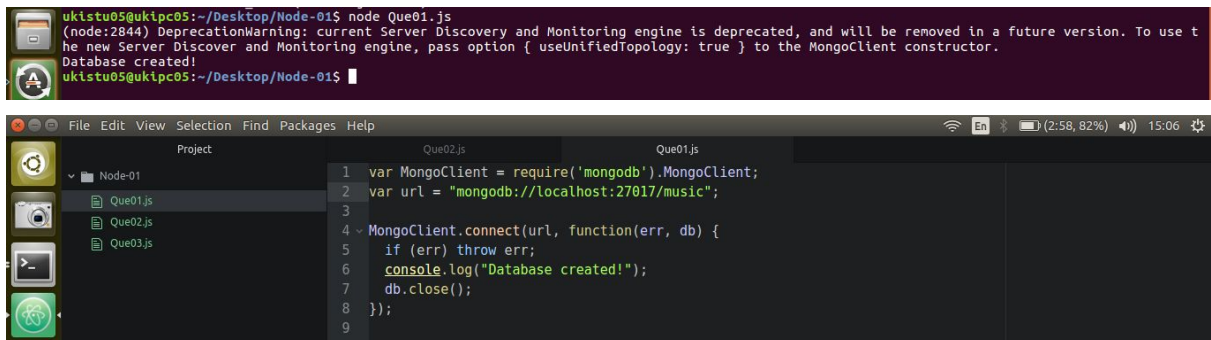


1) Create a Database called music.

```
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();
});
```

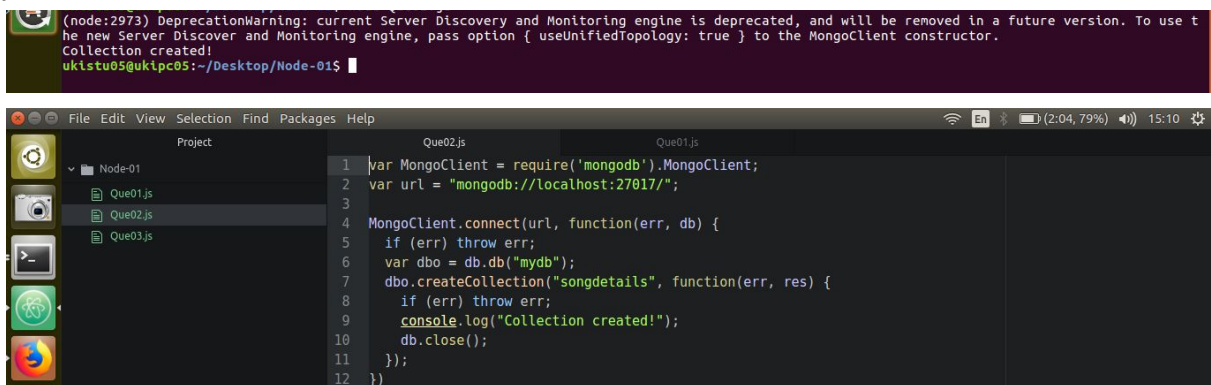


The screenshot shows a terminal window at the top with the command `node Que01.js` and its output: `(node:2844) DeprecationWarning: current Server Discovery and Monitoring engine is deprecated, and will be removed in a future version. To use the new Server Discovery and Monitoring engine, pass option { useUnifiedTopology: true } to the MongoClient constructor. Database created!`. Below the terminal is an IDE window showing the code for `Que01.js`, which matches the code provided in the previous blocks.

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();
  });
});
```

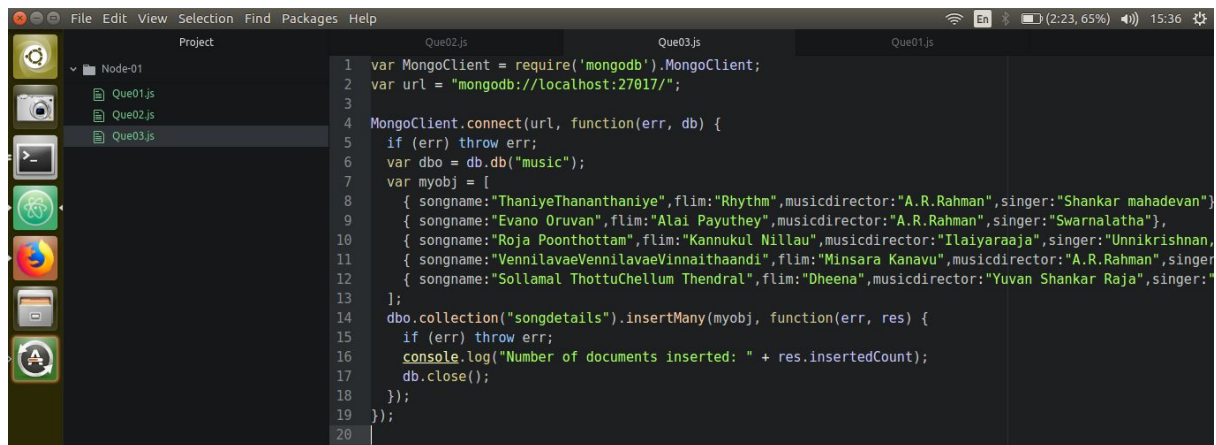


The screenshot shows a terminal window at the top with the command `node Que02.js` and its output: `(node:2973) DeprecationWarning: current Server Discovery and Monitoring engine is deprecated, and will be removed in a future version. To use the new Server Discovery and Monitoring engine, pass option { useUnifiedTopology: true } to the MongoClient constructor. Collection created!`. Below the terminal is an IDE window showing the code for `Que02.js`, which matches the code provided in the previous blocks.

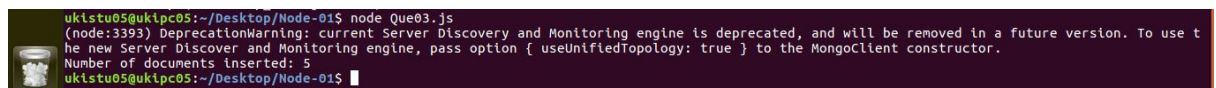
3) Create the above 5 song documents.

```
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 = [
    {
      songname:"ThaniyeThananthaniye",flim:"Rhythm",musicdirector:"A.R.Rahman",singer:"Shan
      kar mahadevan"},
    { songname:"Evano Oruvan",flim:"Alai
      Payuthey",musicdirector:"A.R.Rahman",singer:"Swarnalatha"},
    { songname:"Roja Poonthottam",flim:"Kannukul
      Nillau",musicdirector:"Ilaiyaraaja",singer:"Unnikrishnan,Anuradha Sriram"},
    { songname:"VennilavaeVennilavaeVinnaitaandi",flim:"Minsara
      Kanavu",musicdirector:"A.R.Rahman",singer:"Hariharan,Sadhana Sargam"},
    { songname:"Sollamal ThottuChellum Thendral",flim:"Dheena",musicdirector:"Yuva
      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();
  });
});
```

A screenshot of a code editor window with a dark theme. The editor shows a file named 'Que03.js' with the following JavaScript code:

```
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     { songname:"ThaniyeThananthaniye",flim:"Rhythm",musicdirector:"A.R.Rahman",singer:"Shankar mahadevan"},
9     { songname:"Evano Oruvan",flim:"Alai Payuthey",musicdirector:"A.R.Rahman",singer:"Swarnalatha"},
10    { songname:"Roja Poonthottam",flim:"Kannukul Nillau",musicdirector:"Ilaiyaraaja",singer:"Unnikrishnan,Anuradha Sriram"},
11    { songname:"VennilavaeVennilavaeVinnaitaandi",flim:"Minsara Kanavu",musicdirector:"A.R.Rahman",singer:"Hariharan,Sadhana Sargam"},
12    { songname:"Sollamal ThottuChellum Thendral",flim:"Dheena",musicdirector:"Yuva 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
```

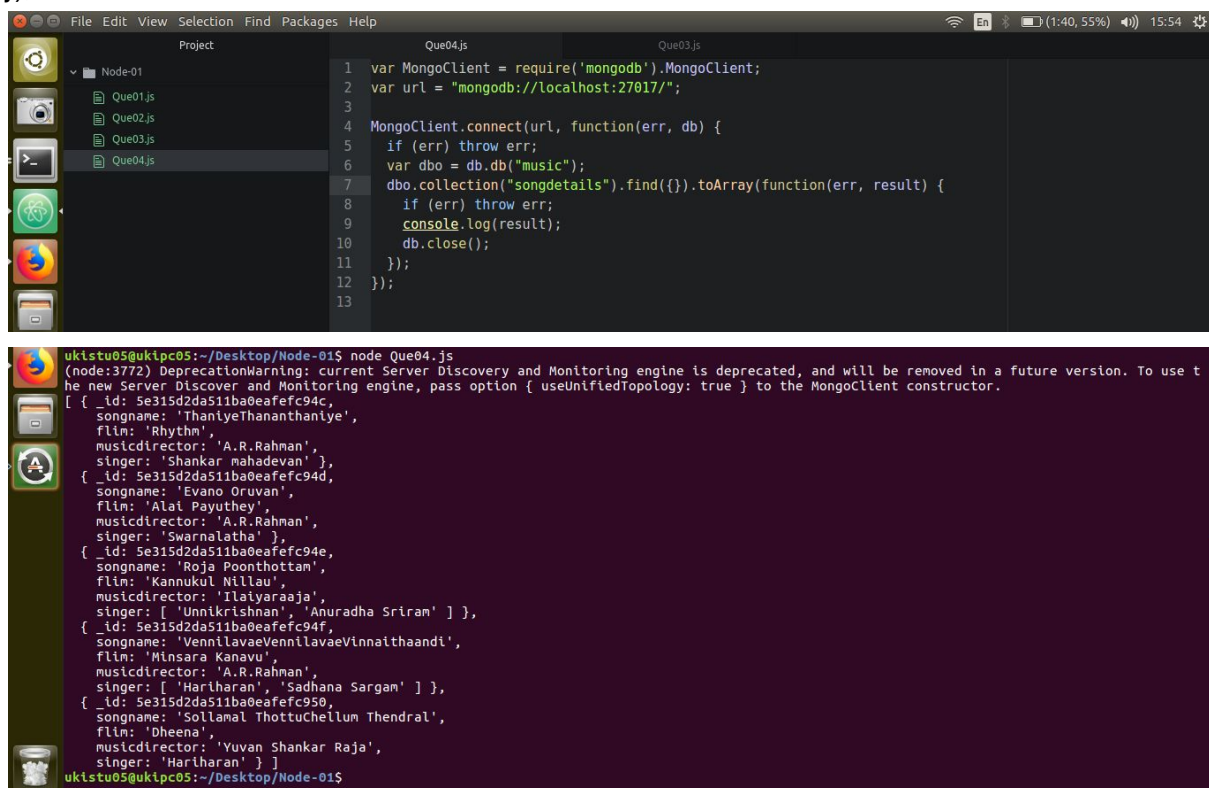
The left sidebar shows a project structure with 'Node-01' containing 'Que01.js', 'Que02.js', and 'Que03.js'. The top status bar shows 'En', battery level '(2:23, 65%)', and time '15:36'.A screenshot of a terminal window with a dark background. The prompt is 'uklstu05@uklpc05:~/Desktop/Node-01\$'. The command entered is 'node Que03.js'. The output shows a deprecation warning from MongoDB and the result of the insertMany operation:

```
(node:3393) 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
uklstu05@uklpc05:~/Desktop/Node-01$
```

4) List all documents created.

```
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();
  });
});
```



The screenshot shows a code editor with a file named 'Que04.js' containing the following JavaScript code:

```
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
```

Below the code editor, a terminal window shows the command `node Que04.js` being executed. The output displays a deprecation warning followed by a JSON array of song details:

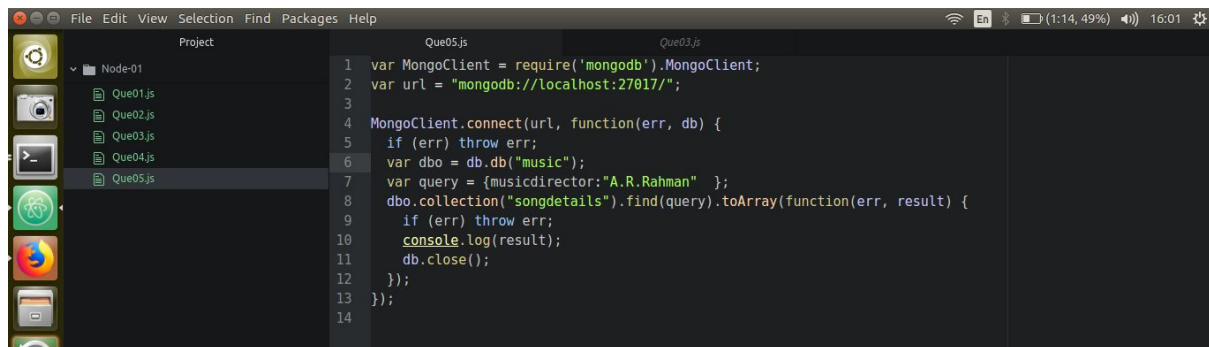
```
uklstu05@uklpc05:~/Desktop/Node-01$ node Que04.js
(node:3772) DeprecationWarning: current Server Discovery and Monitoring engine is deprecated, and will be removed in a future version. To use the new Server Discovery and Monitoring engine, pass option { useUnifiedTopology: true } to the MongoClient constructor.
[ { _id: 5e315d2da511ba0eafefc94c,
  Songname: 'ThanIyeThananthanIye',
  flim: 'Rhythm',
  musicdirector: 'A.R.Rahman',
  singer: 'Shankar mahadevan' },
  { _id: 5e315d2da511ba0eafefc94d,
  Songname: 'Evano Oruvan',
  flim: 'Alai Payuthey',
  musicdirector: 'A.R.Rahman',
  singer: 'Swarnalatha' },
  { _id: 5e315d2da511ba0eafefc94e,
  Songname: 'Roja Poonthottam',
  flim: 'Kannukul Nillai',
  musicdirector: 'Ilaiyaraaja',
  singer: [ 'Unnikrishnan', 'Anuradha Srinam' ] },
  { _id: 5e315d2da511ba0eafefc94f,
  Songname: 'VennilavaeVennilavaeVinnaitaandi',
  flim: 'Minsara Kanavu',
  musicdirector: 'A.R.Rahman',
  singer: [ 'Hartharan', 'Sadhana Sargam' ] },
  { _id: 5e315d2da511ba0eafefc950,
  Songname: 'Sollamal Thottuchellum Thendral',
  flim: 'Dheena',
  musicdirector: 'Yuvan Shankar Raja',
  singer: 'Hartharan' } ]
uklstu05@uklpc05:~/Desktop/Node-01$
```

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

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

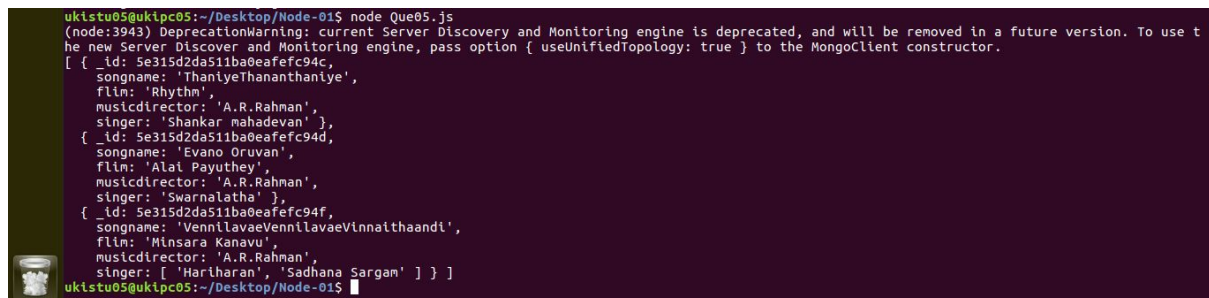
```
MongoClient.connect(url, function(err, db) {
  if (err) throw err;
  var dbo = db.db("music");
  var query = {musicdirector:"A.R.Rahman" };
  dbo.collection("songdetails").find(query).toArray(function(err, result) {
    if (err) throw err;
    console.log(result);
    db.close();
  });
});
```

```
});
});
```



```

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 = {musicdirector:"A.R.Rahman" };
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
```



```

uklstu05@uklpc05:~/Desktop/Node-01$ node Que05.js
(node:3943) DeprecationWarning: current Server Discovery and Monitoring engine is deprecated, and will be removed in a future version. To use t
he new Server Discover and Monitoring engine, pass option { useUnifiedTopology: true } to the MongoClient constructor.
[ { _id: 5e315d2da511ba0eafefc94c,
  songname: 'ThaniyeThananthaniye',
  flim: 'Rhythm',
  musicdirector: 'A.R.Rahman',
  singer: 'Shankar mahadevan' },
  { _id: 5e315d2da511ba0eafefc94d,
  songname: 'Evano Oruvan',
  flim: 'Alai Payuthey',
  musicdirector: 'A.R.Rahman',
  singer: 'Swarnalatha' },
  { _id: 5e315d2da511ba0eafefc94f,
  songname: 'VennilavaeVennilavaeVlnnatthaandi',
  flim: 'Minsara Kanavu',
  musicdirector: 'A.R.Rahman',
  singer: [ 'Hariharan', 'Sadhana Sargam' ] } ]
uklstu05@uklpc05:~/Desktop/Node-01$
```

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("music");
  var query = {musicdirector:"A.R.Rahman",singer:"Unnikrishnan"};
  dbo.collection("songdetails").find(query).toArray(function(err, result) {
    if (err) throw err;
    console.log(result);
    db.close();
  });
});
});
```



```

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 = {musicdirector:"A.R.Rahman",singer:"Unnikrishnan"};
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
```



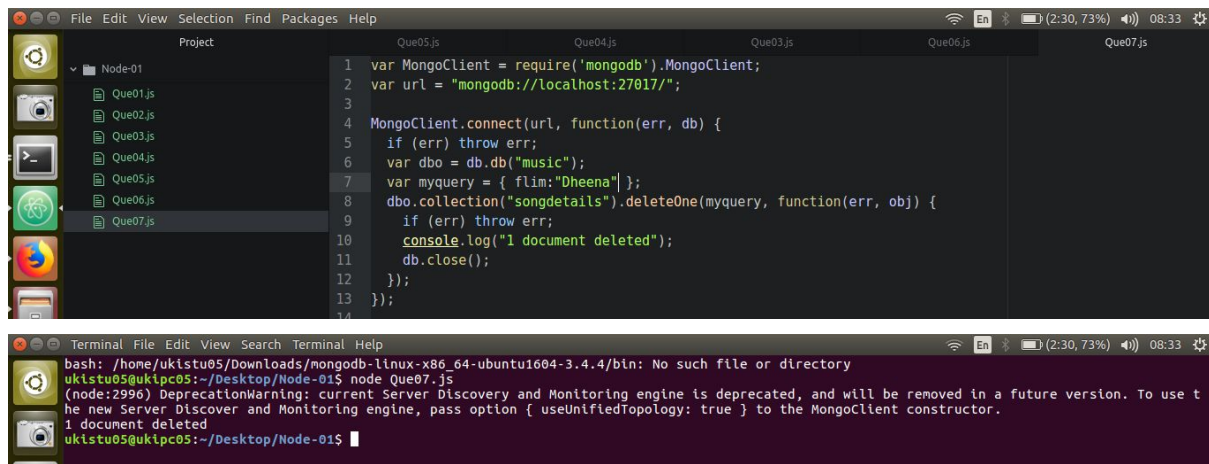
```

uklstu05@uklpc05:~/Desktop/Node-01$ node Que06.js
(node:4058) DeprecationWarning: current Server Discovery and Monitoring engine is deprecated, and will be removed in a future version. To use t
he new Server Discover and Monitoring engine, pass option { useUnifiedTopology: true } to the MongoClient constructor.
[]
uklstu05@uklpc05:~/Desktop/Node-01$
```

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 = { flim:"Dheena" };
  dbo.collection("songdetails").deleteOne(myquery, function(err, obj) {
    if (err) throw err;
    console.log("1 document deleted");
    db.close();
  });
});
```

The screenshot shows a code editor with a project named 'Node-01' containing files 'Que01.js' through 'Que07.js'. The active file, 'Que05.js', contains the JavaScript code for deleting a document from a MongoDB database. The terminal window below shows the command 'node Que07.js' being executed, resulting in the output '1 document deleted'. A deprecation warning for the MongoDB Server Discovery and Monitoring engine is also visible in the terminal.

```
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 = { flim:"Dheena" };
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 });
```

```
bash: /home/ukistu05/Downloads/mongodb-linux-x86_64-ubuntu1604-3.4.4/bin: No such file or directory
ukistu05@uklpc05:~/Desktop/Node-01$ node Que07.js
(node:2996) DeprecationWarning: current Server Discovery and Monitoring engine is deprecated, and will be removed in a future version. To use t
he new Server Discover and Monitoring engine, pass option { useUnifiedTopology: true } to the MongoClient constructor.
1 document deleted
ukistu05@uklpc05:~/Desktop/Node-01$
```

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 = {songname:"Verithanam
Verithanam",flim:"Bigil",musicdirector:"A.R.Rahman",singer:"Vijay" };
  dbo.collection("songdetails").insertOne(myobj, function(err, res) {
    if (err) throw err;
    console.log("1 document inserted");
    db.close();
  });
});
```

The screenshot shows a terminal window where the command 'node Que08.js' has been executed. The output shows a deprecation warning followed by '1 document inserted'.

```
ukistu05@uklpc05:~/Desktop/Node-01$ node Que08.js
(node:3420) DeprecationWarning: current Server Discovery and Monitoring engine is deprecated, and will be removed in a future version. To use t
he new Server Discover and Monitoring engine, pass option { useUnifiedTopology: true } to the MongoClient constructor.
1 document inserted
ukistu05@uklpc05:~/Desktop/Node-01$
```



```

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 = {songname:"Verithanam Verithanam",flim:"Bigil",musicdirector:"A.R.Rahman",singer:"Vijay" };
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

```

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

```

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

```

```

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 = {flim:"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

```

```

uklstu05@uklpc05:~/Desktop/Node-01$ node Que09.js
(node:4452) 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: 5e315d2da511ba0eafefc94f,
  Songname: 'VennilavaeVennilavaeVinnatthaandi',
  flim: 'Minsara Kanavu',
  musicdirector: 'A.R.Rahman',
  singer: [ 'Hariharan', 'Sadhana Sargam' ] } ]
uklstu05@uklpc05:~/Desktop/Node-01$

```

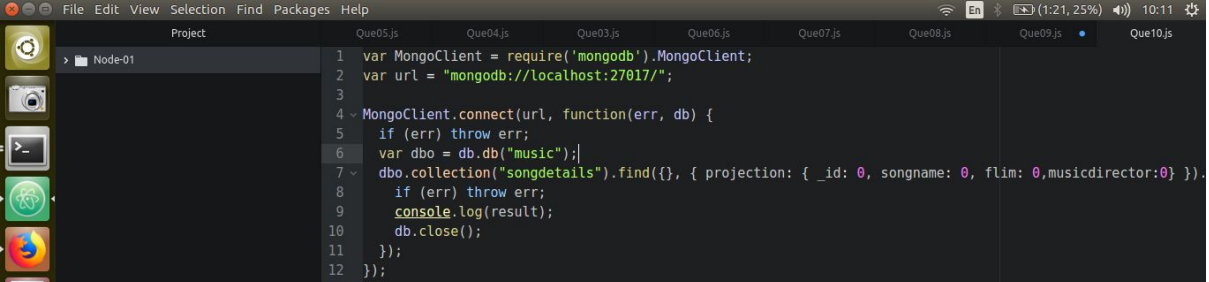
10)List out the singers' names in your document.

```

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, songname: 0, flim: 0,musicdirector:0 } }).toArray(function(err, result) {
    if (err) throw err;
    console.log(result);
    db.close();
  });
});

```



The screenshot shows an IDE window with a menu bar (File, Edit, View, Selection, Find, Packages, Help) and a toolbar. The left sidebar shows a project tree with a folder named 'Node-01'. The main editor area displays a file named 'Que10.js' with the following JavaScript code:

```
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, songname: 0, flim: 0, musicdirector: 0 } }).
8     if (err) throw err;
9     console.log(result);
10    db.close();
11  });
12 }
```



The screenshot shows a terminal window with the following output:

```
ukistu05@uklpc05:~/Desktop/Node-01$ node Que10.js
(node:4251) DeprecationWarning: current Server Discovery and Monitoring engine is deprecated, and will be removed in a future version. To use t
he new Server Discover and Monitoring engine, pass option { useUnifiedTopology: true } to the MongoClient constructor.
[ { singer: 'Shankar mahadevan' },
  { singer: 'Swarnalatha' },
  { singer: [ 'Unnikrishnan', 'Anuradha Sriram' ] },
  { singer: [ 'Hartharan', 'Sadhana Sargam' ] },
  { singer: 'Vijay' } ]
ukistu05@uklpc05:~/Desktop/Node-01$
```