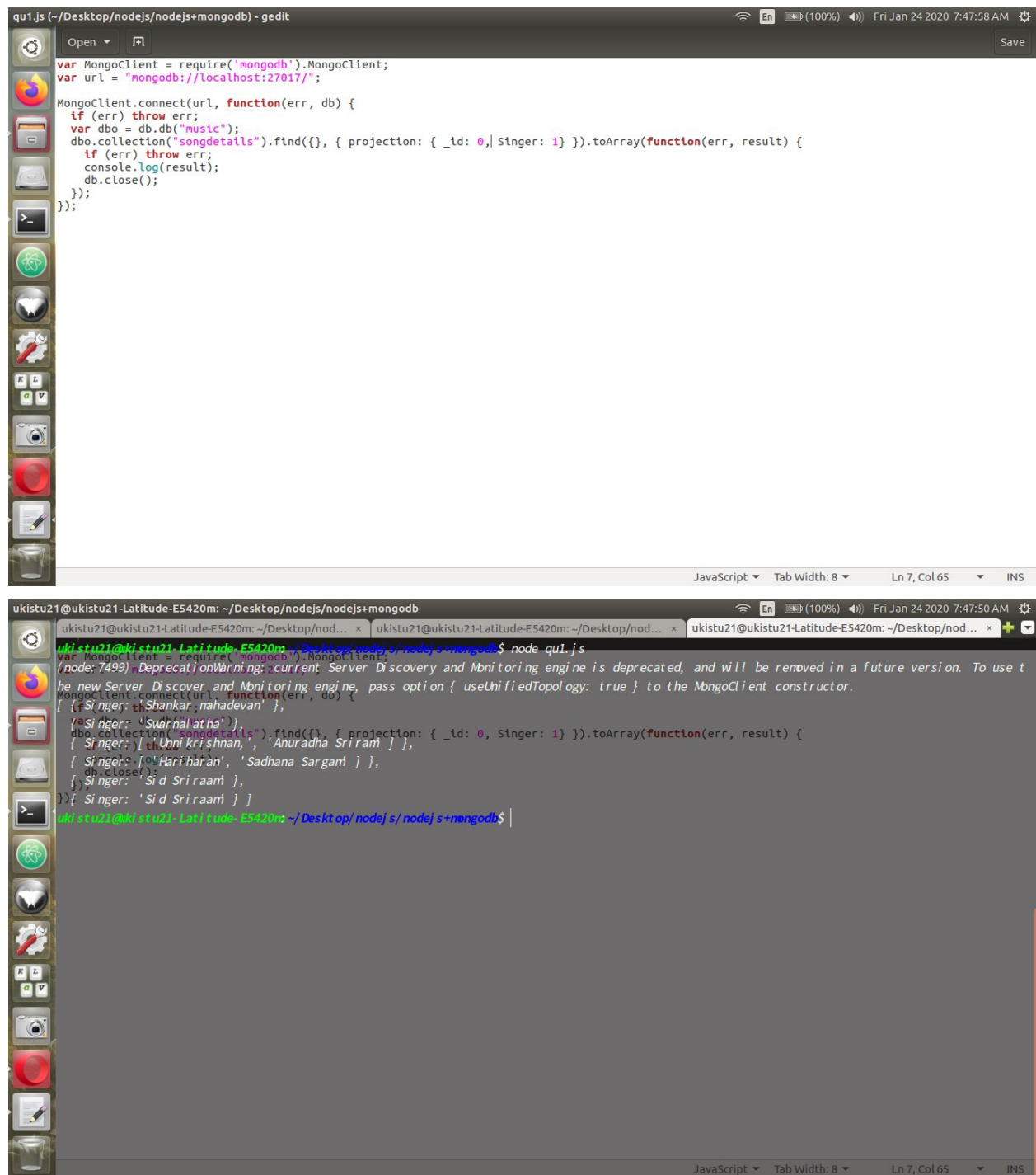
9) List Songs sung by Hariharan from Minsara kanavu film.



```
qu1.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("mydb");
  var query = {$and:[{ Film:'Minsara Kanavu' },{Song_Name:'Hariharan'}]};
  dbo.collection("customers").find(query).toArray(function(err, result) {
    if (err) throw err;
    console.log(result);
    db.close();
  });
});
```

JavaScript Tab Width: 8 Ln 12, Col 4 INS

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



The image shows a code editor window titled 'qui1.js (~/.Desktop/nodejs/nodejs+mongodb) - gedit' and a terminal window below it. The code in the editor connects to a MongoDB instance and retrieves singer names from a collection named '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.collection("songdetails").find({}, { projection: { _id: 0, Singer: 1 } }).toArray(function(err, result) {
    if (err) throw err;
    console.log(result);
    db.close();
  });
});
```

The terminal window shows the execution of the script. It displays a deprecation warning and the output of the query, which is an array of singer names: [ { Singer: 'Shankar mahadevan' }, { Singer: 'Svarnalatha' }, { Singer: 'Uthirakrishnan', 'Anuradha Sriram' }, { Singer: 'Hariharan', 'Sadhana Sargam' }, { Singer: 'Sid Sriaram' } ].

```
ukistu21@ukistu21-Latitude-E5420m: ~/.Desktop/nodejs/nodejs+mongodb
ukistu21@ukistu21-Latitude-E5420m: ~/.Desktop/nodejs/nodejs+mongodb$ node qui1.js
(node:7499) 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.
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();
  });
});
[ { Singer: 'Shankar mahadevan' },
  { Singer: 'Svarnalatha' },
  { Singer: 'Uthirakrishnan', 'Anuradha Sriram' },
  { Singer: 'Hariharan', 'Sadhana Sargam' },
  { Singer: 'Sid Sriaram' } ]
ukistu21@ukistu21-Latitude-E5420m: ~/.Desktop/nodejs/nodejs+mongodb$
```