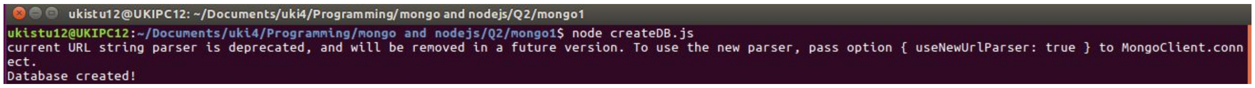
Node.js Exercise

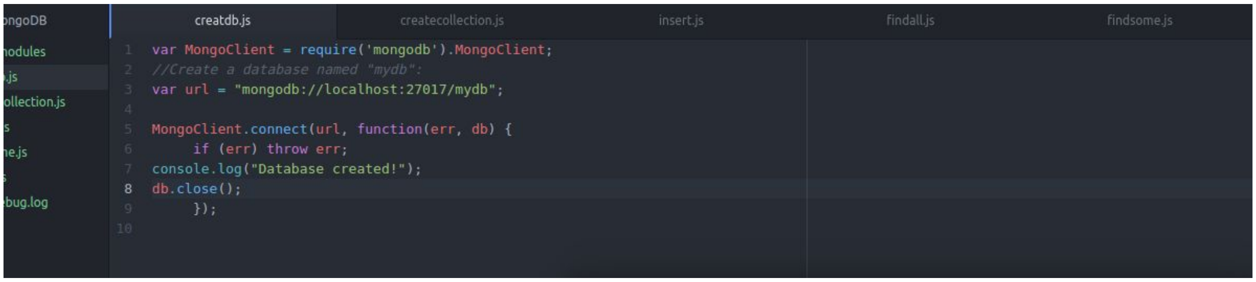
1 Create a MongoClient object, then specify a connection URL with the correct ip address and the name of the database you want to create.

1) Create a Database called music.



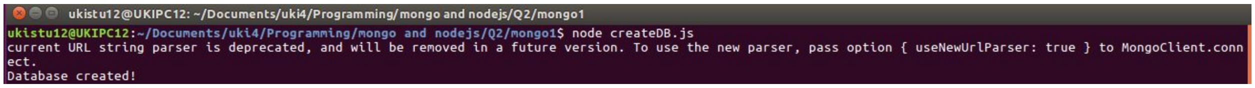
ukistu12@UKIPC12: ~/Documents/uki4/Programming/mongo and nodejs/Q2/mongo1

```
ukistu12@UKIPC12:~/Documents/uki4/Programming/mongo and nodejs/Q2/mongo1$ node createDB.js
current URL string parser is deprecated, and will be removed in a future version. To use the new parser, pass option { useNewUrlParser: true } to MongoClient.conn
ect.
Database created!
```



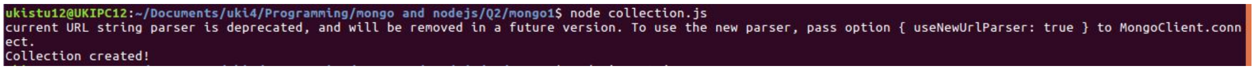
```
var MongoClient = require('mongodb').MongoClient;
//Create a database named "mydb":
var url = "mongodb://localhost:27017/mydb";

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

2) Create a collection called song details.

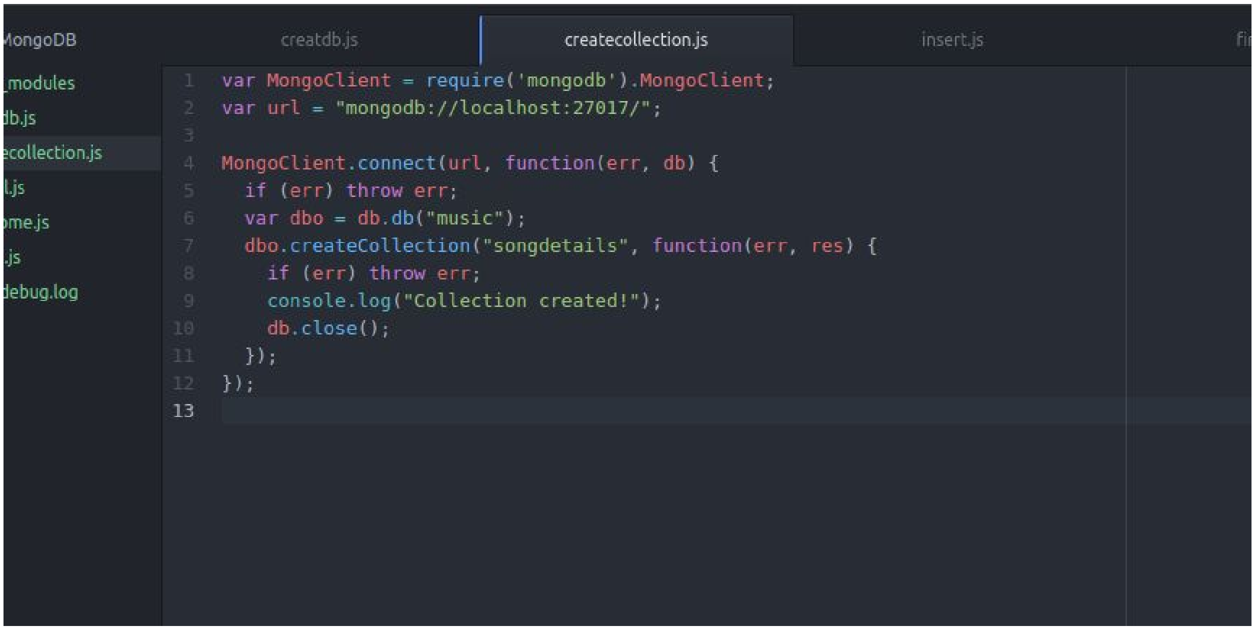


ukistu12@UKIPC12: ~/Documents/uki4/Programming/mongo and nodejs/Q2/mongo1

```
ukistu12@UKIPC12:~/Documents/uki4/Programming/mongo and nodejs/Q2/mongo1$ node createDB.js
current URL string parser is deprecated, and will be removed in a future version. To use the new parser, pass option { useNewUrlParser: true } to MongoClient.conn
ect.
Database created!
```



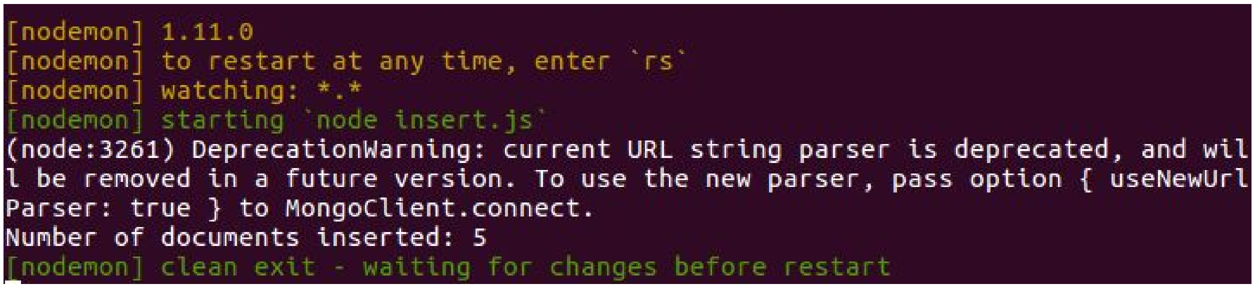
```
ukistu12@UKIPC12:~/Documents/uki4/Programming/mongo and nodejs/Q2/mongo1$ node collection.js
current URL string parser is deprecated, and will be removed in a future version. To use the new parser, pass option { useNewUrlParser: true } to MongoClient.conn
ect.
Collection created!
```



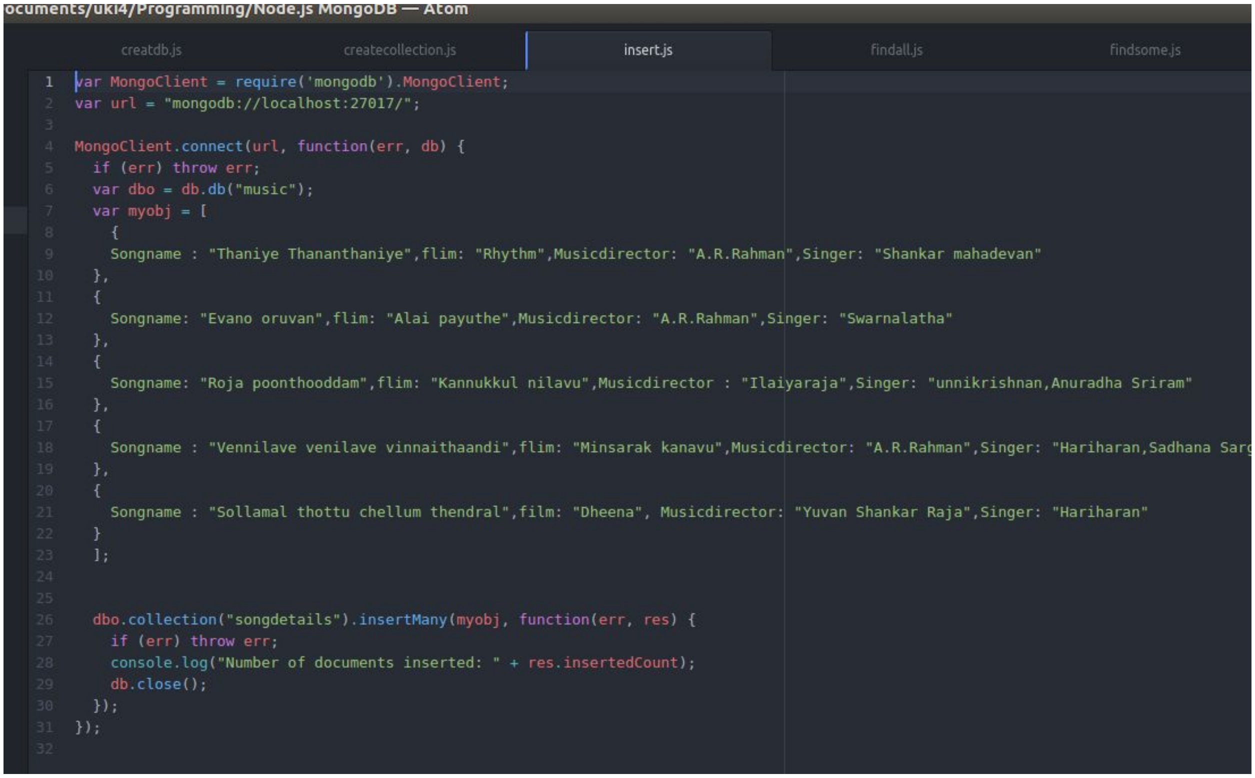
```
var MongoClient = require('mongodb').MongoClient;
var url = "mongodb://localhost:27017/";

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

-

3) Create the above 5 song documents.

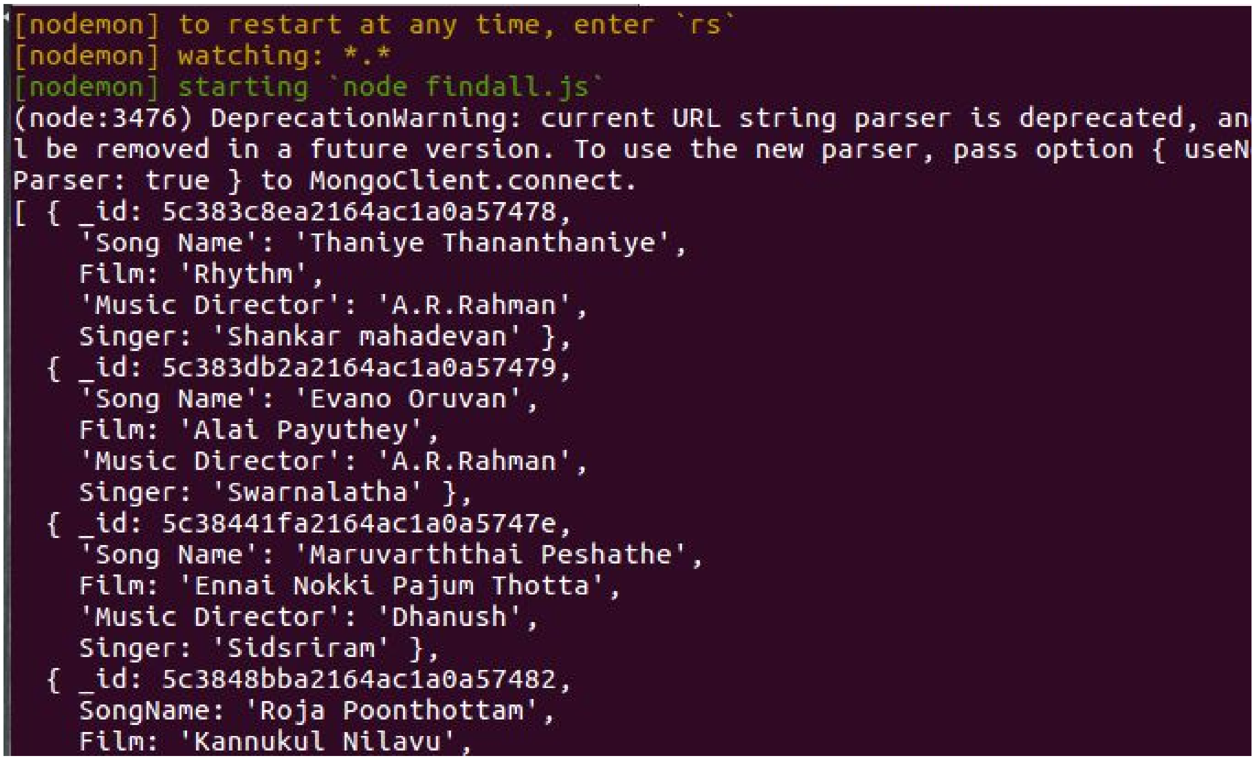


```
[nodemon] 1.11.0
[nodemon] to restart at any time, enter `rs`
[nodemon] watching: *.*
[nodemon] starting `node insert.js`
(node:3261) DeprecationWarning: current URL string parser is deprecated, and wil
l be removed in a future version. To use the new parser, pass option { useNewUrl
Parser: true } to MongoClient.connect.
Number of documents inserted: 5
[nodemon] clean exit - waiting for changes before restart
```

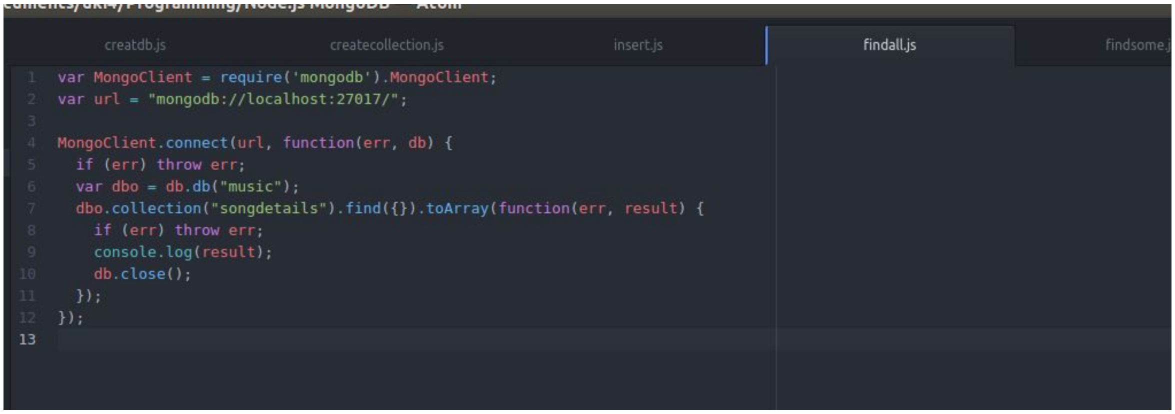


```
var MongoClient = require('mongodb').MongoClient;
var url = "mongodb://localhost:27017/";

MongoClient.connect(url, function(err, db) {
  if (err) throw err;
  var dbo = db.db("music");
  var myobj = [
    {
    Songname : "Thaniye Thananthaniye",flim: "Rhythm",Musicdirector: "A.R.Rahman",Singer: "Shankar mahadevan"
  },
  {
    Songname: "Evano oruvan",flim: "Alai payuthe",Musicdirector: "A.R.Rahman",Singer: "Swarnalatha"
  },
  {
    Songname: "Roja poonthooddam",flim: "Kannukkul nilavu",Musicdirector : "Ilaiyaraja",Singer: "unnikrishnan,Anuradha Sriram"
  },
  {
    Songname : "Vennilave venilave vinnaithaandi",flim: "Minsarak kanavu",Musicdirector: "A.R.Rahman",Singer: "Hariharan,Sadhana Sarg
  },
  {
    Songname : "Sollamal thottu chellum thendral",film: "Dheena", Musicdirector: "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();
  });
});

```

4) List all documents created.

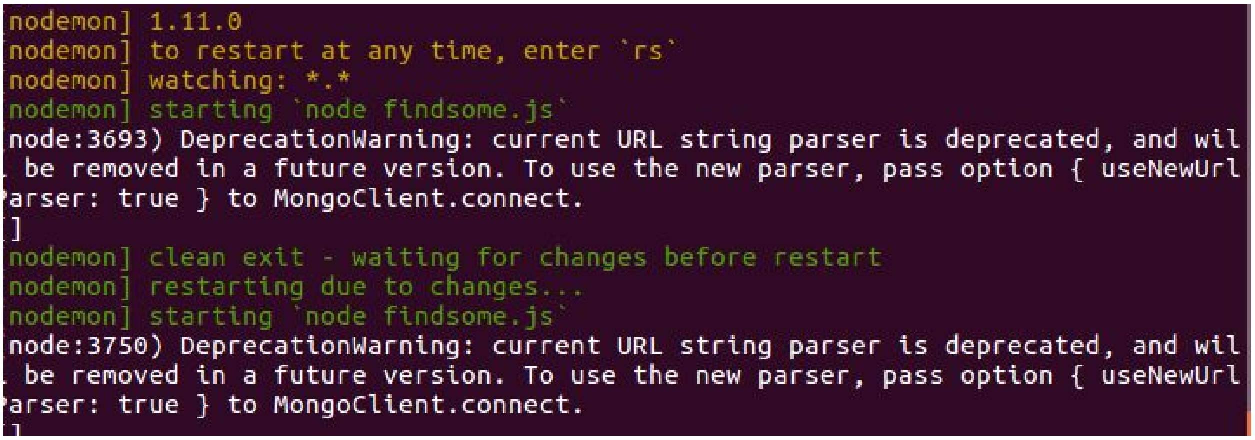


```
[nodemon] to restart at any time, enter `rs`
[nodemon] watching: *.*
[nodemon] starting `node findall.js`
(node:3476) DeprecationWarning: current URL string parser is deprecated, an
l be removed in a future version. To use the new parser, pass option { useN
Parser: true } to MongoClient.connect.
[ { _id: 5c383c8ea2164ac1a0a57478,
    'Song Name': 'Thaniye Thananthaniye',
    Film: 'Rhythm',
    'Music Director': 'A.R.Rahman',
    Singer: 'Shankar mahadevan' },
  { _id: 5c383db2a2164ac1a0a57479,
    'Song Name': 'Evano Oruvan',
    Film: 'Alai Payuthey',
    'Music Director': 'A.R.Rahman',
    Singer: 'Swarnalatha' },
  { _id: 5c38441fa2164ac1a0a5747e,
    'Song Name': 'Maruvarththai Peshathe',
    Film: 'Ennai Nokki Pajum Thotta',
    'Music Director': 'Dhanush',
    Singer: 'Sidsriram' },
  { _id: 5c3848bba2164ac1a0a57482,
    SongName: 'Roja Poonthottam',
    Film: 'Kannukul Nilavu',
```

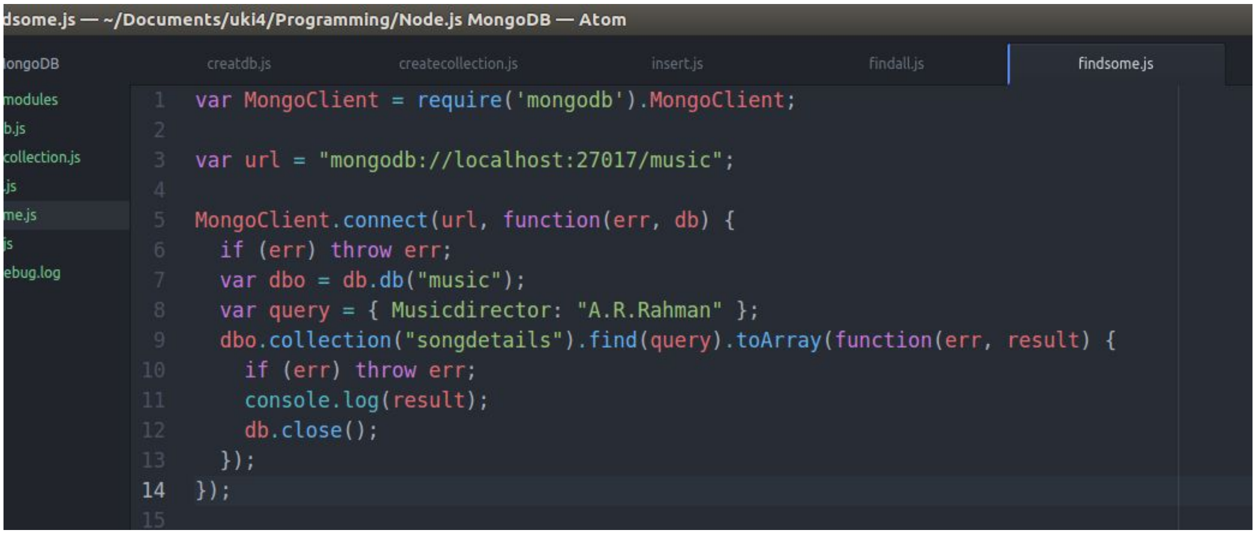


```
var MongoClient = require('mongodb').MongoClient;
var url = "mongodb://localhost:27017/";

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.



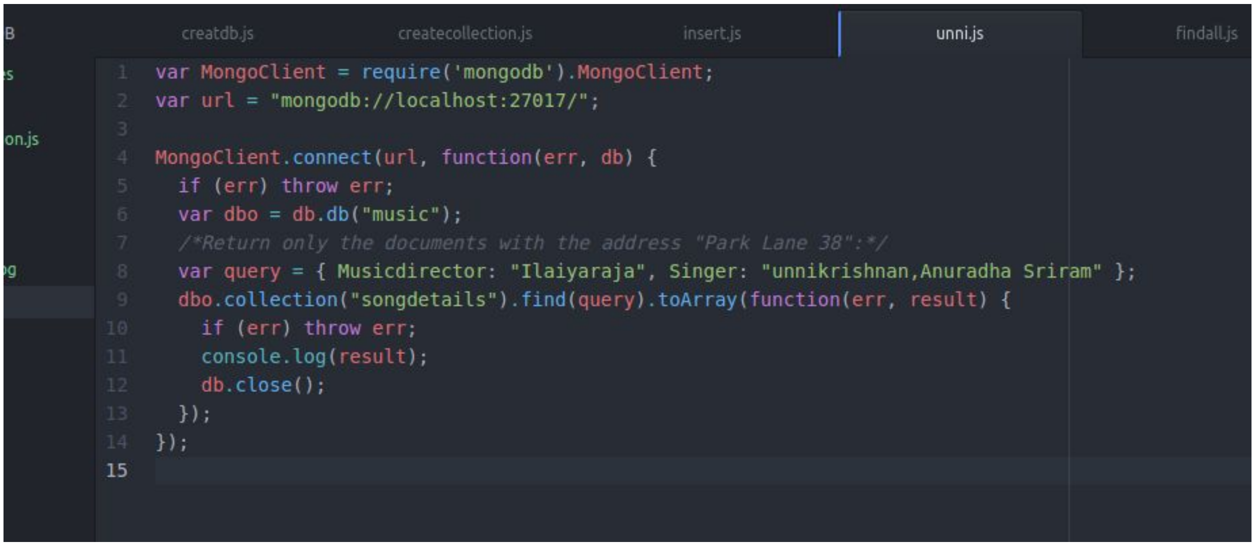
```
nodemon] 1.11.0
nodemon] to restart at any time, enter `rs`
nodemon] watching: *.*
nodemon] starting `node findsome.js`
node:3693) DeprecationWarning: current URL string parser is deprecated, and wil
. be removed in a future version. To use the new parser, pass option { useNewUrl
'arser: true } to MongoClient.connect.
]
nodemon] clean exit - waiting for changes before restart
nodemon] restarting due to changes...
nodemon] starting `node findsome.js`
node:3750) DeprecationWarning: current URL string parser is deprecated, and wil
. be removed in a future version. To use the new parser, pass option { useNewUrl
'arser: true } to MongoClient.connect.
```



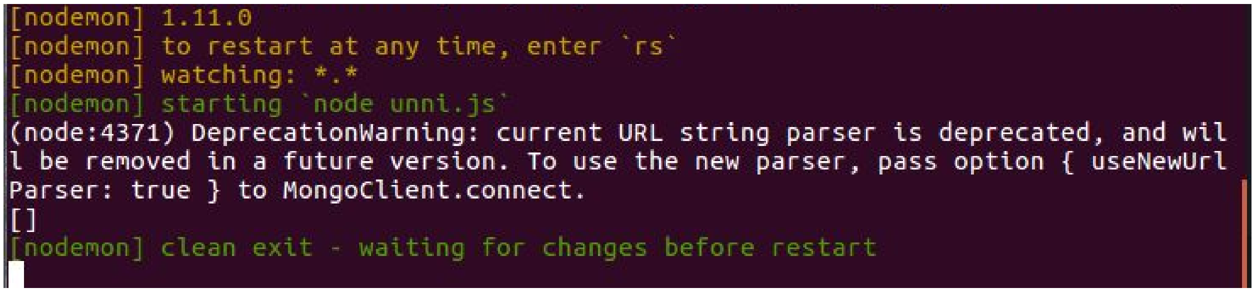
dsome.js — ~/Documents/uki4/Programming/Node.js MongoDB — Atom

```
var MongoClient = require('mongodb').MongoClient;

var url = "mongodb://localhost:27017/music";

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

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

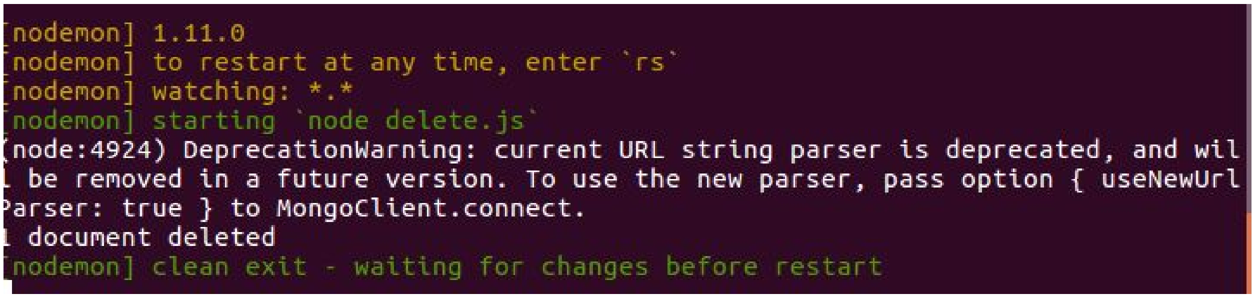


```
var MongoClient = require('mongodb').MongoClient;
var url = "mongodb://localhost:27017/";

MongoClient.connect(url, function(err, db) {
  if (err) throw err;
  var dbo = db.db("music");
  /*Return only the documents with the address "Park Lane 38":*/
  var query = { Musicdirector: "Ilaiyaraja", Singer: "unnikrishnan,Anuradha Sriram" };
  dbo.collection("songdetails").find(query).toArray(function(err, result) {
    if (err) throw err;
    console.log(result);
    db.close();
  });
});

```

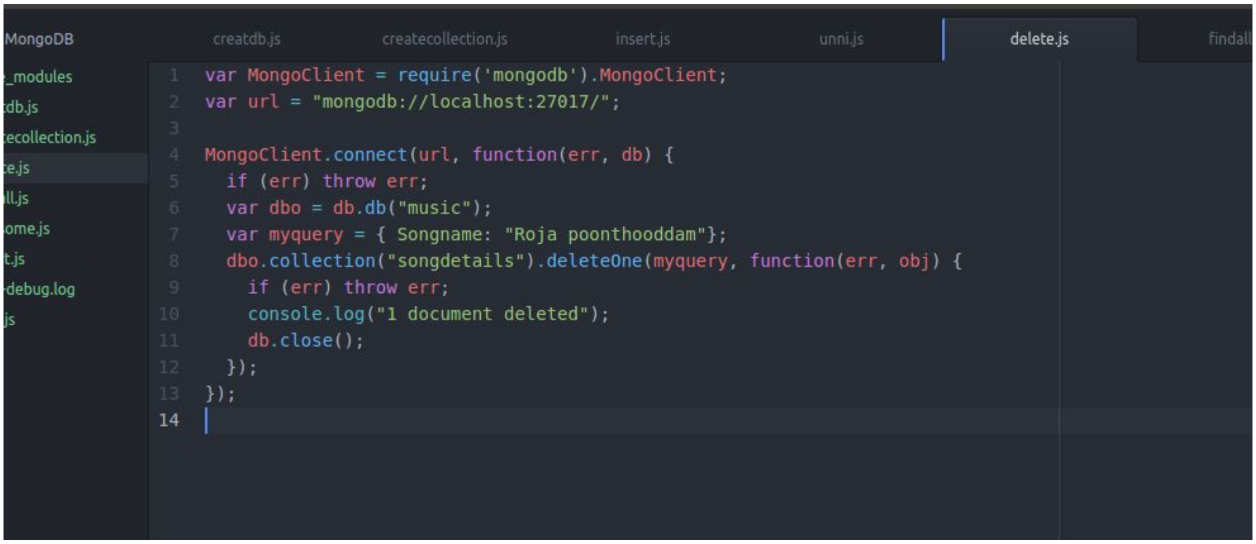


```
[nodemon] 1.11.0
[nodemon] to restart at any time, enter `rs`
[nodemon] watching: *.*
[nodemon] starting `node unni.js`
(node:4371) DeprecationWarning: current URL string parser is deprecated, and wil
l be removed in a future version. To use the new parser, pass option { useNewUrl
Parser: true } to MongoClient.connect.
[]
[nodemon] clean exit - waiting for changes before restart
```

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

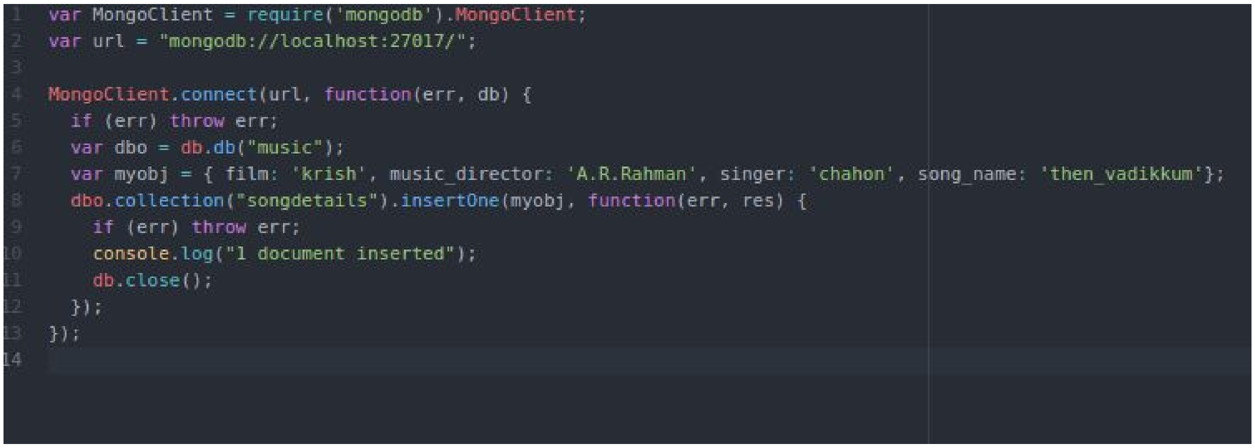


```
[nodemon] 1.11.0
[nodemon] to restart at any time, enter `rs`
[nodemon] watching: *.*
[nodemon] starting `node delete.js`
(node:4924) DeprecationWarning: current URL string parser is deprecated, and wil
l be removed in a future version. To use the new parser, pass option { useNewUrl
Parser: true } to MongoClient.connect.
1 document deleted
[nodemon] clean exit - waiting for changes before restart
```

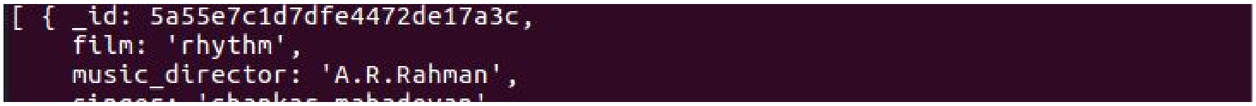


```
var MongoClient = require('mongodb').MongoClient;
var url = "mongodb://localhost:27017/";

MongoClient.connect(url, function(err, db) {
  if (err) throw err;
  var dbo = db.db("music");
  var myquery = { Songname: "Roja poonthooddam"};
  dbo.collection("songdetails").deleteOne(myquery, function(err, obj) {
    if (err) throw err;
    console.log("1 document deleted");
    db.close();
  });
});

```

8) Add new song which is your favourite.

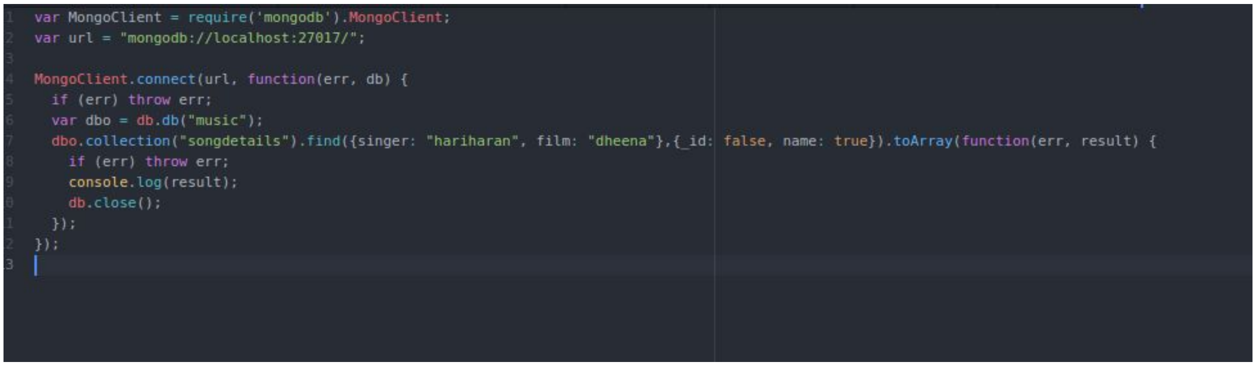


```
var MongoClient = require('mongodb').MongoClient;
var url = "mongodb://localhost:27017/";

MongoClient.connect(url, function(err, db) {
  if (err) throw err;
  var dbo = db.db("music");
  var myobj = { film: 'krish', music_director: 'A.R.Rahman', singer: 'chahon', song_name: 'then_vadikkum'};
  dbo.collection("songdetails").insertOne(myobj, function(err, res) {
    if (err) throw err;
    console.log("1 document inserted");
    db.close();
  });
});

```

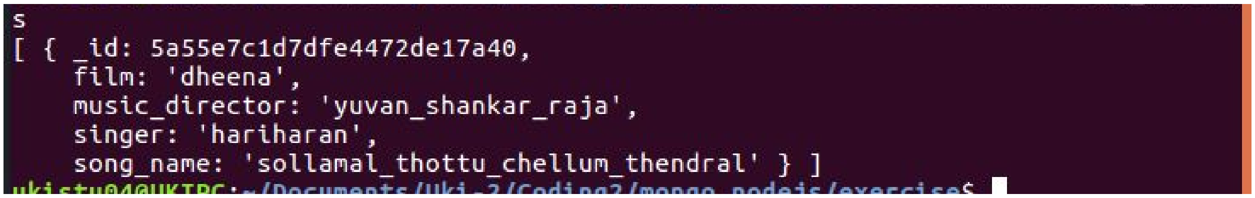


```
[ { _id: 5a55e7c1d7dfe4472de17a3c,
    film: 'rhythm',
    music_director: 'A.R.Rahman',
    singer: 'shankar_mahadevan',
```

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

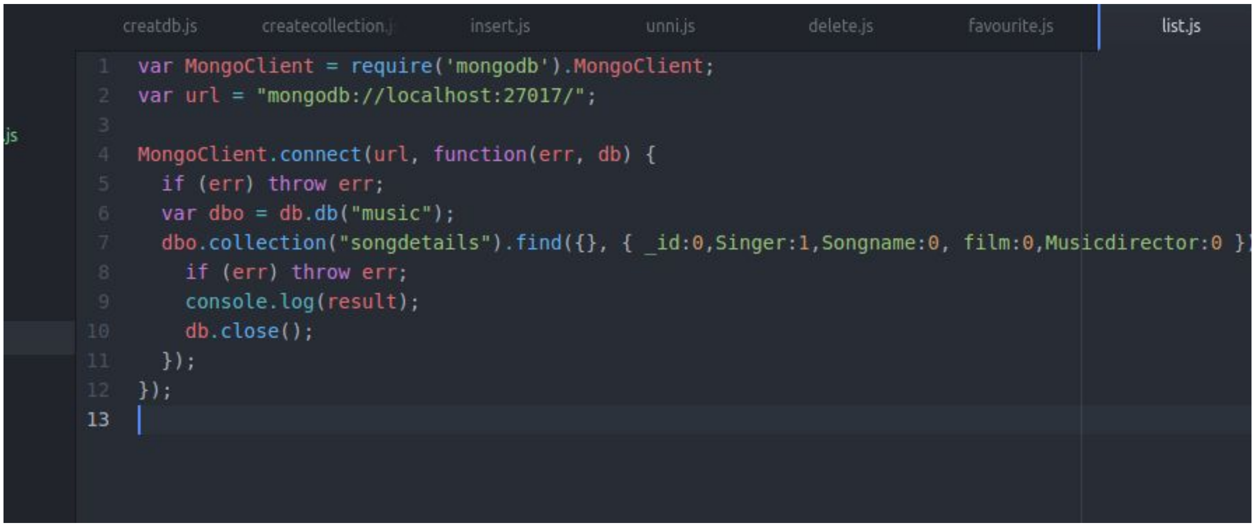


```
var MongoClient = require('mongodb').MongoClient;
var url = "mongodb://localhost:27017/";

MongoClient.connect(url, function(err, db) {
  if (err) throw err;
  var dbo = db.db("music");
  dbo.collection("songdetails").find({singer: "hariharan", film: "dheena"},{_id: false, name: true}).toArray(function(err, result) {
    if (err) throw err;
    console.log(result);
    db.close();
  });
});

```

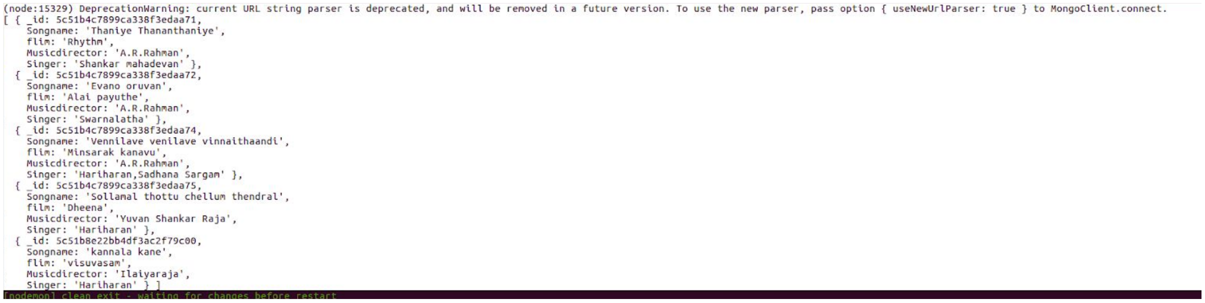


```
s
[ { _id: 5a55e7c1d7dfe4472de17a40,
    film: 'dheena',
    music_director: 'yuvan_shankar_raja',
    singer: 'hariharan',
    song_name: 'sollamal_thottu_chellum_thendral' } ]
ukistu04@UKIPC:~/Documents/Uki-2/Coding2/mongo_nodejs/exercise$
```

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



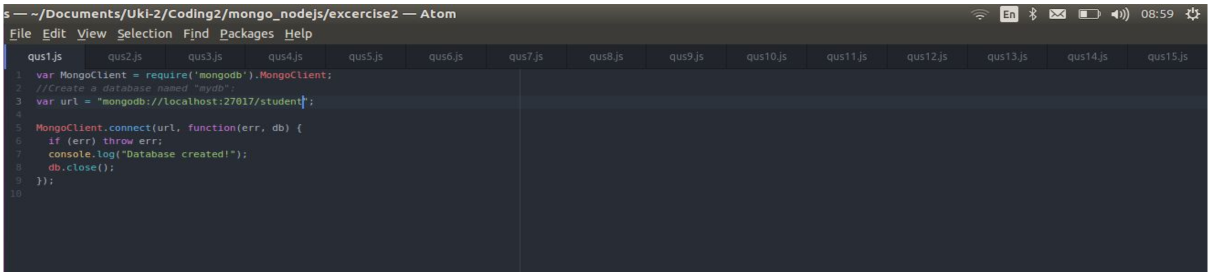
```
var MongoClient = require('mongodb').MongoClient;
var url = "mongodb://localhost:27017/";

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



```
(node:15329) DeprecationWarning: current URL string parser is deprecated, and will be removed in a future version. To use the new parser, pass option { useNewUrlParser: true } to MongoClient.connect.
[ { _id: 5c51b4c7899ca338f3edaa71,
    Songname: 'Thaniye Thananthaniye',
    flim: 'Rhythm',
    Musicdirector: 'A.R.Rahman',
    Singer: 'Shankar mahadevan' },
  { _id: 5c51b4c7899ca338f3edaa72,
    Songname: 'Evano oruvan',
    flim: 'Alai payuthe',
    Musicdirector: 'A.R.Rahman',
    Singer: 'Swarnalatha' },
  { _id: 5c51b4c7899ca338f3edaa74,
    Songname: 'Vennilave venilave vinnaithaandi',
    flim: 'Minsarak kanavu',
    Musicdirector: 'A.R.Rahman',
    Singer: 'Hariharan,Sadhana Sargam' },
  { _id: 5c51b4c7899ca338f3edaa75,
    Songname: 'Sollamal thottu chellum thendral',
    film: 'Dheena',
    Musicdirector: 'Yuvan Shankar Raja',
    Singer: 'Hariharan' },
  { _id: 5c51b8e22bb4df3ac2f79c00,
    Songname: 'kannala kane',
    flim: 'visuvasam',
    Musicdirector: 'Ilaiyaraja',
    Singer: 'Hariharan' } ]
[nodemon] clean exit - waiting for changes before restart
```

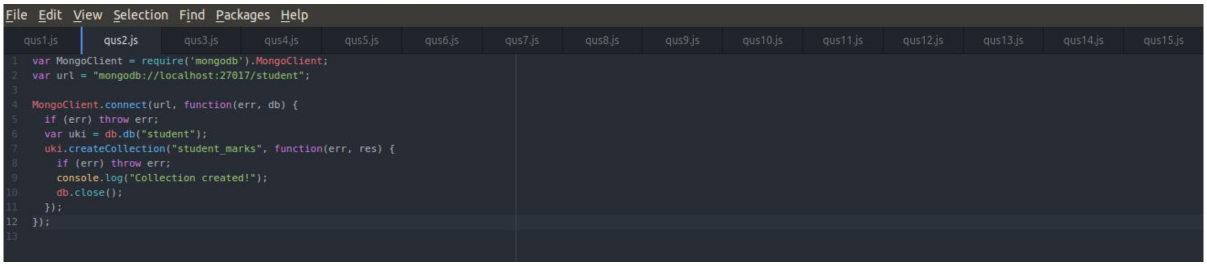
2. Complete mongodb ex1 and mongodb ex2 with the mongoclient object you just created in Node.js.

1) Create a Database called student



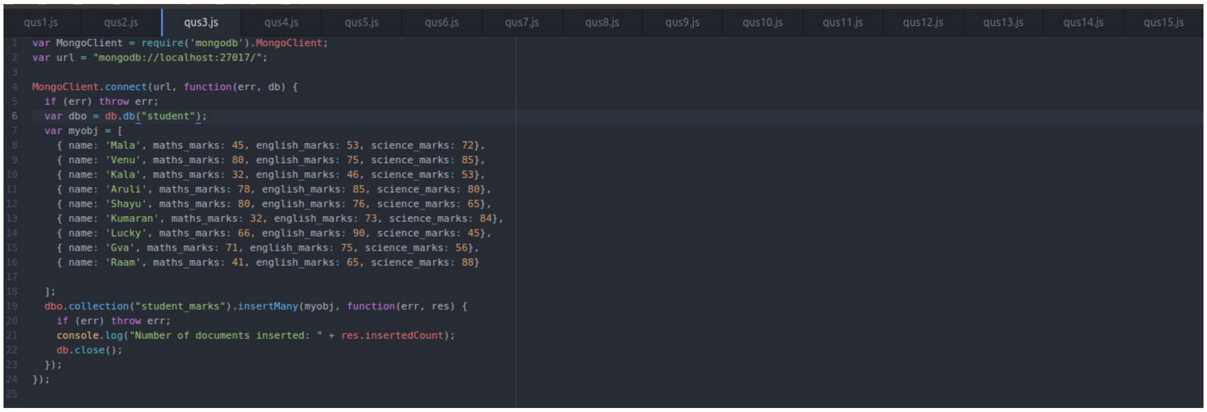
```
var MongoClient = require('mongodb').MongoClient;
//Create a database named "mydb":
var url = "mongodb://localhost:27017/student";

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

2) Create a collection called student marks



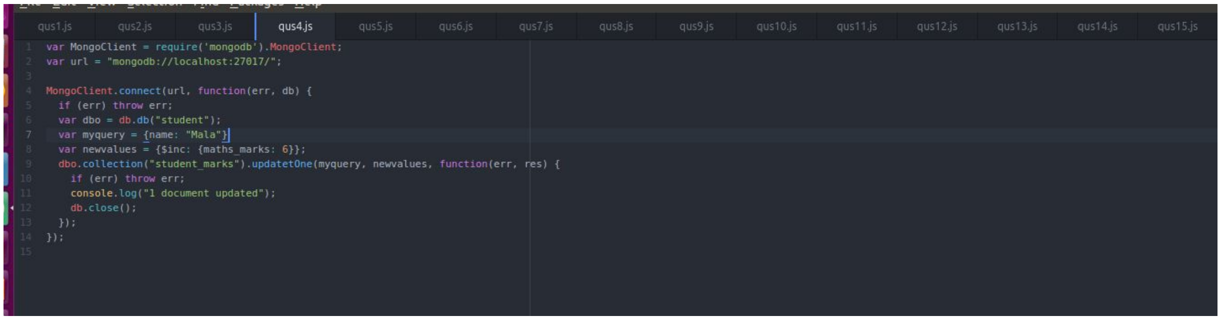
```
var MongoClient = require('mongodb').MongoClient;
var url = "mongodb://localhost:27017/student";

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

3) Create the documents listed in above table.



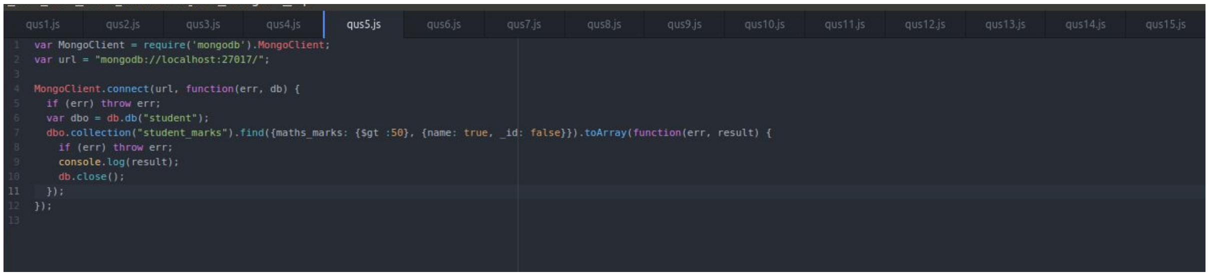
```
var MongoClient = require('mongodb').MongoClient;
var url = "mongodb://localhost:27017/";

MongoClient.connect(url, function(err, db) {
  if (err) throw err;
  var dbo = db.db("student");
  var myobj = [
    { name: 'Mala', maths_marks: 45, english_marks: 53, science_marks: 72},
    { name: 'Venu', maths_marks: 80, english_marks: 75, science_marks: 85},
    { name: 'Kala', maths_marks: 32, english_marks: 46, science_marks: 53},
    { name: 'Aruli', maths_marks: 78, english_marks: 85, science_marks: 80},
    { name: 'Shayu', maths_marks: 80, english_marks: 76, science_marks: 65},
    { name: 'Kumaran', maths_marks: 32, english_marks: 73, science_marks: 84},
    { name: 'Lucky', maths_marks: 66, english_marks: 90, science_marks: 45},
    { name: 'Gva', maths_marks: 71, english_marks: 75, science_marks: 56},
    { name: 'Raam', maths_marks: 41, english_marks: 65, science_marks: 88}

  ];
  dbo.collection("student_marks").insertMany(myobj, function(err, res) {
    if (err) throw err;
    console.log("Number of documents inserted: " + res.insertedCount);
    db.close();
  });
});
```

4) Increase the maths marks of Mala by 6 marks



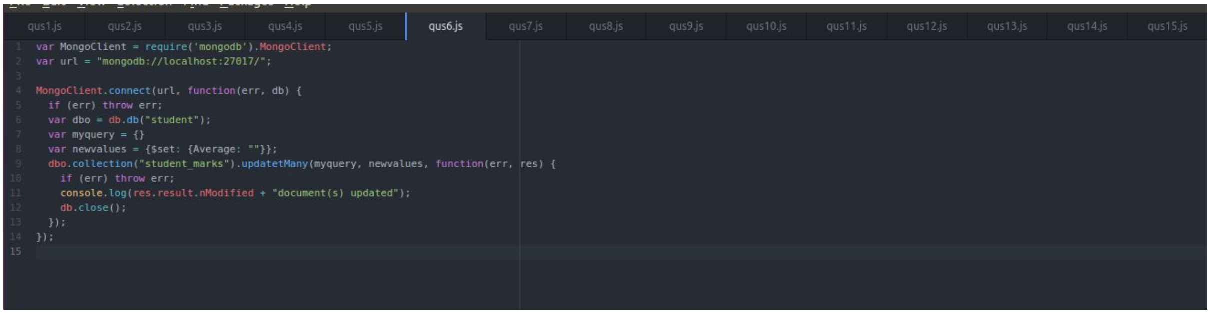
```
var MongoClient = require('mongodb').MongoClient;
var url = "mongodb://localhost:27017/";

MongoClient.connect(url, function(err, db) {
  if (err) throw err;
  var dbo = db.db("student");
  var myquery = {name: "Mala"}
  var newvalues = {$inc: {maths_marks: 6}};
  dbo.collection("student_marks").updatetOne(myquery, newvalues, function(err, res) {
    if (err) throw err;
    console.log("1 document updated");
    db.close();
  });
});
```

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



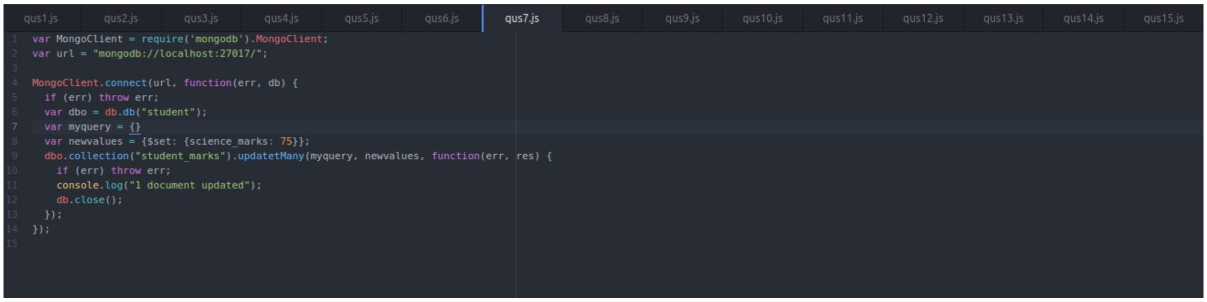
```
var MongoClient = require('mongodb').MongoClient;
var url = "mongodb://localhost:27017/";

MongoClient.connect(url, function(err, db) {
  if (err) throw err;
  var dbo = db.db("student");
  dbo.collection("student_marks").find({maths_marks: {$gt :50}, {name: true, _id: false}}).toArray(function(err, result) {
    if (err) throw err;
    console.log(result);
    db.close();
  });
});
```

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



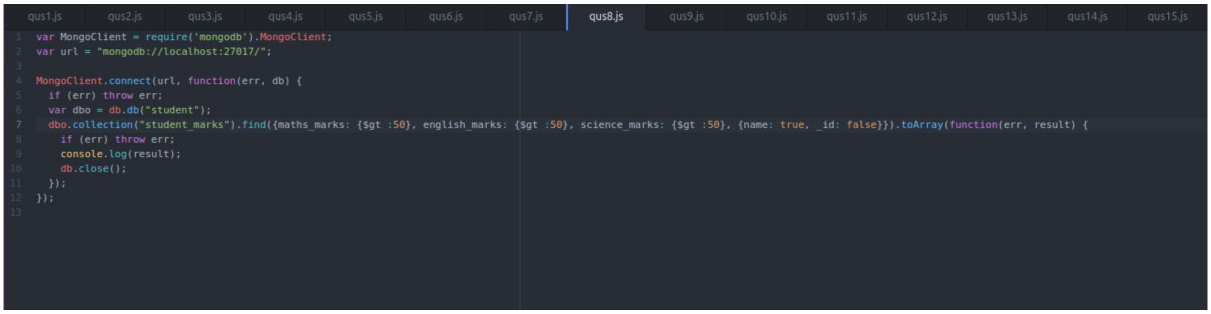
```
var MongoClient = require('mongodb').MongoClient;
var url = "mongodb://localhost:27017/";

MongoClient.connect(url, function(err, db) {
  if (err) throw err;
  var dbo = db.db("student");
  var myquery = {}
  var newvalues = {$set: {Average: ""}};
  dbo.collection("student_marks").updatetMany(myquery, newvalues, function(err, res) {
    if (err) throw err;
    console.log(res.result.nModified + "document(s) updated");
    db.close();
  });
});
```

7) Update Marks_Science=75 to Lucky .



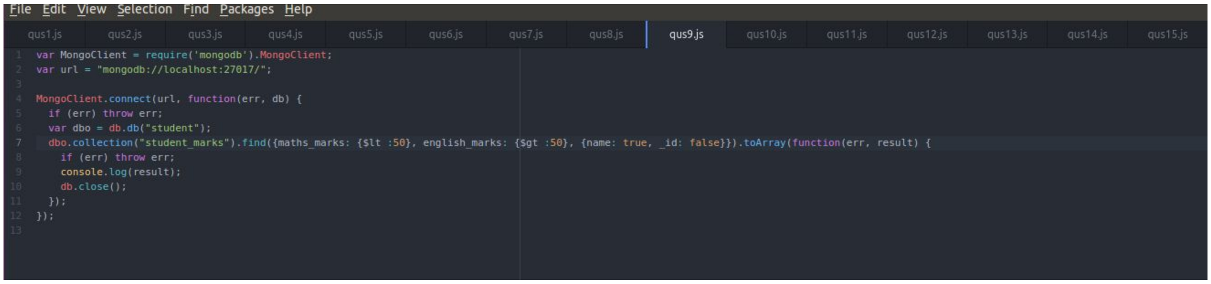
```
var MongoClient = require('mongodb').MongoClient;
var url = "mongodb://localhost:27017/";

MongoClient.connect(url, function(err, db) {
  if (err) throw err;
  var dbo = db.db("student");
  var myquery = {}
  var newvalues = {$set: {science_marks: 75}};
  dbo.collection("student_marks").updatetMany(myquery, newvalues, function(err, res) {
    if (err) throw err;
    console.log("1 document updated");
    db.close();
  });
});
```

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



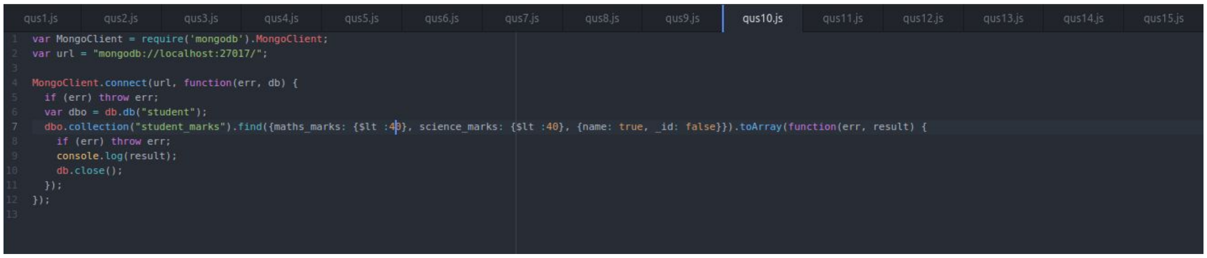
```
var MongoClient = require('mongodb').MongoClient;
var url = "mongodb://localhost:27017/";

MongoClient.connect(url, function(err, db) {
  if (err) throw err;
  var dbo = db.db("student");
  dbo.collection("student_marks").find({maths_marks: {$gt :50}, english_marks: {$gt :50}, science_marks: {$gt :50}, {name: true, _id: false}}).toArray(function(err, result) {
    if (err) throw err;
    console.log(result);
    db.close();
  });
});
```

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



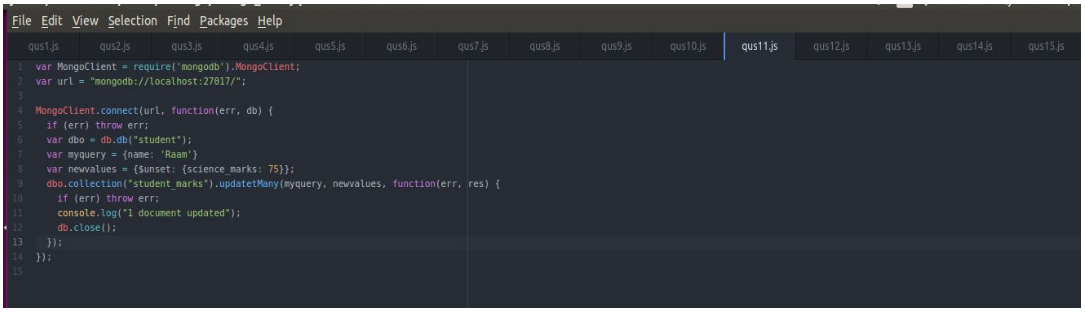
```
var MongoClient = require('mongodb').MongoClient;
var url = "mongodb://localhost:27017/";

MongoClient.connect(url, function(err, db) {
  if (err) throw err;
  var dbo = db.db("student");
  dbo.collection("student_marks").find({maths_marks: {$lt :50}, english_marks: {$gt :50}, {name: true, _id: false}}).toArray(function(err, result) {
    if (err) throw err;
    console.log(result);
    db.close();
  });
});
```

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



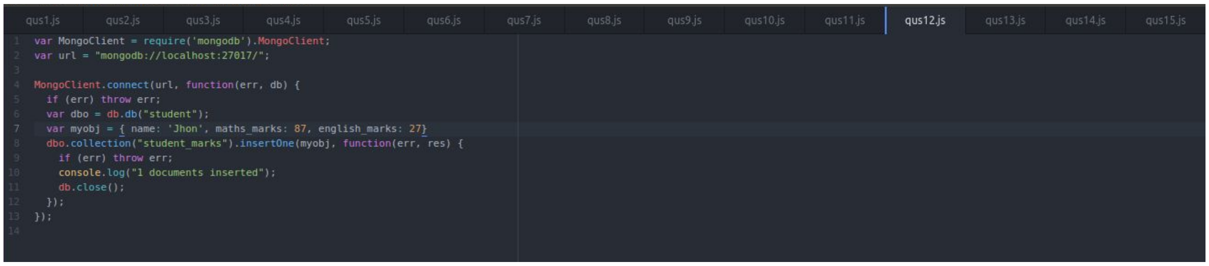
```
var MongoClient = require('mongodb').MongoClient;
var url = "mongodb://localhost:27017/";

MongoClient.connect(url, function(err, db) {
  if (err) throw err;
  var dbo = db.db("student");
  dbo.collection("student_marks").find({maths_marks: {$lt :40}, science_marks: {$lt :40}, {name: true, _id: false}}).toArray(function(err, result) {
    if (err) throw err;
    console.log(result);
    db.close();
  });
});
```

11) Remove Science column/field for Raam.



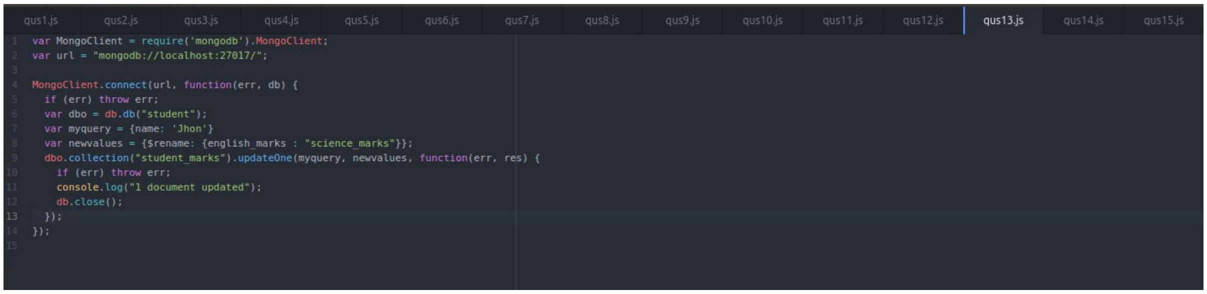
```
var MongoClient = require('mongodb').MongoClient;
var url = "mongodb://localhost:27017/";

MongoClient.connect(url, function(err, db) {
  if (err) throw err;
  var dbo = db.db("student");
  var myquery = {name: 'Raam'}
  var newvalues = {$unset: {science_marks: 75}};
  dbo.collection("student_marks").updatetMany(myquery, newvalues, function(err, res) {
    if (err) throw err;
    console.log("1 document updated");
    db.close();
  });
});
```

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



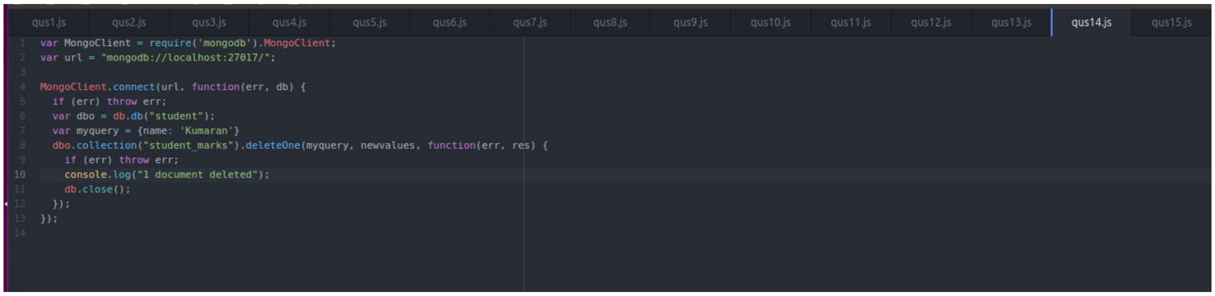
```
var MongoClient = require('mongodb').MongoClient;
var url = "mongodb://localhost:27017/";

MongoClient.connect(url, function(err, db) {
  if (err) throw err;
  var dbo = db.db("student");
  var myobj = { name: 'Jhon', maths_marks: 87, english_marks: 27}
  dbo.collection("student_marks").insertOne(myobj, function(err, res) {
    if (err) throw err;
    console.log("1 documents inserted");
    db.close();
  });
});
```

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



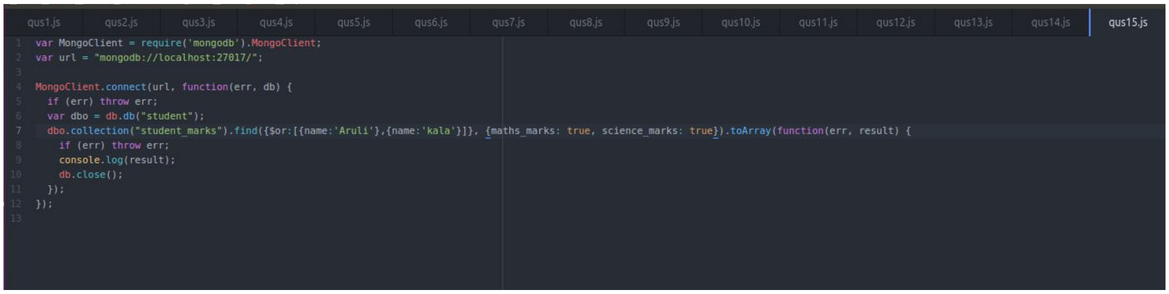
```
var MongoClient = require('mongodb').MongoClient;
var url = "mongodb://localhost:27017/";

MongoClient.connect(url, function(err, db) {
  if (err) throw err;
  var dbo = db.db("student");
  var myquery = {name: 'Jhon'}
  var newvalues = {$rename: {english_marks : "science_marks"}};
  dbo.collection("student_marks").updateOne(myquery, newvalues, function(err, res) {
    if (err) throw err;
    console.log("1 document updated");
    db.close();
  });
});
```

14) Remove Kumaran’s document from collection



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

MongoClient.connect(url, function(err, db) {
  if (err) throw err;
  var dbo = db.db("student");
  var myquery = {name: 'Kumaran'}
  dbo.collection("student_marks").deleteOne(myquery, newvalues, function(err, res) {
    if (err) throw err;
    console.log("1 document deleted");
    db.close();
  });
});
```

15) Find Kala’s or Aruli’s math_marks and science_marks.



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

MongoClient.connect(url, function(err, db) {
  if (err) throw err;
  var dbo = db.db("student");
  dbo.collection("student_marks").find({$or:[{name:'Aruli'},{name:'kala'}]}, {maths_marks: true, science_marks: true}).toArray(function(err, result) {
    if (err) throw err;
    console.log(result);
    db.close();
  });
});
```