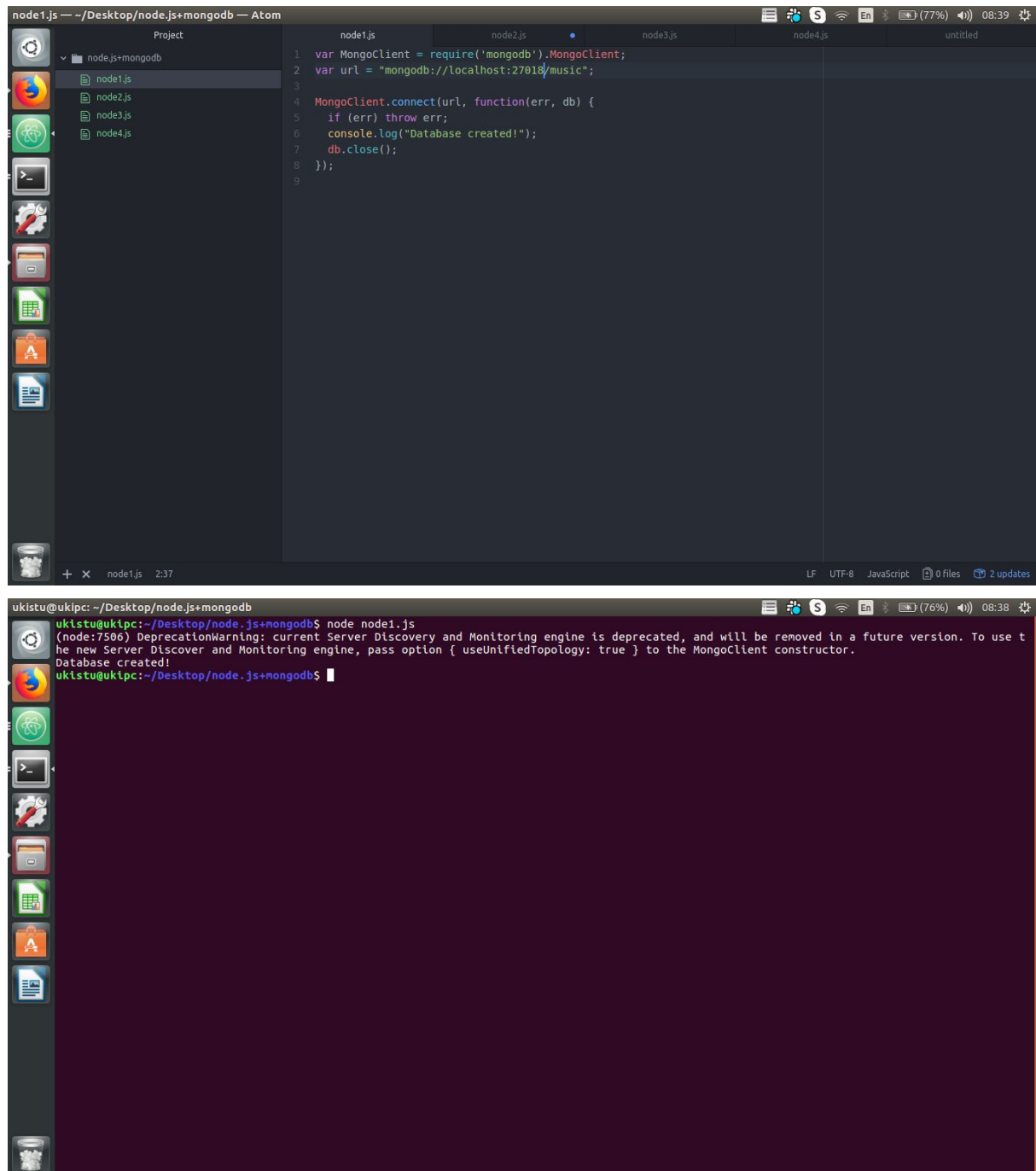


Mongod exercise 1

1) Create a Database called music.



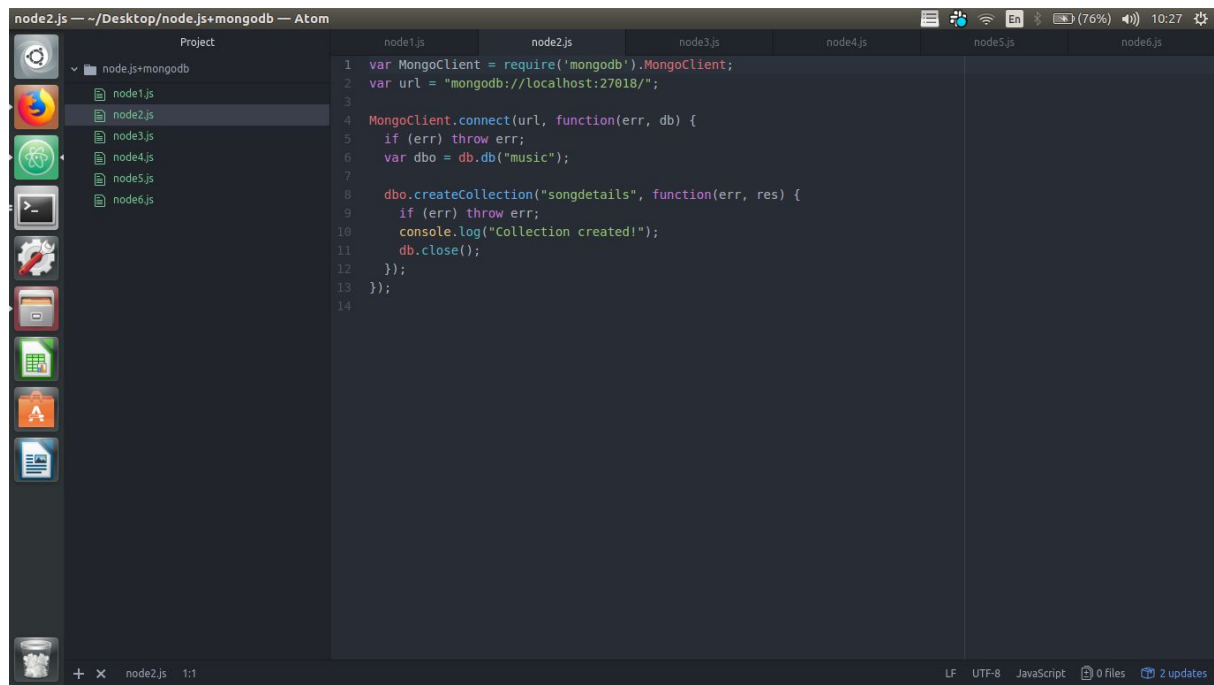
The image shows two windows from a Linux desktop environment. The top window is the Atom text editor, open to a file named `node1.js` in the project `~/Desktop/node.js+mongodb`. The code in `node1.js` is as follows:

```
1 var MongoClient = require('mongodb').MongoClient;
2 var url = "mongodb://localhost:27018/music";
3
4 MongoClient.connect(url, function(err, db) {
5   if (err) throw err;
6   console.log("Database created!");
7   db.close();
8 });
9
```

The bottom window is a terminal with the prompt `ukistu@ukipc: ~/Desktop/node.js+mongodb`. It shows the command `node node1.js` being executed. The output of the command is:

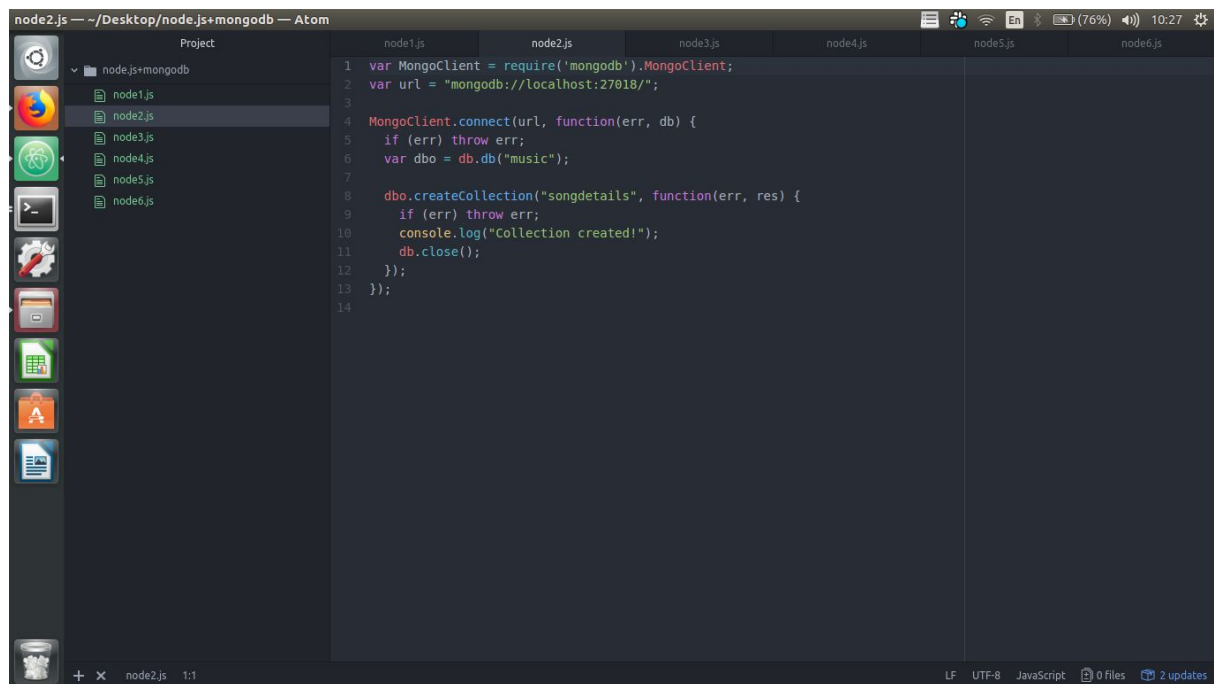
```
(node:7506) DeprecationWarning: current Server Discovery and Monitoring engine is deprecated, and will be removed in a future version. To use the new Server Discovery and Monitoring engine, pass option { useUnifiedTopology: true } to the MongoClient constructor.
Database created!
```

2) Create a collection called songdetails.



The screenshot shows the Atom text editor with a project named 'node.js+mongodb'. The file explorer on the left lists files from node1.js to node6.js. The main editor pane displays the code for node2.js, which connects to a MongoDB instance at localhost:27018, creates a database named 'music', and then creates a collection named 'songdetails'. The status bar at the bottom indicates the file is in LF, UTF-8 encoding, JavaScript language, with 0 files and 2 updates.

```
1 var MongoClient = require('mongodb').MongoClient;
2 var url = "mongodb://localhost:27018/";
3
4 MongoClient.connect(url, function(err, db) {
5   if (err) throw err;
6   var dbo = db.db("music");
7
8   dbo.createCollection("songdetails", function(err, res) {
9     if (err) throw err;
10    console.log("Collection created!");
11    db.close();
12  });
13 });
14
```



This is a duplicate of the first screenshot, showing the same Atom editor interface and JavaScript code for connecting to MongoDB and creating a collection.

```
1 var MongoClient = require('mongodb').MongoClient;
2 var url = "mongodb://localhost:27018/";
3
4 MongoClient.connect(url, function(err, db) {
5   if (err) throw err;
6   var dbo = db.db("music");
7
8   dbo.createCollection("songdetails", function(err, res) {
9     if (err) throw err;
10    console.log("Collection created!");
11    db.close();
12  });
13 });
14
```

3.Create the above 5 song documents.

The screenshot shows a code editor with a dark theme. On the left, there's a sidebar with a file explorer showing a project named 'node.js+mongodb' with files 'node1.js', 'node2.js', 'node3.js', 'node4.js', 'node5.js', and 'node6.js'. The main editor area displays the code for 'node1.js'. The code uses the 'mongodb' package to connect to a local MongoDB instance. It defines a collection 'songdetails' and inserts an array of song objects. The console log shows the number of documents inserted.

```

1 var MongoClient = require('mongodb').MongoClient;
2 var url = "mongodb://localhost:27018/";
3
4 MongoClient.connect(url, function(err, db) {
5   if (err) throw err;
6   var dbo = db.db("music");
7   var myobj = [
8     { songname: 'Thaniye thananthaniye', film: 'rhythm', musicdirector: 'A.R.Rahman', singer: 'Shanker Mahadevan' },
9     { songname: 'evano oruvan', film: 'alai payuthe', musicdirector: 'A.R.Rahman', singer: 'Swanalatha' },
10    { songname: 'roja poonthotaam', film: 'kannukul nilavu', musicdirector: 'Ilaiyaraaja', singer: ['Unnikrishnan', 'Anurag Kashyap'] },
11    { songname: 'venilave venilave', film: 'minsara kanavu', musicdirector: 'A.R.Rahman', singer: ['Harikaran', 'sadhana'] },
12    { songname: 'sollamal thootu sellum thendral', film: 'dheena', musicdirector: 'Yuvan', singer: 'Harikaran' },
13  ];
14
15  dbo.collection("songdetails").insertMany(myobj, function(err, res) {
16    if (err) throw err;
17    console.log("Number of documents inserted: " + res.insertedCount);
18    db.close();
19  });
20 });
21

```

```
node4.js -- ~/Desktop/node.js+mongodb — Atom

Project
  node.js+mongodb
    node1.js
    node2.js
    node3.js
    node4.js

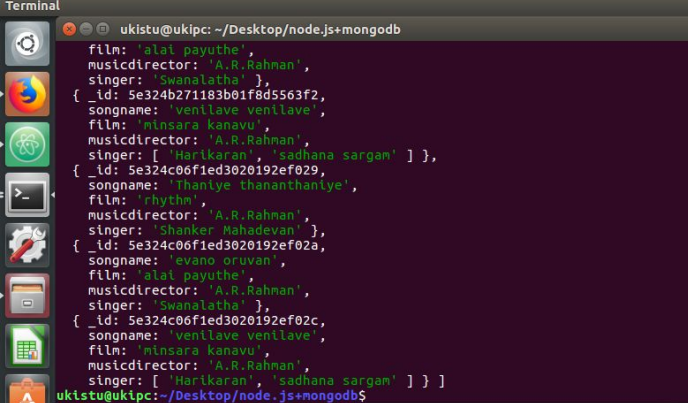
node1.js
node2.js
node3.js
node4.js
untitled

1 var MongoClient = require('mongodb').MongoClient;
2 var url = "mongodb://localhost:27018/";
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
```

```
ukistu@ukip: ~/Desktop/node.js+mongodb

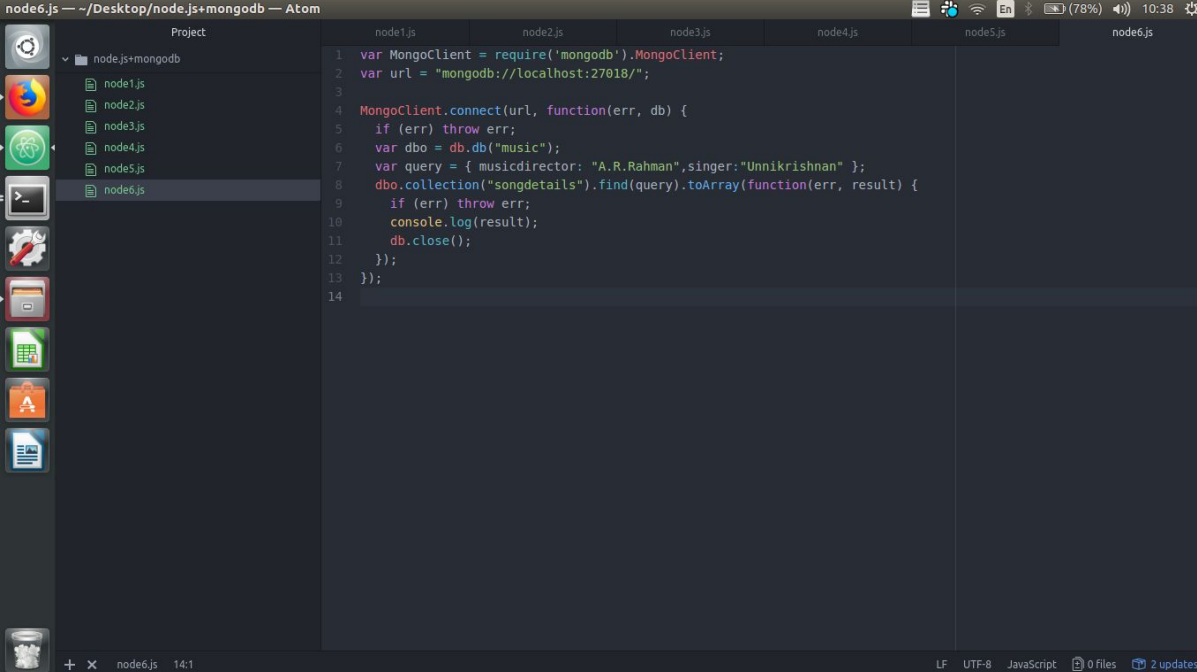
{ _id: 5e32498eda7aab1e0e663a28,
  songname: 'evano oruvan',
  film: 'aial payuthe',
  musicdirector: 'A.R.Rahman',
  singer: 'Swanulatha' },
{ _id: 5e32498eda7aab1e0e663a29,
  songname: 'roja poonthotaam',
  film: 'kannukul nilavu',
  musicdirector: 'Ilaiyaraaja',
  singer: 'Unnikrishnan,Anuradha sritam' },
{ _id: 5e32498eda7aab1e0e663a2a,
  songname: 'venilave ventilave',
  film: 'minsara kanavu',
  musicdirector: 'A.R.Rahman',
  singer: 'Harikaran,sadhana sargam' },
{ _id: 5e32498eda7aab1e0e663a2b,
  songname: 'sollamal thootu sellum thendral',
  film: 'dheena',
  musicdirector: 'Yuvan',
  singer: 'Harikaran' },
{ _id: 5e324b271183b01f8d5563ef,
  songname: 'Thanlye thananthanlye',
  film: 'rhythm',
  musicdirector: 'A.R.Rahman',
  singer: 'Shanker Mahadevan' },
{ _id: 5e324b271183b01f8d5563f0,
  songname: 'evano oruvan',
  film: 'aial payuthe',
  musicdirector: 'A.R.Rahman',
  singer: 'Swanulatha' },
{ _id: 5e324b271183b01f8d5563f1,
  songname: 'roja poonthotaam',
  film: 'kannukul nilavu',
  musicdirector: 'Ilaiyaraaja',
  singer: [ 'Unnikrishnan', 'Anuradha sritam' ] },
{ _id: 5e324b271183b01f8d5563f2,
  songname: 'venilave ventilave',
  film: 'minsara kanavu',
  musicdirector: 'A.R.Rahman',
  singer: [ 'Harikaran', 'sadhana sargam' ] },
{ _id: 5e324b271183b01f8d5563f3,
  songname: 'sollamal thootu sellum thendral',
  film: 'dheena',
```

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



```
Terminal
ukistu@ukipc: ~/Desktop/node.js+mongodb
film: 'alai payuthe',
musicdirector: 'A.R.Rahman',
singer: 'Swanlatha' },
{ _id: 5e324b271183b01f8d5563f2,
  songname: 'venilave venilave',
  film: 'minsara kanavu',
  musicdirector: 'A.R.Rahman',
  singer: [ 'Harikaran', 'sadhana sargam' ] },
{ _id: 5e324c06f1ed3020192ef029,
  songname: 'Thaniye thananthaniye',
  film: 'rhythm',
  musicdirector: 'A.R.Rahman',
  singer: 'Shanker Mahadevan' },
{ _id: 5e324c06f1ed3020192ef02a,
  songname: 'evano oruvan',
  film: 'alai payuthe',
  musicdirector: 'A.R.Rahman',
  singer: 'Swanlatha' },
{ _id: 5e324c06f1ed3020192ef02c,
  songname: 'venilave venilave',
  film: 'minsara kanavu',
  musicdirector: 'A.R.Rahman',
  singer: [ 'Harikaran', 'sadhana sargam' ] } ]
ukistu@ukipc:~/Desktop/node.js+mongodb$
```

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



```
node6.js — ~/Desktop/node.js+mongodb — Atom
Project
node.js+mongodb
node1.js
node2.js
node3.js
node4.js
node5.js
node6.js

1 var MongoClient = require('mongodb').MongoClient;
2 var url = "mongodb://localhost:27018/";
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
```



```
ukistu@ukipc: ~/Desktop/node.js+mongodb
{
  singer: 'Shanker Mahadevan' },
{
  id: 5e324b271183b01f8d5563f0,
  songname: 'evano oruvan',
  film: 'alal payuthe',
  musicdirector: 'A.R.Rahman',
  singer: 'Swanulatha' },
{
  id: 5e324b271183b01f8d5563f2,
  songname: 'venilave venilave',
  film: 'minsara kanavu',
  musicdirector: 'A.R.Rahman',
  singer: [ 'Harikaran', 'sadhana sargam' ] },
{
  id: 5e324c06f1ed3020192ef029,
  songname: 'Thanlye thananthanlye',
  film: 'rhythm',
  musicdirector: 'A.R.Rahman',
  singer: 'Shanker Mahadevan' },
{
  id: 5e324c06f1ed3020192ef02a,
  songname: 'evano oruvan',
  film: 'alal payuthe',
  musicdirector: 'A.R.Rahman',
  singer: 'Swanulatha' },
{
  id: 5e324c06f1ed3020192ef02c,
  songname: 'venilave venilave',
  film: 'minsara kanavu',
  musicdirector: 'A.R.Rahman',
  singer: [ 'Harikaran', 'sadhana sargam' ] } ]
ukistu@ukipc:~/Desktop/node.js+mongodb$ node node6.js
(node:4759) 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.
[]
ukistu@ukipc:~/Desktop/node.js+mongodb$ node node6.js
(node:4778) 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.
[]
ukistu@ukipc:~/Desktop/node.js+mongodb$ node node6.js
(node:4832) 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.
[]
ukistu@ukipc:~/Desktop/node.js+mongodb$ node node6.js
(node:5246) 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.
[]
ukistu@ukipc:~/Desktop/node.js+mongodb$
```

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

```
node1.js node2.js node3.js node4.js node5.js node6.js node7.js
1 var MongoClient = require('mongodb').MongoClient;
2 var url = "mongodb://localhost:27018/";
3
4 MongoClient.connect(url, function(err, db) {
5   if (err) throw err;
6   var dbo = db.db("music");
7   var myquery = { film:"rhythm" };
8   dbo.collection("songname").deleteOne(myquery, function(err, obj) {
9     if (err) throw err;
10    console.log("1 document deleted");
11    db.close();
12  });
13 });
14
```

```
ukistu@ukipc: ~/Desktop/node.js+mongodb
musicdirector: 'A.R.Rahman',
singer: 'Sonalatha' ] },
{ id: 5e324b271183b01f8d5563f2,
  songname: 'Venilave venilave',
  film: 'minsara kanavu',
  musicdirector: 'A.R.Rahman',
  singer: [ 'Harikaran', 'sadhana sargam' ] },
{ id: 5e324c06f1ed3020192ef029,
  songname: 'Thaniye thananthaniye',
  film: 'rhythm',
  musicdirector: 'A.R.Rahman',
  singer: 'Shanker Mahadevan' },
{ id: 5e324c06f1ed3020192ef02a,
  songname: 'evano oruvan',
  film: 'alai payuthe',
  musicdirector: 'A.R.Rahman',
  singer: 'Sonalatha' },
{ id: 5e324c06f1ed3020192ef02c,
  songname: 'Venilave venilave',
  film: 'minsara kanavu',
  musicdirector: 'A.R.Rahman',
  singer: [ 'Harikaran', 'sadhana sargam' ] } ] }
```

8) Add new song which is your favourite.

```
node1.js node2.js node3.js node4.js node5.js node6.js node7.js node8.js
1 MongoClient = require('mongodb').MongoClient;
2 url = "mongodb://localhost:27017/";
3
4 client.connect(url, function(err, db) {
5   (err) throw err;
6   dbo = db.db("music");
7   myobj = { songname: "munbee vaa", film: "sillnu oru kadhal", musicdirector: "A.R.Rahman", singer: "sherya gosha" };
8   .collection("songdetails").insertOne(myobj, function(err, res) {
9     f (err) throw err;
10    console.log("1 document inserted");
11    .close();
12  }
13
14
```

```
ukistu@ukipc: ~/Desktop/node.js+mongodb$ node node6.js
(node:4759) 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.
[]
ukistu@ukipc: ~/Desktop/node.js+mongodb$ node node6.js
(node:4778) 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.
[]
ukistu@ukipc: ~/Desktop/node.js+mongodb$ node node6.js
(node:4832) 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.
[]
ukistu@ukipc: ~/Desktop/node.js+mongodb$ node node6.js
(node:5246) 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.
[]
ukistu@ukipc: ~/Desktop/node.js+mongodb$ node node7.js
(node:5522) 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
ukistu@ukipc: ~/Desktop/node.js+mongodb$ node node8.js
(node:5629) 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
ukistu@ukipc: ~/Desktop/node.js+mongodb$
```

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

```
ukistu@ukipc:~/Desktop/node.js+mongodb$ node node9.js
(node:5889) DeprecationWarning: current Server Discovery and Monitoring engine is deprecated, and will be removed in a future version. To use t
he new Server Discoverer and Monitoring engine, pass option { useUnifiedTopology: true } to the MongoClient constructor.
[ { _id: 5e324b271183b01f8d5563f2,
  songname: 'venilavevenilave',
  film: 'minsara kanavu',
  musicdirector: 'A.R.Rahman',
  singer: [ 'Harikaran', 'sadhana sargam' ] },
  { _id: 5e324b271183b01f8d5563f3,
  songname: 'venilavevenilave',
  film: 'minsara kanavu',
  musicdirector: 'A.R.Rahman',
  singer: [ 'Harikaran', 'sadhana sargam' ] } ],
```

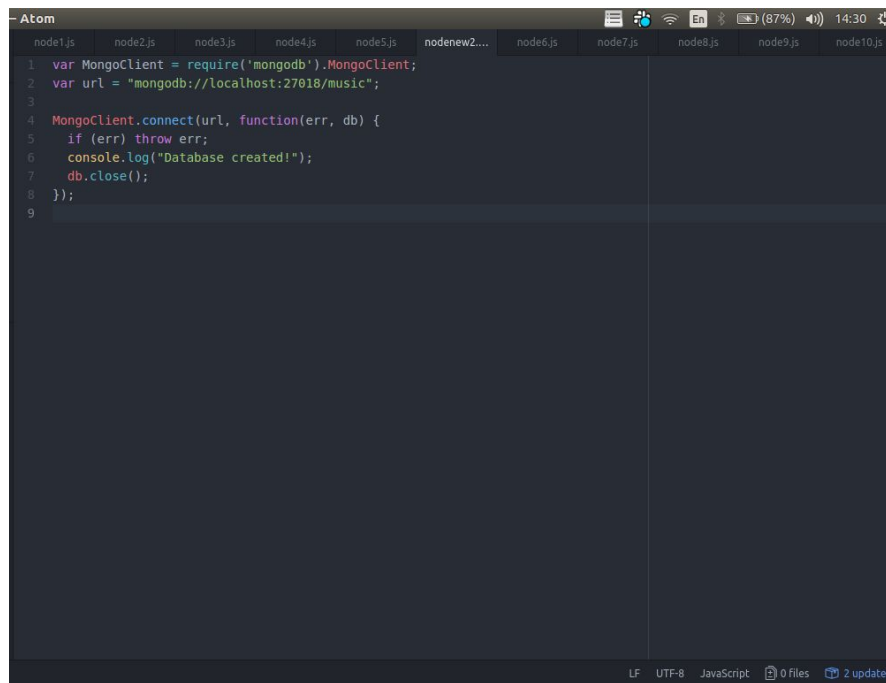
```
node1.js node2.js node3.js node4.js node5.js node6.js node7.js node8.js node9.js x
1 var MongoClient = require('mongodb').MongoClient;
2 var url = "mongodb://localhost:27018/";
3
4 MongoClient.connect(url, function(err, db) {
5   if (err) throw err;
6   var dbo = db.db("music");
7   var query = { film: "minsara kanavu", singer: "Harikaran" };
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
```

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

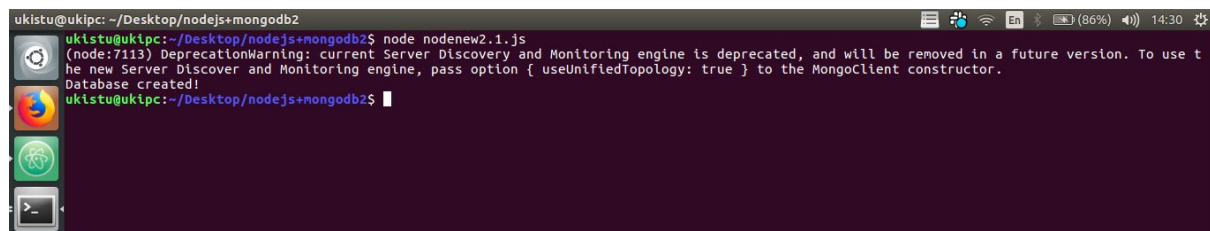
```
node1.js node2.js node3.js node4.js node5.js node6.js node7.js node8.js node9.js node10.js
1 var MongoClient = require('mongodb').MongoClient;
2 var url = "mongodb://localhost:27018/";
3
4 MongoClient.connect(url, function(err, db) {
5   if (err) throw err;
6   var dbo = db.db("music");
7   dbo.collection("songdetails").find({}, { projection: { _id: 0, songname: 0, film: 0, musicdirector: 0, singer: 1 } }).t
8   if (err) throw err;
9   console.log(result);
10  db.close();
11 });
12 });
13
```


Mongodb 2

1) Create a Database called student

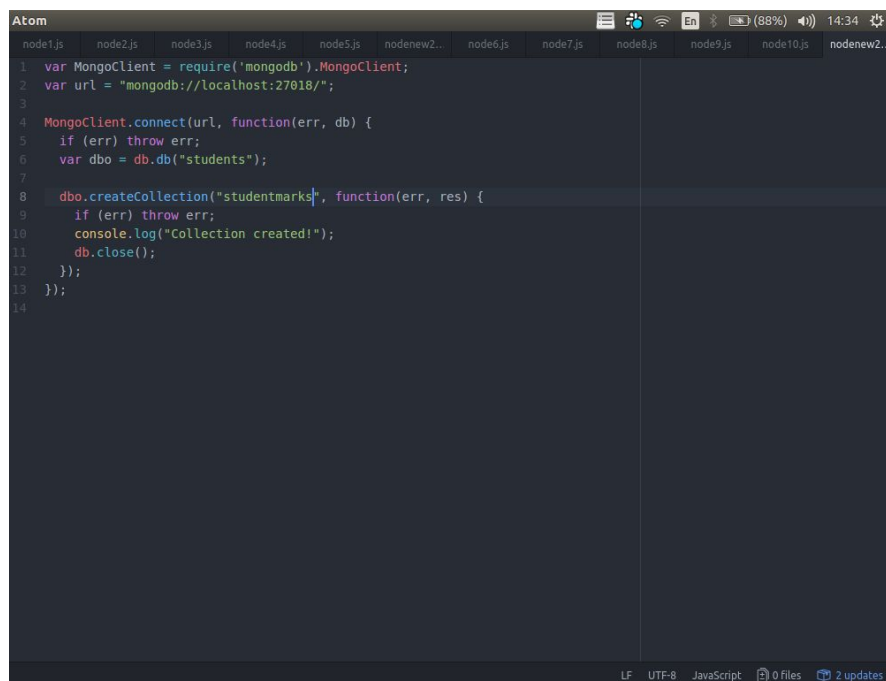


```
1 var MongoClient = require('mongodb').MongoClient;
2 var url = "mongodb://localhost:27018/music";
3
4 MongoClient.connect(url, function(err, db) {
5   if (err) throw err;
6   console.log("Database created!");
7   db.close();
8 });
9
```



```
ukistu@ukipc: ~/Desktop/nodejs+mongodb2
ukistu@ukipc:~/Desktop/nodejs+mongodb2$ node nodenew2.1.js
(node:7113) 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!
ukistu@ukipc:~/Desktop/nodejs+mongodb2$
```

2) Create a collection called studentmarks



```
1 var MongoClient = require('mongodb').MongoClient;
2 var url = "mongodb://localhost:27018/";
3
4 MongoClient.connect(url, function(err, db) {
5   if (err) throw err;
6   var dbo = db.db("students");
7
8   dbo.createCollection("studentmarks", function(err, res) {
9     if (err) throw err;
10    console.log("Collection created!");
11    db.close();
12  });
13 });
14
```

```
ukistu@ukipc: ~/Desktop/nodejs+mongodb2
ukistu@ukipc:~/Desktop/nodejs+mongodb2$ node nodenew2.1.js
(node:7113) 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!
ukistu@ukipc:~/Desktop/nodejs+mongodb2$ node nodenew2.2.js
(node:7264) 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!
ukistu@ukipc:~/Desktop/nodejs+mongodb2$
```

3) Create the documents listed in above table.

```
Atom
node1.js node2.js node3.js node4.js node5.js node6.js node7.js node8.js node9.js node10.js
1 var MongoClient = require('mongodb').MongoClient;
2 var url = "mongodb://localhost:27018/";
3
4 MongoClient.connect(url, function(err, db) {
5   if (err) throw err;
6   var dbo = db.db("music");
7   var myobj = [
8     { name: 'Mala', maths_marks: 45, english_marks: 53, science_marks: 72 },
9     { name: 'Venu', maths_marks: 80, english_marks: 75, science_marks: 85 },
10    { name: 'Kala', maths_marks: 32, english_marks: 46, science_marks: 53 },
11    { name: 'Aruli', maths_marks: 78, english_marks: 85, science_marks: 80 },
12    { name: 'Shayu', maths_marks: 80, english_marks: 76, science_marks: 65 },
13    { name: 'Kumaran', maths_marks: 32, english_marks: 73, science_marks: 84 },
14    { name: 'Lucky', maths_marks: 64, english_marks: 90, science_marks: 45 },
15    { name: 'Gva', maths_marks: 71, english_marks: 75, science_marks: 56 },
16    { name: 'Raam', maths_marks: 41, english_marks: 65, science_marks: 88 },
17  ];
18  dbo.collection("songdetails").insertMany(myobj, function(err, res) {
19    if (err) throw err;
20    console.log("Number of documents inserted: " + res.insertedCount);
21    db.close();
22  });
23 });
24
```

```
ukistu@ukipc: ~/Desktop/nodejs+mongodb2
ukistu@ukipc:~/Desktop/nodejs+mongodb2$ node nodenew2.1.js
(node:7113) 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!
ukistu@ukipc:~/Desktop/nodejs+mongodb2$ node nodenew2.2.js
(node:7264) 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!
ukistu@ukipc:~/Desktop/nodejs+mongodb2$ node nodes2.3.js
(node:7528) 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: 9
ukistu@ukipc:~/Desktop/nodejs+mongodb2$
```

```
ukistu@ukipc: ~/Desktop/nodejs+mongodb2
{
  english_marks: 53,
  science_marks: 75 },
{
  _id: 5e32a9e0ccc4c521b618297a,
  name: 'Venu',
  maths_marks: 80,
  english_marks: 75,
  science_marks: 85 },
{
  _id: 5e32a9e0ccc4c521b618297b,
  name: 'Kala',
  maths_marks: 32,
  english_marks: 46,
  science_marks: 53 },
{
  _id: 5e32a9e0ccc4c521b618297c,
  name: 'Arull',
  maths_marks: 78,
  english_marks: 85,
  science_marks: 86 },
{
  _id: 5e32a9e0ccc4c521b618297d,
  name: 'Shayu',
  maths_marks: 80,
  english_marks: 76,
  science_marks: 65 },
{
  _id: 5e32a9e0ccc4c521b618297e,
  name: 'Kunaran',
  maths_marks: 32,
  english_marks: 73,
  science_marks: 84 },
{
  _id: 5e32a9e0ccc4c521b618297f,
  name: 'Lucky',
  maths_marks: 64,
  english_marks: 90,
  science_marks: 45 },
{
  _id: 5e32a9e0ccc4c521b6182980,
  name: 'Giva',
  maths_marks: 71,
  english_marks: 75,
  science_marks: 56 },
{
  _id: 5e32a9e0ccc4c521b6182981,
  name: 'Raam',
  maths_marks: 41,
  english_marks: 65,
  science_marks: 88 } ]
ukistu@ukipc:~/Desktop/nodejs+mongodb2$
```

4) Increase the maths marks of Mala by 6 marks.

```
Atom
nodenes2.3.js  nodenew2.4.js  nodenew2.5.js  node6.js  nodenew2.1.js  nodenew2.2.js

1 var MongoClient = require('mongodb').MongoClient;
2 var url = "mongodb://localhost:27018/";
3
4 MongoClient.connect(url, function(err, db) {
5   if (err) throw err;
6   var dbo = db.db("students");
7
8
9   dbo.collection("studentmarks").update({name: 'Mala'}, {$inc: {maths_marks: 6}}, function(err, res) {
10     if (err) throw err;
11     console.log("1 document updated");
12     db.close();
13   });
14 });
15
```

```

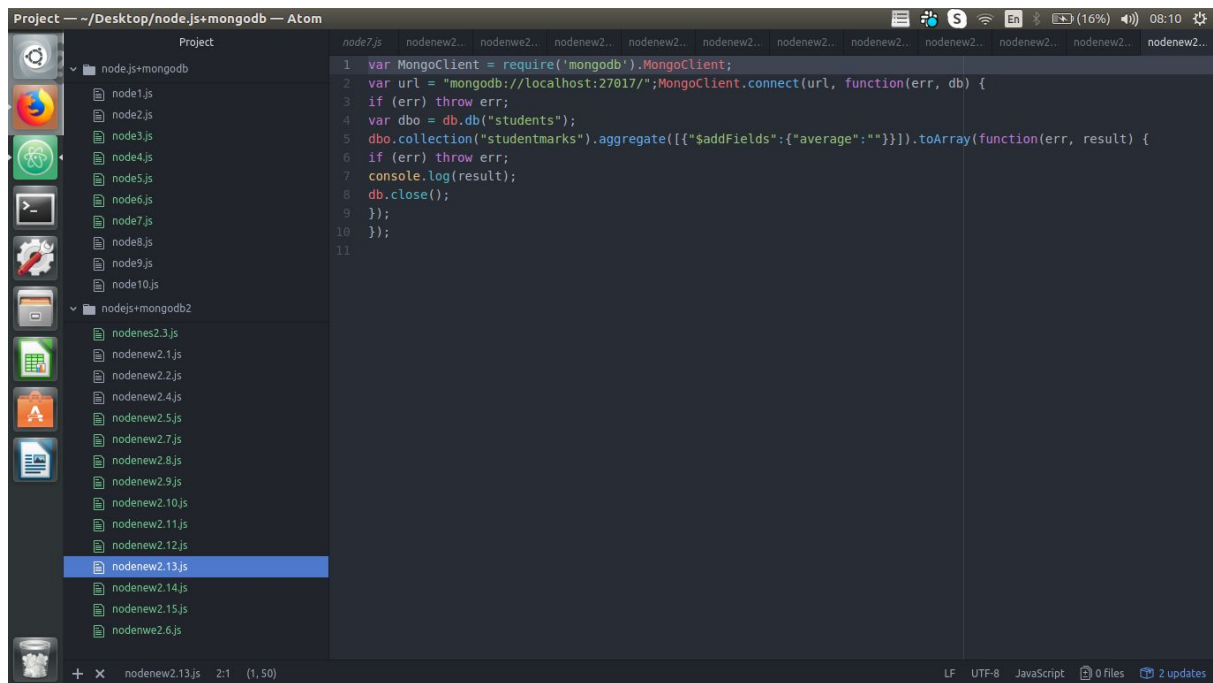
ukistu@ukitpc: ~/Desktop/nodejs+mongodb2
Database created!
ukistu@ukitpc:~/Desktop/nodejs+mongodb2$ node nodenew2.2.js
(node:7264) DeprecationWarning: current Server Discovery and Monitoring engine is deprecated, and will be removed in a future version. To use the new Server Discover and Monitoring engine, pass option { useUnifiedTopology: true } to the MongoClient constructor.
Collection created!
ukistu@ukitpc:~/Desktop/nodejs+mongodb2$ node nodenes2.3.js
(node:7528) 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: 9
ukistu@ukitpc:~/Desktop/nodejs+mongodb2$ node nodenes2.3.js
(node:7620) 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: 9
ukistu@ukitpc:~/Desktop/nodejs+mongodb2$ node nodenew2.4.js
(node:7645) 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.
[]
ukistu@ukitpc:~/Desktop/nodejs+mongodb2$ node nodenew2.4.js
(node:7666) 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.
[]
ukistu@ukitpc:~/Desktop/nodejs+mongodb2$ node nodenew2.5.js
/home/ukistu/Desktop/nodejs+mongodb2/nodenew2.5.js:9
dbo.collection("studentmarks").update({name:'Mala',{$inc:{maths_marks:6}}}, function(err, res) {
                                     ^
SyntaxError: Unexpected token {
    at new Script (vm.js:79:7)
    at createScript (vm.js:251:10)
    at Object.runInThisContext (vm.js:303:10)
    at Module._compile (internal/modules/cjs/loader.js:657:28)
    at Object.Module._extensions..js (internal/modules/cjs/loader.js:700:10)
    at Module.load (internal/modules/cjs/loader.js:599:32)
    at tryModuleLoad (internal/modules/cjs/loader.js:538:12)
    at Function.Module.load (internal/modules/cjs/loader.js:530:3)
    at Function.Module.runMain (internal/modules/cjs/loader.js:742:12)
    at startup (internal/bootstrap/node.js:279:19)
ukistu@ukitpc:~/Desktop/nodejs+mongodb2$ node nodenew2.5.js
(node:7917) 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.
(node:7917) DeprecationWarning: collection.update is deprecated. Use updateOne, updateMany, or bulkWrite instead.
1 document updated
ukistu@ukitpc:~/Desktop/nodejs+mongodb2$

```

5) List the names of students who got more than 50 marks in Maths Subject.

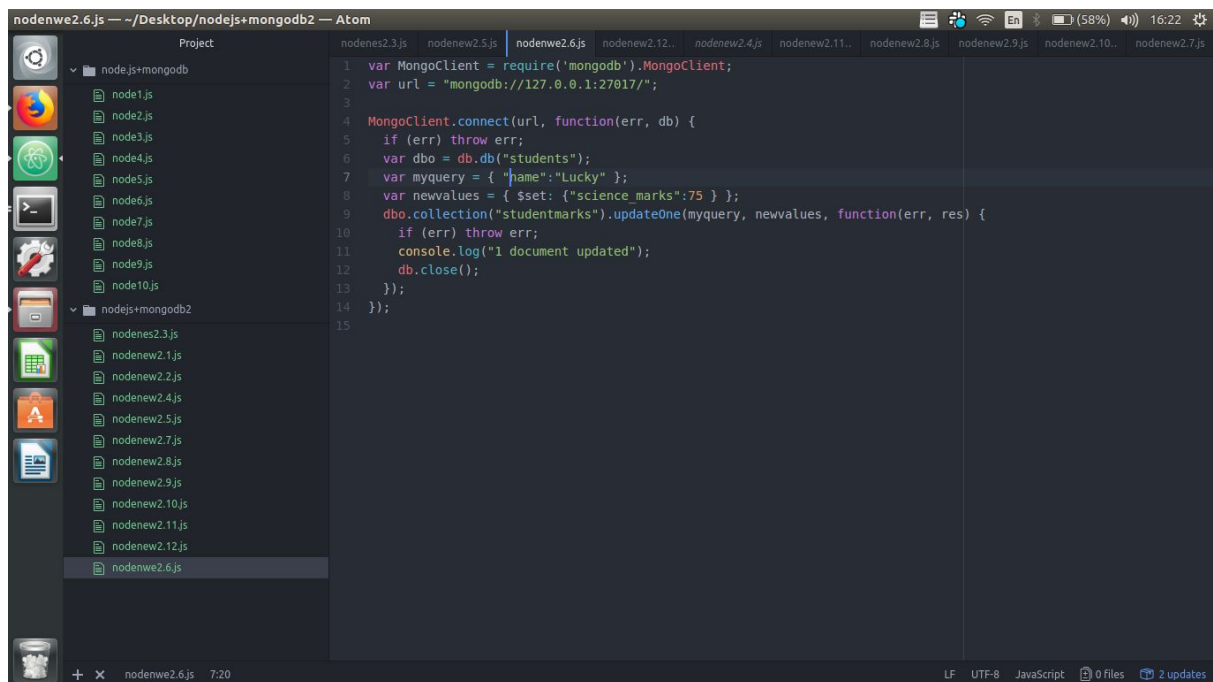
The screenshot shows the Atom code editor interface. The top status bar indicates the file is 'node1.js' in the directory '--/Desktop/nodejs+mongodb2'. The left sidebar shows the project structure with a folder 'nodejs+mongodb2' containing files 'node1.js' through 'node10.js'. The main editor displays the code for 'node1.js', which uses the MongoClient to connect to a MongoDB instance at localhost:27018. The code defines a collection 'studentmarks' and finds documents where 'maths_marks' is greater than or equal to 50. The status bar at the bottom indicates the file is 'node1.js' and the editor is at line 9, column 62.

6) Add a new column(field) for Average for all students.



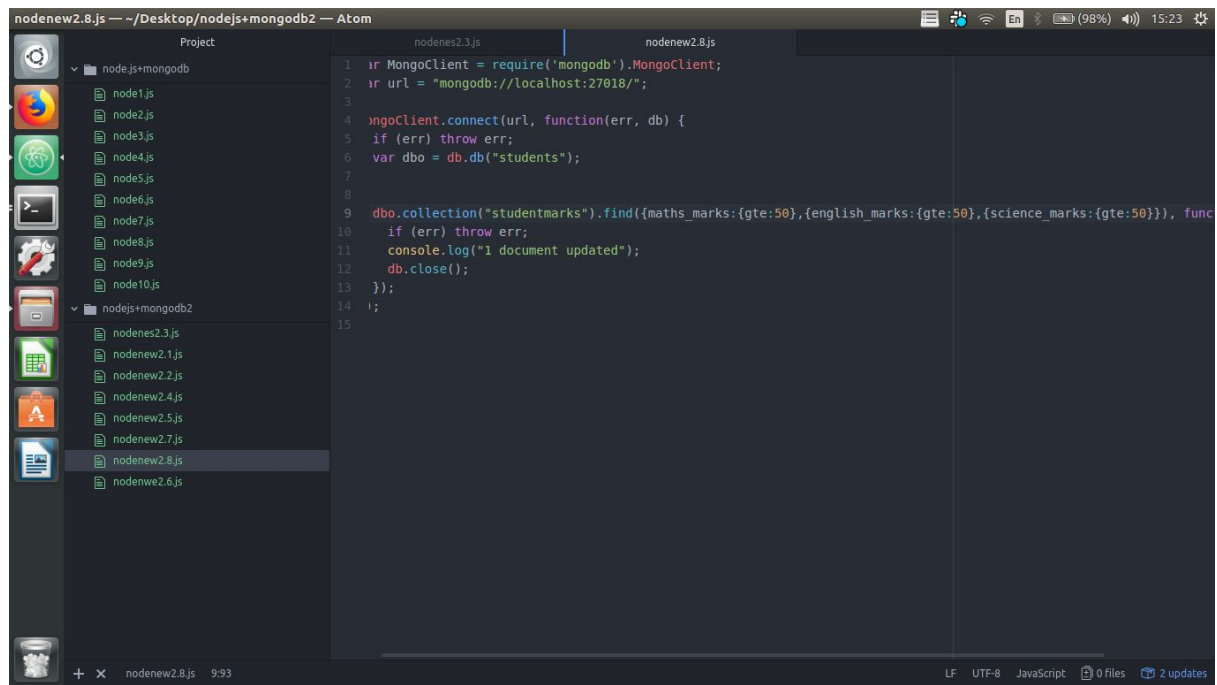
```
1 var MongoClient = require('mongodb').MongoClient;
2 var url = "mongodb://localhost:27017/";
3 MongoClient.connect(url, function(err, db) {
4   if (err) throw err;
5   var dbo = db.db("students");
6   dbo.collection("studentmarks").aggregate([{$addFields: {"average": ""}}]).toArray(function(err, result) {
7     if (err) throw err;
8     console.log(result);
9   });
10 });
11 db.close();
```

7) Update Marks_Science=75 to Lucky .



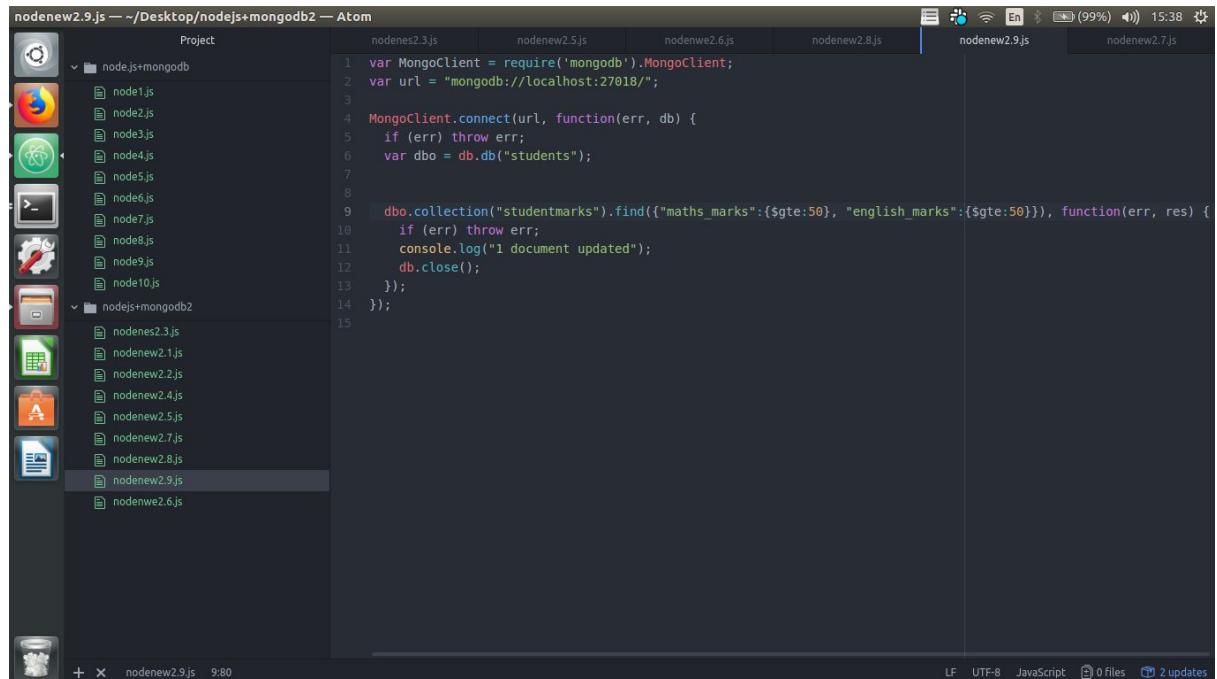
```
1 var MongoClient = require('mongodb').MongoClient;
2 var url = "mongodb://127.0.0.1:27017/";
3
4 MongoClient.connect(url, function(err, db) {
5   if (err) throw err;
6   var dbo = db.db("students");
7   var myquery = { "name": "Lucky" };
8   var newvalues = { $set: { "science_marks": 75 } };
9   dbo.collection("studentmarks").updateOne(myquery, newvalues, function(err, res) {
10     if (err) throw err;
11     console.log("1 document updated");
12     db.close();
13   });
14 });
15
```

8) List the names who got more than 50 marks in all subjects.



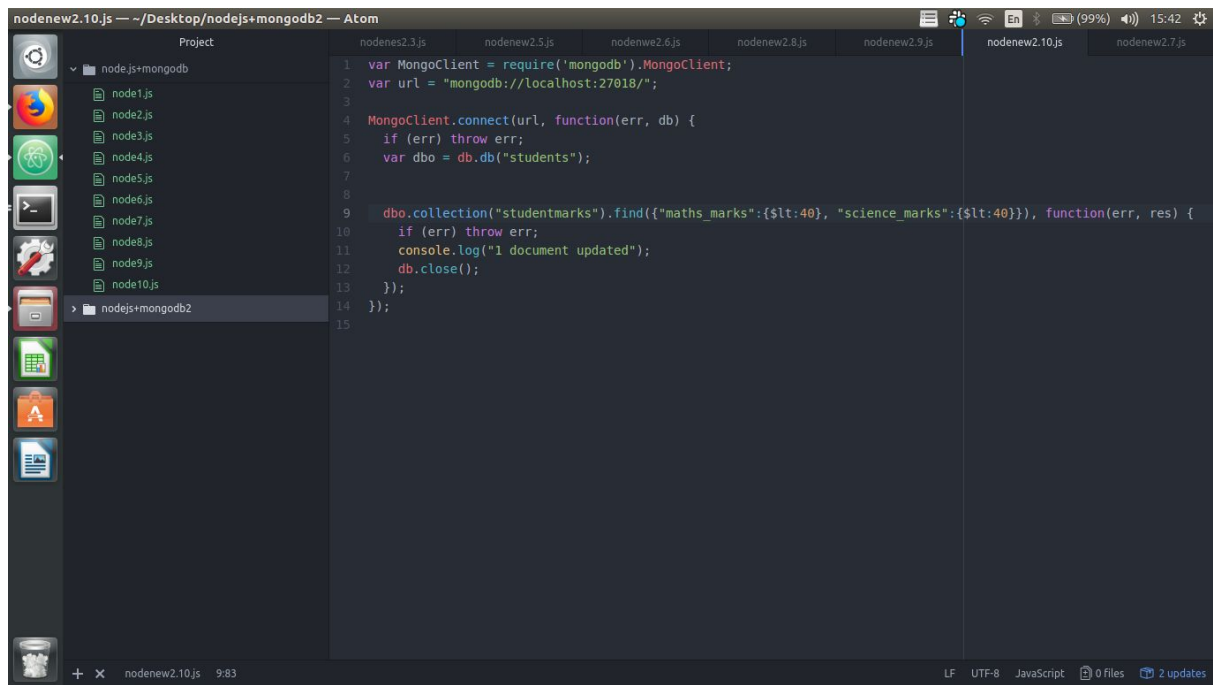
```
1 var MongoClient = require('mongodb').MongoClient;
2 var url = "mongodb://localhost:27018/";
3
4 MongoClient.connect(url, function(err, db) {
5   if (err) throw err;
6   var dbo = db.db("students");
7
8
9   dbo.collection("studentmarks").find({maths_marks:{gte:50},{english_marks:{gte:50},{science_marks:{gte:50}}}, function(err, docs) {
10     if (err) throw err;
11     console.log("1 document updated");
12     db.close();
13   });
14 };
```

9) List the names who got less than 50 marks in Maths subject and more than 50 marks in English



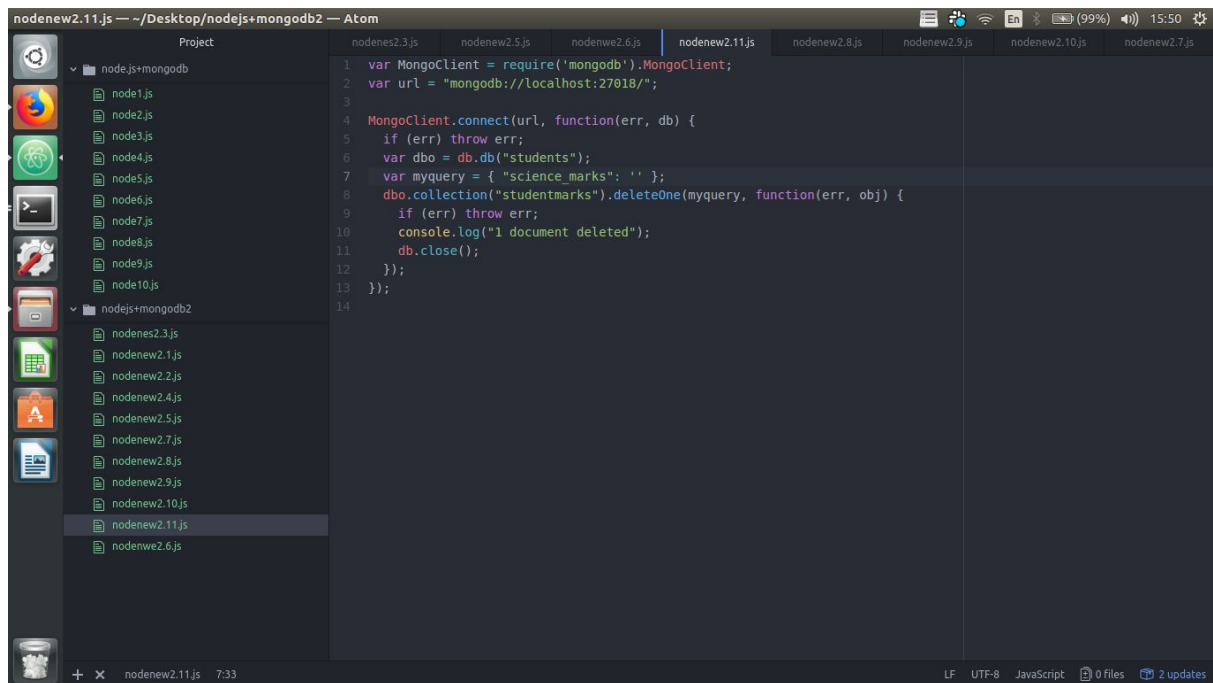
```
1 var MongoClient = require('mongodb').MongoClient;
2 var url = "mongodb://localhost:27018/";
3
4 MongoClient.connect(url, function(err, db) {
5   if (err) throw err;
6   var dbo = db.db("students");
7
8
9   dbo.collection("studentmarks").find({"maths_marks":{$lt:50}, "english_marks":{$gte:50}}, function(err, res) {
10     if (err) throw err;
11     console.log("1 document updated");
12     db.close();
13   });
14 };
```

10) List the names who got less than 40 in both Maths and Science.



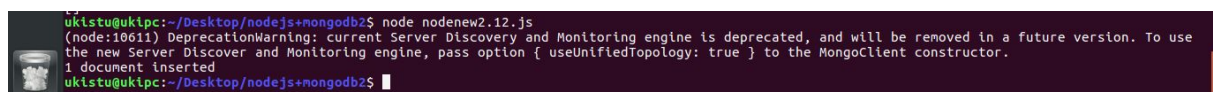
```
1 var MongoClient = require('mongodb').MongoClient;
2 var url = "mongodb://localhost:27018/";
3
4 MongoClient.connect(url, function(err, db) {
5   if (err) throw err;
6   var dbo = db.db("students");
7
8
9   dbo.collection("studentmarks").find({"maths_marks":{$lt:40}, "science_marks":{$lt:40}}).function(err, res) {
10     if (err) throw err;
11     console.log("1 document updated");
12     db.close();
13   });
14 });
15
```

11) Remove Science column/field for Raam



```
1 var MongoClient = require('mongodb').MongoClient;
2 var url = "mongodb://localhost:27018/";
3
4 MongoClient.connect(url, function(err, db) {
5   if (err) throw err;
6   var dbo = db.db("students");
7   var myquery = { "science_marks": '' };
8   dbo.collection("studentmarks").deleteOne(myquery, function(err, obj) {
9     if (err) throw err;
10    console.log("1 document deleted");
11    db.close();
12  });
13 });
14
```

12) Update John's Math mark as 87 and English mark as 23, if john not available upsert.



```
ukistu@ukipc:~/Desktop/nodejs+mongodb2$ node nodenew2.12.js
(node:10611) 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
ukistu@ukipc:~/Desktop/nodejs+mongodb2$
```

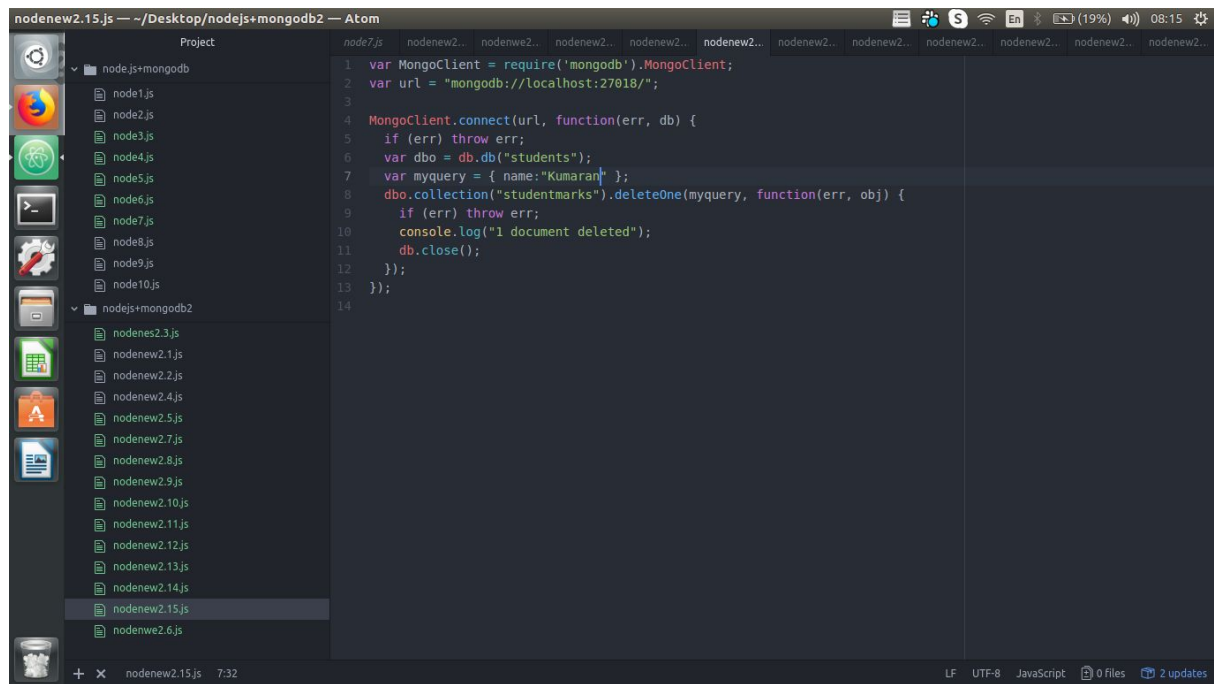
```
1 var MongoClient = require('mongodb').MongoClient;
2 var url = "mongodb://localhost:27017/";
3
4 MongoClient.connect(url, function(err, db) {
5   if (err) throw err;
6   var dbo = db.db("students");
7   var myobj = { name: "John", "maths_marks": 87, "english_marks": 23 };
8   dbo.collection("studentmarks").insertOne(myobj, function(err, res) {
9     if (err) throw err;
10    console.log("1 document inserted");
11    db.close();
12  });
13 });
14
```

13) Rename the english_marks column/field for John to science_marks

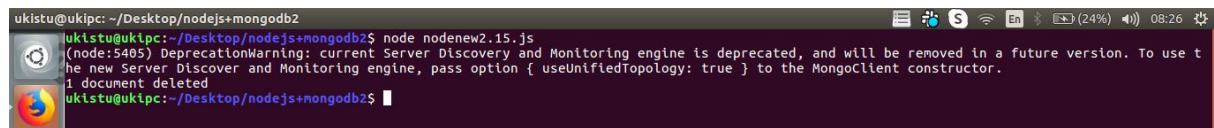
```
1 var MongoClient = require('mongodb').MongoClient;
2 var url = "mongodb://localhost:27017/";
3 MongoClient.connect(url, function(err, db) {
4   if (err) throw err;
5   var dbo = db.db("students");
6   var myquery = {name:"john"};
7   var newvalues = { $rename: {"englishmarks":"sciencemarks" } };
8   dbo.collection("studentmarks").updateOne(myquery, newvalues, function(err, res) {
9     if (err) throw err;
10    console.log("1 document updated");
11    db.close();
12  });
13 });
14
```

```
ukistu@ukipic:~/Desktop/nodejs+mongodb2$ node nodenew2.14.js
(node:5502) 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 updated
ukistu@ukipic:~/Desktop/nodejs+mongodb2$
```

14) Remove Kumaran's document from collection



```
1 var MongoClient = require('mongodb').MongoClient;
2 var url = "mongodb://localhost:27018/";
3
4 MongoClient.connect(url, function(err, db) {
5   if (err) throw err;
6   var dbo = db.db("students");
7   var myquery = { name:"Kumaran" };
8   dbo.collection("studentmarks").deleteOne(myquery, function(err, obj) {
9     if (err) throw err;
10    console.log("1 document deleted");
11    db.close();
12  });
13 });
14
```



```
ukistu@ukipc: ~/Desktop/nodejs+mongodb2
ukistu@ukipc:~/Desktop/nodejs+mongodb2$ node nodenew2.15.js
(node:5405) 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
ukistu@ukipc:~/Desktop/nodejs+mongodb2$
```