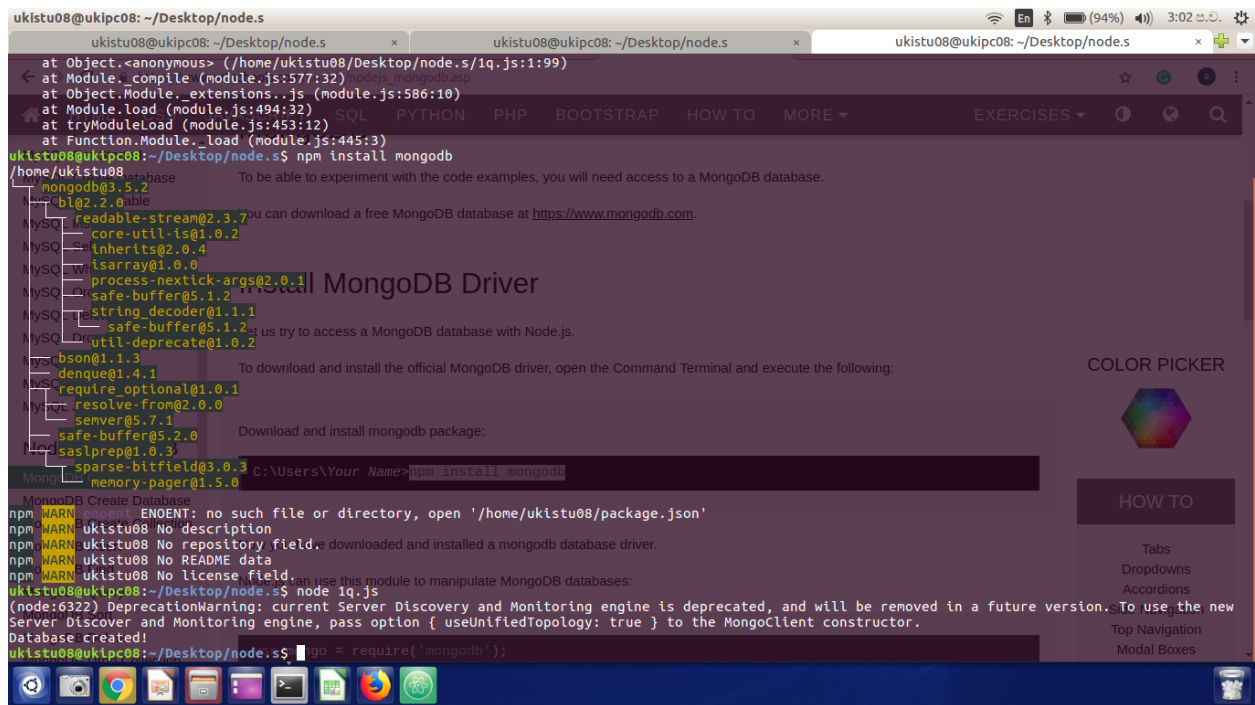
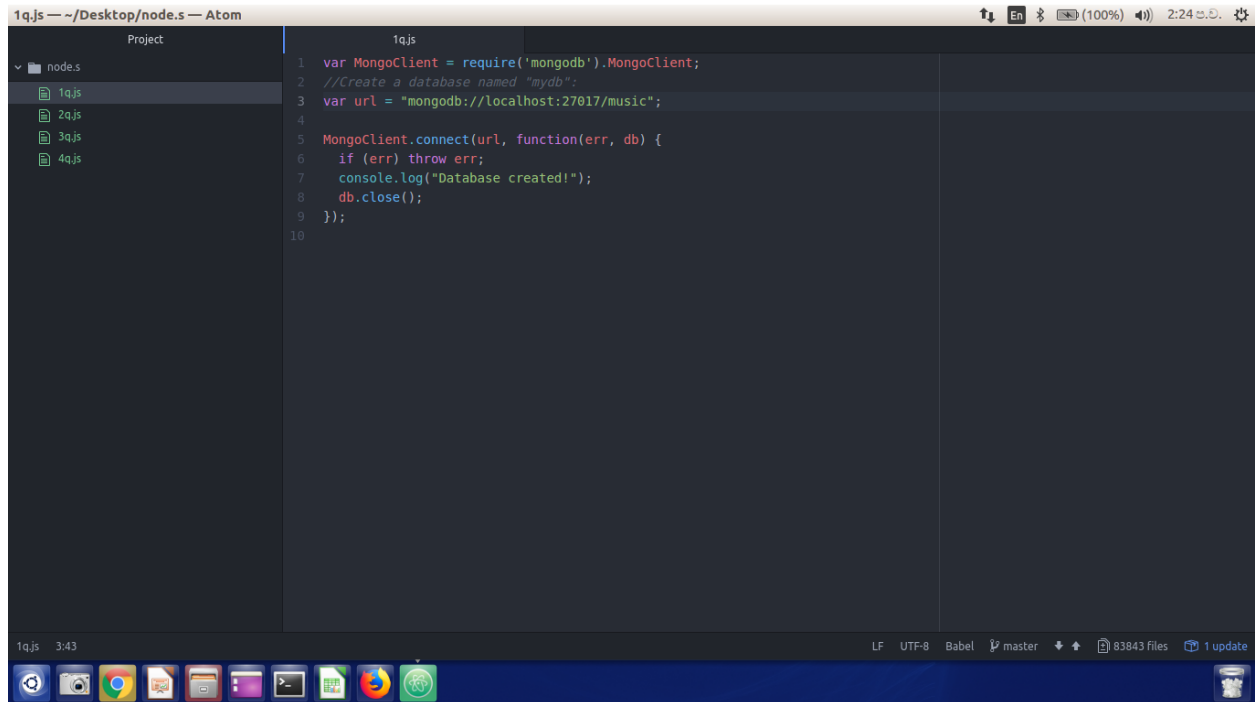


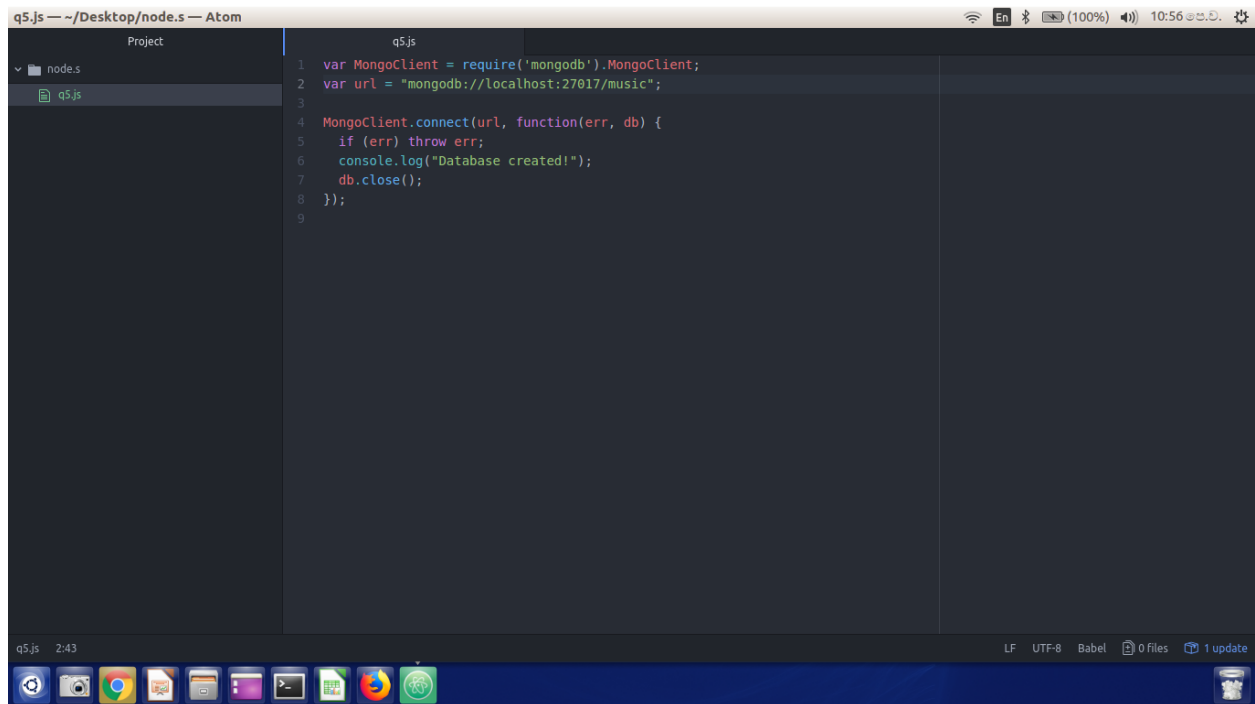
1.Create a Database called music.





The screenshot shows the Atom editor interface. The top status bar indicates the file is '1q.js' located at '~/Desktop/node.s' in the 'Atom' workspace. The left sidebar shows a project tree with a folder 'node.s' containing files '1q.js', '2q.js', '3q.js', and '4q.js'. The main editor area displays the code for '1q.js' with line numbers 1 through 10. The code connects to a MongoDB instance at localhost:27017/music and logs a message. The bottom status bar shows '1q.js 3:43', 'LF UTF-8 Babel master', and '83843 Files 1 update'. The system tray at the bottom includes icons for various applications and system status.

```
1 var MongoClient = require('mongodb').MongoClient;
2 //Create a database named "mydb":
3 var url = "mongodb://localhost:27017/music";
4
5 MongoClient.connect(url, function(err, db) {
6   if (err) throw err;
7   console.log("Database created!");
8   db.close();
9 });
10
```



The screenshot shows the Atom editor interface. The top status bar indicates the file is 'q5.js' located at '~/Desktop/node.s' in the 'Atom' workspace. The left sidebar shows a project tree with a folder 'node.s' containing the file 'q5.js'. The main editor area displays the code for 'q5.js' with line numbers 1 through 9. The code is identical to the one in the first screenshot. The bottom status bar shows 'q5.js 2:43', 'LF UTF-8 Babel', and '0 Files 1 update'. The system tray at the bottom includes icons for various applications and system status.

```
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
```

2.Create a collection called songdetails.

```
ukistu08@ukipc08: ~/Desktop/node.s
ukistu08@ukipc08: ~/Desktop/node.s
ukistu08@ukipc08:~/Desktop/node.s$ node 2q.js
(node:7109) 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.
Collection created!
ukistu08@ukipc08:~/Desktop/node.s$  MongoClient.connect(url, function(err, db) {
  3  if (err) throw err;
  4  var dbo = db.db("music");
  5  dbo.createCollection("songdetails", function(err, res) {
  6    if (err) throw err;
  7    console.log("Collection created!");
  8    db.close();
  9  });
10 });
11
```

2q.js 9:29 LF UTF-8 Babel master 84723 Files 1 update

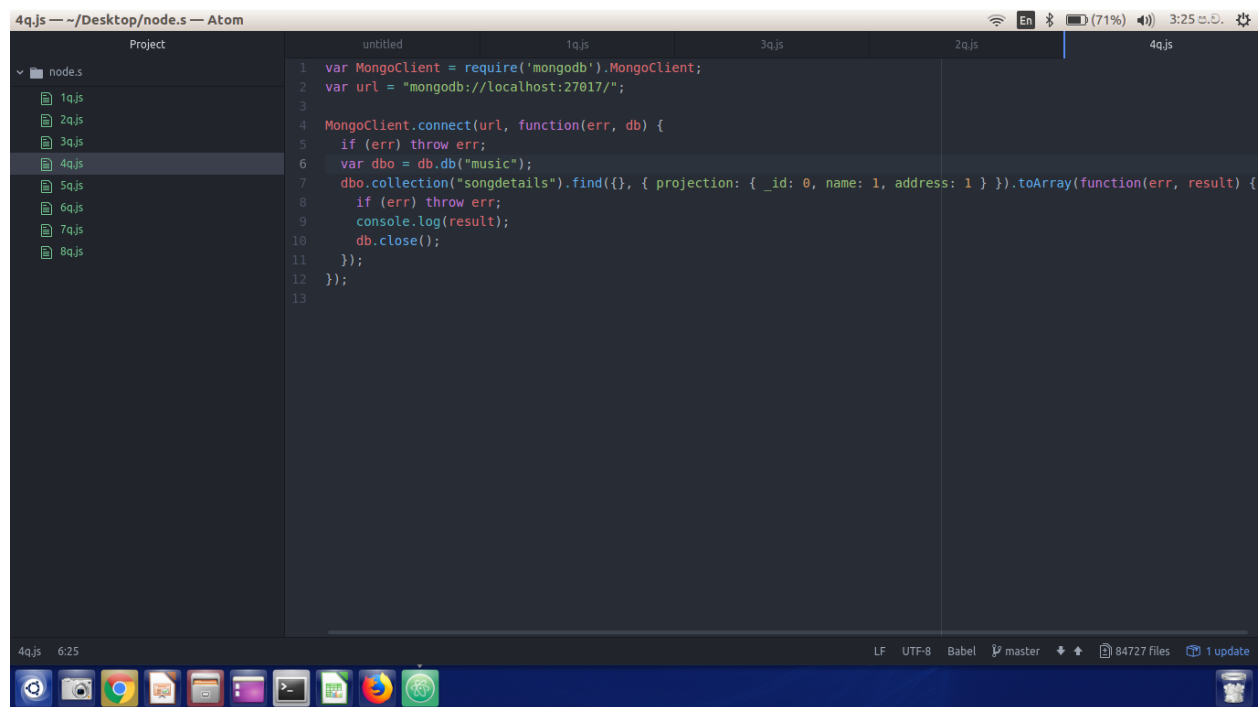
3. Create the above 5 song documents.

The image shows a terminal window and an Atom code editor. The terminal window displays the execution of a Node.js script (3q.js) that connects to a MongoDB database and inserts 5 song documents. The Atom code editor shows the source code of the 3q.js file, which uses the MongoClient to connect to the database and insert the documents.

```
ukistu08@ukipc08: ~/Desktop/node.s
ukistu08@ukipc08: ~/Desktop/node.s
ukistu08@ukipc08: ~/Desktop/node.s$ node 2q.js
(node:7109) 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!
ukistu08@ukipc08: ~/Desktop/node.s$ node 3q.js
(node:7176) 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
ukistu08@ukipc08: ~/Desktop/node.s$
```

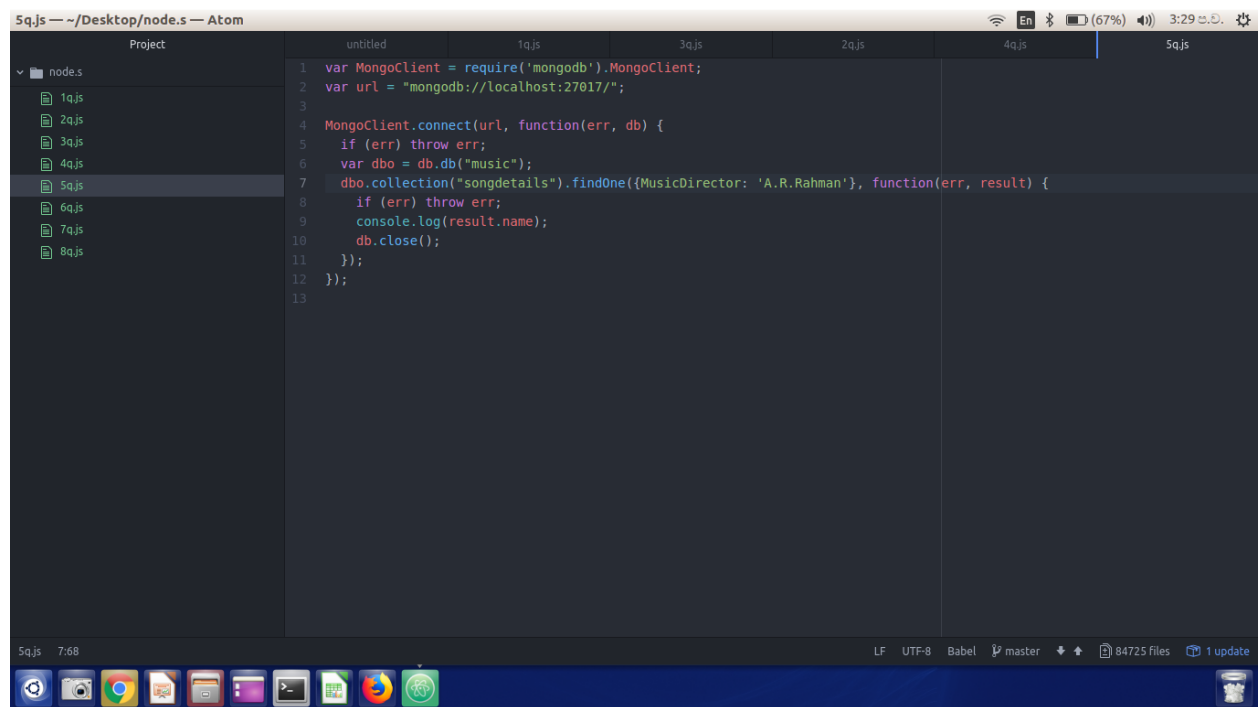
```
1 var MongoClient = require('mongodb').MongoClient;
2 var url = "mongodb://localhost:27017/";
3
4 MongoClient.connect(url, function(err, db) {
5   if (err) throw err;
6   var dbo = db.db("music");
7   var myobj = [
8     { SongName: 'ThaniyeThananthaniye', Film: 'Rhythm', MusicDirector: 'A.R.Rahman', Singer: 'Shankar mahadevan' },
9     { SongName: 'Evano Oruvan', Film: 'Alai Payuthey', MusicDirector: 'A.R.Rahman', Singer: 'Swarnalatha' },
10    { SongName: 'Raja Poonthottam', Film: 'Kannukkul Nilavu', MusicDirector: 'Ilaiyaraaja', Singer: 'Unnikrishnan, Anuradha Sr' },
11    { SongName: 'Vennilavae Vennilavae Vinnaitaandi', Film: 'Minsara Kanavu', MusicDirector: 'A.R.Rahman', Singer: 'Harihar' },
12    { SongName: 'Sollamal Thottu Chellum Thendral', Film: 'Dheena', MusicDirector: '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
```

4. List all documents created.



```
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, name: 1, address: 1 } }).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.

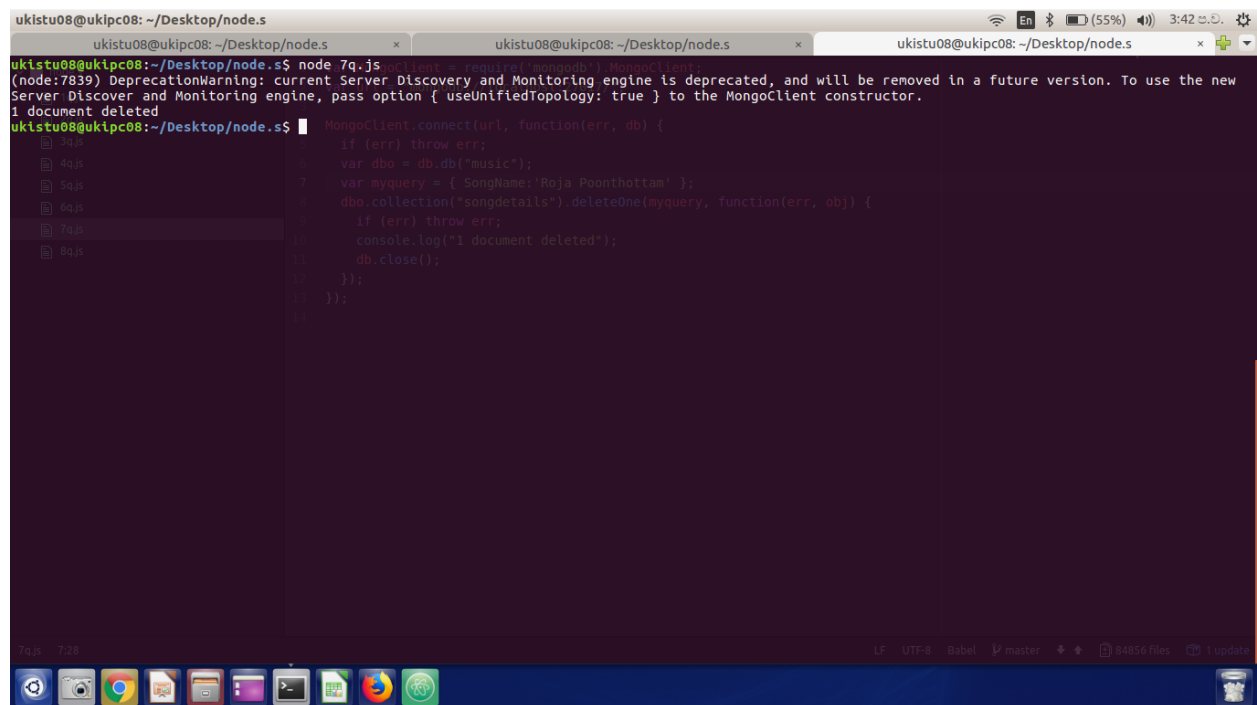


```
1 var MongoClient = require('mongodb').MongoClient;
2 var url = "mongodb://localhost:27017/";
3
4 MongoClient.connect(url, function(err, db) {
5   if (err) throw err;
6   var dbo = db.db("music");
7   dbo.collection("songdetails").findOne({MusicDirector: 'A.R.Rahman'}, function(err, result) {
8 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

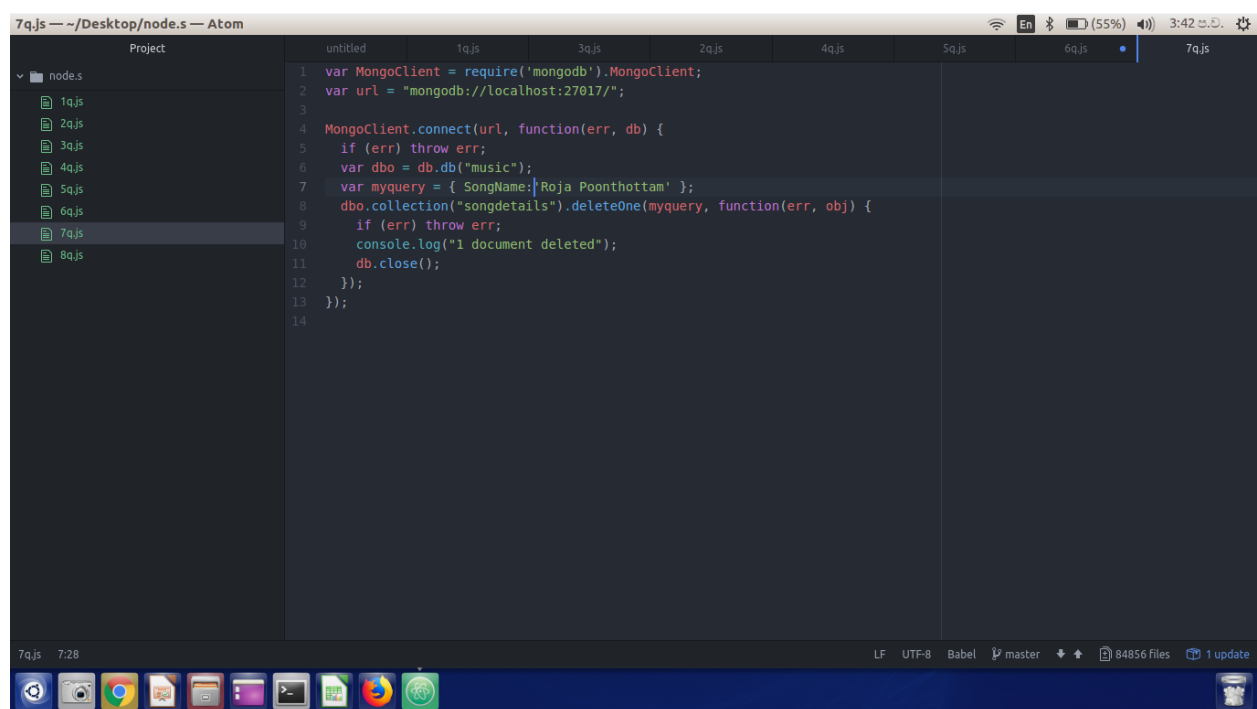
6) List A.R.Rahman's songs sung by Unnikrishnan.

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

```
ukistu08@ukipc08: ~/Desktop/node.s
ukistu08@ukipc08: ~/Desktop/node.s
ukistu08@ukipc08:~/Desktop/node.s$ node 7q.js
(node:7839) 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
ukistu08@ukipc08:~/Desktop/node.s$
```



```
7q.js -- ~/Desktop/node.s -- Atom
Project
  node.s
    1q.js
    2q.js
    3q.js
    4q.js
    5q.js
    6q.js
    7q.js
    8q.js
  untitled
  1q.js
  3q.js
  2q.js
  4q.js
  5q.js
  6q.js
  7q.js
  8q.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   var myquery = { SongName: 'Roja Poonthottam' };
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
```



8) Add new song which is your favourite.

The image consists of two screenshots from a Linux desktop environment. The top screenshot shows a terminal window with the following output:

```
ukistu08@ukipc08: ~/Desktop/node.s
ukistu08@ukipc08:~/Desktop/node.s$ node 7q.js
(node:7839) 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.
1 document deleted
ukistu08@ukipc08:~/Desktop/node.s$ node 8q.js
(node:8013) 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.
1 document inserted
ukistu08@ukipc08:~/Desktop/node.s$
```

The bottom screenshot shows the Atom code editor with the following code in the file 8q.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   var myobj = { SongName: "Company";
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
```

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

