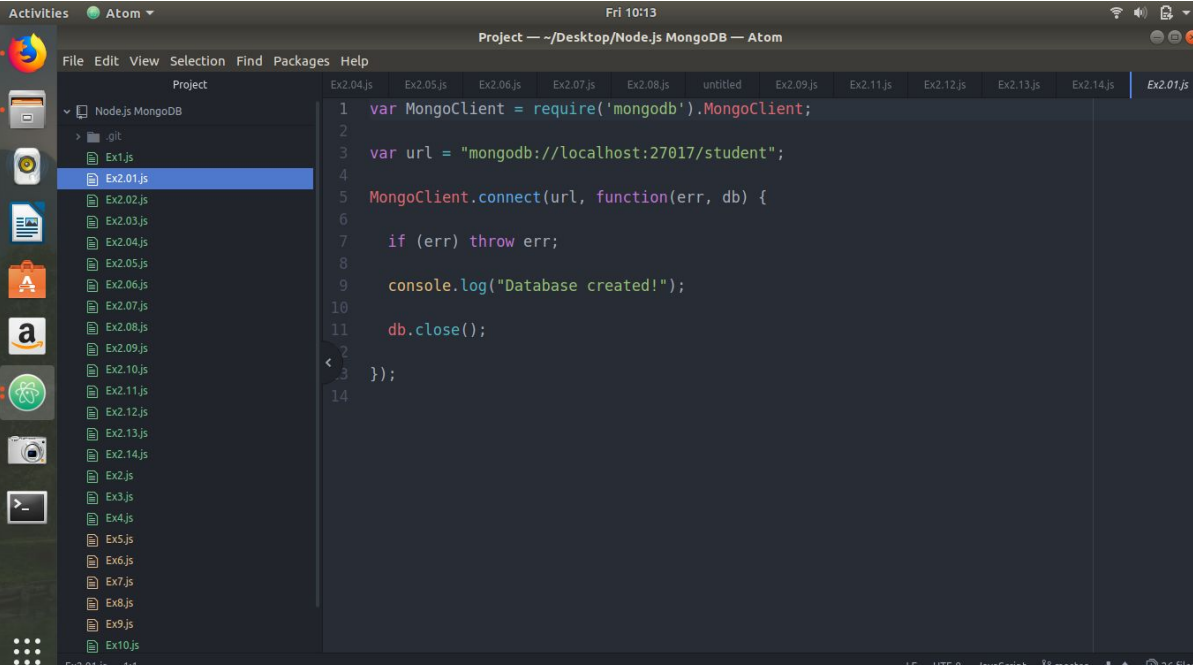


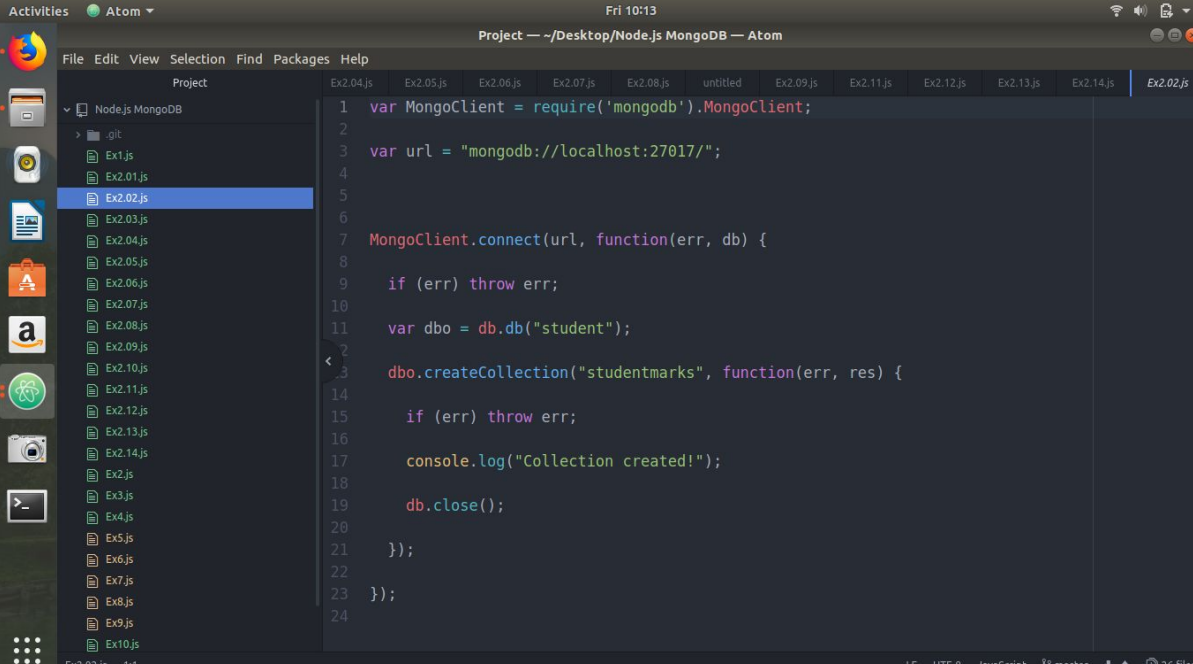
## 1) Create a Database called student



The screenshot shows the Atom editor interface with a project named "Node.js MongoDB". The left sidebar displays a file tree with a .git folder and a series of files named Ex1.js through Ex10.js. The main editor pane shows the code for Ex2.01.js, which is used to create a MongoDB database. The code includes the following logic: it requires the 'mongodb' module, sets a connection URL to 'mongodb://localhost:27017/student', connects to the database, logs a success message, and closes the connection.

```
1 var MongoClient = require('mongodb').MongoClient;
2
3 var url = "mongodb://localhost:27017/student";
4
5 MongoClient.connect(url, function(err, db) {
6
7   if (err) throw err;
8
9   console.log("Database created!");
10
11   db.close();
12
13 });
```

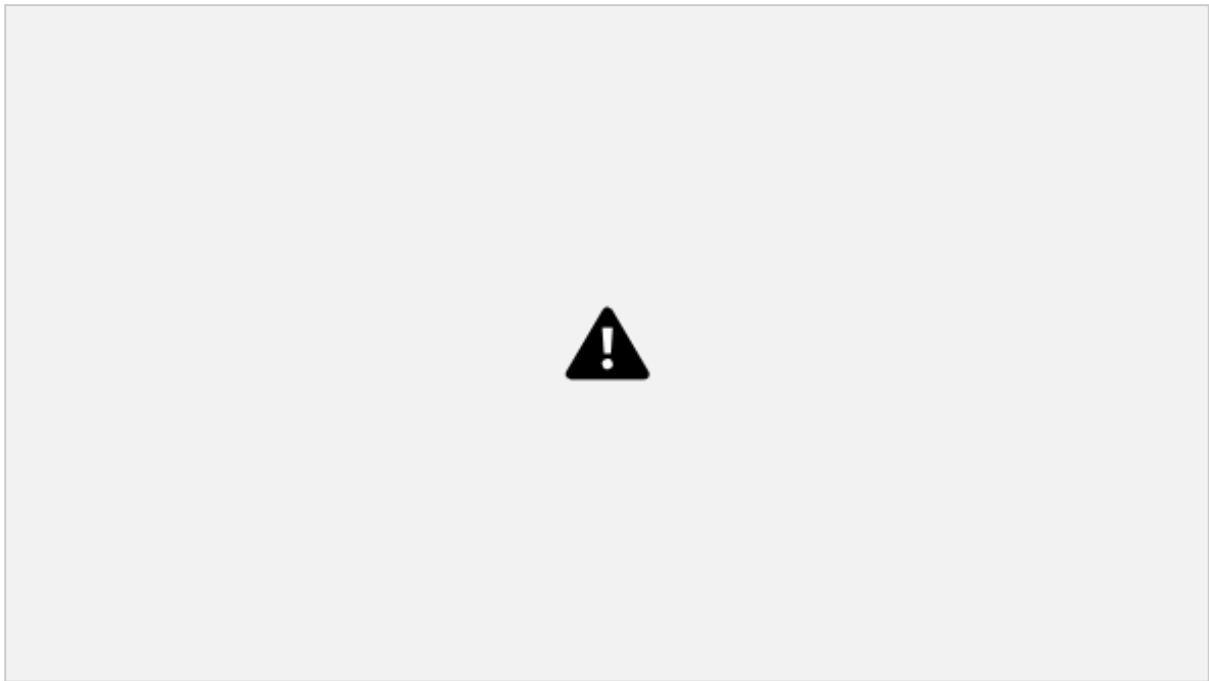
## 2) Create a collection called studentmarks



The screenshot shows the Atom editor interface with the same project. The main editor pane now displays the code for Ex2.02.js, which creates a MongoDB collection. The code logic is: it requires the 'mongodb' module, sets a connection URL to 'mongodb://localhost:27017/', connects to the database, creates a collection named 'studentmarks', logs a success message, and closes the connection.

```
1 var MongoClient = require('mongodb').MongoClient;
2
3 var url = "mongodb://localhost:27017/";
4
5 MongoClient.connect(url, function(err, db) {
6
7   if (err) throw err;
8
9   var dbo = db.db("student");
10
11   dbo.createCollection("studentmarks", function(err, res) {
12
13     if (err) throw err;
14
15     console.log("Collection created!");
16
17     db.close();
18
19   });
20
21 });
```

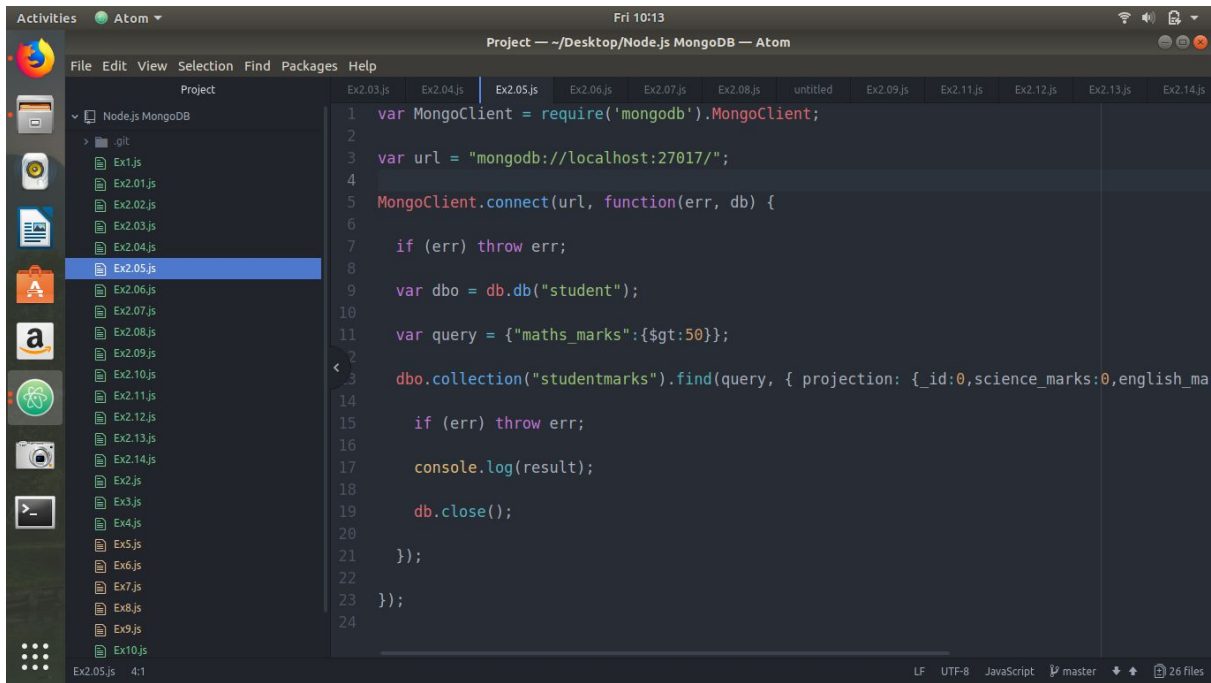
## 3) Create the documents listed in above table.



4) Increase the maths marks of Mala by 6 marks

```
1 var MongoClient = require('mongodb').MongoClient;
2
3 var url = 'mongodb://127.0.0.1:27017/';
4
5 MongoClient.connect(url, function(err, db) {
6
7   if (err) throw err;
8
9   var dbo = db.db('student');
10
11   var myquery = { name: 'Mala' };
12
13   var newvalues = { $inc: { maths_marks: 6 } };
14
15   dbo.collection('studentmarks').updateOne(myquery, newvalues, function(err, res) {
16
17     if (err) throw err;
18
19     console.log('1 document updated');
20
21     db.close();
22
23   });
24
25 });
```

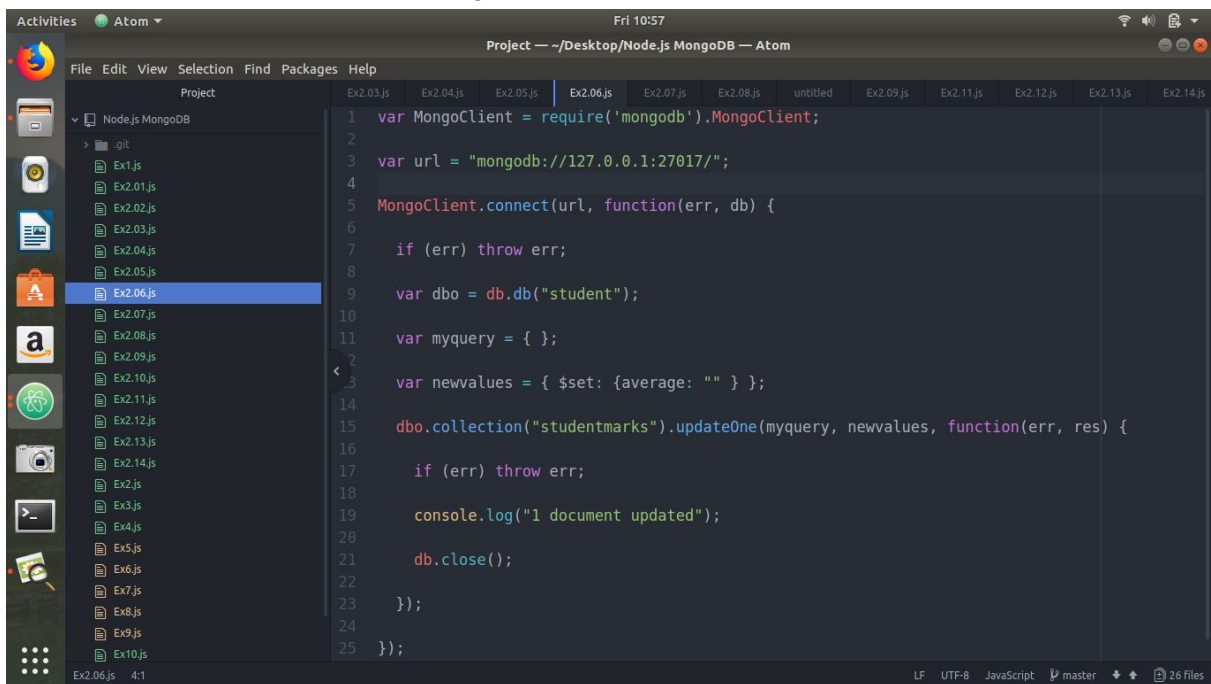
5) List the names of students who got more than 50 marks in Maths Subject.

A screenshot of the Atom text editor interface. The title bar shows 'Fri 10:13' and 'Project - ~/Desktop/Node.js MongoDB - Atom'. The menu bar includes File, Edit, View, Selection, Find, Packages, and Help. The left sidebar shows a file explorer for 'Node.js MongoDB' with a list of files from Ex1.js to Ex10.js. Ex2.05.js is selected and highlighted in blue. The main editor area displays the code for Ex2.05.js, which connects to a MongoDB instance at localhost:27017, creates a 'student' database, and inserts a document into the 'studentmarks' collection with 'maths\_marks' set to 50. The code is as follows:

```
1 var MongoClient = require('mongodb').MongoClient;
2
3 var url = "mongodb://localhost:27017/";
4
5 MongoClient.connect(url, function(err, db) {
6
7     if (err) throw err;
8
9     var dbo = db.db("student");
10
11     var query = {"maths_marks":{"$gt:50}};
12
13     dbo.collection("studentmarks").find(query, { projection: {_id:0,science_marks:0,english_ma
14
15     if (err) throw err;
16
17     console.log(result);
18
19     db.close();
20
21 });
22
23 });
24
```

The status bar at the bottom indicates 'LF UTF-8 JavaScript master' and '26 files'.

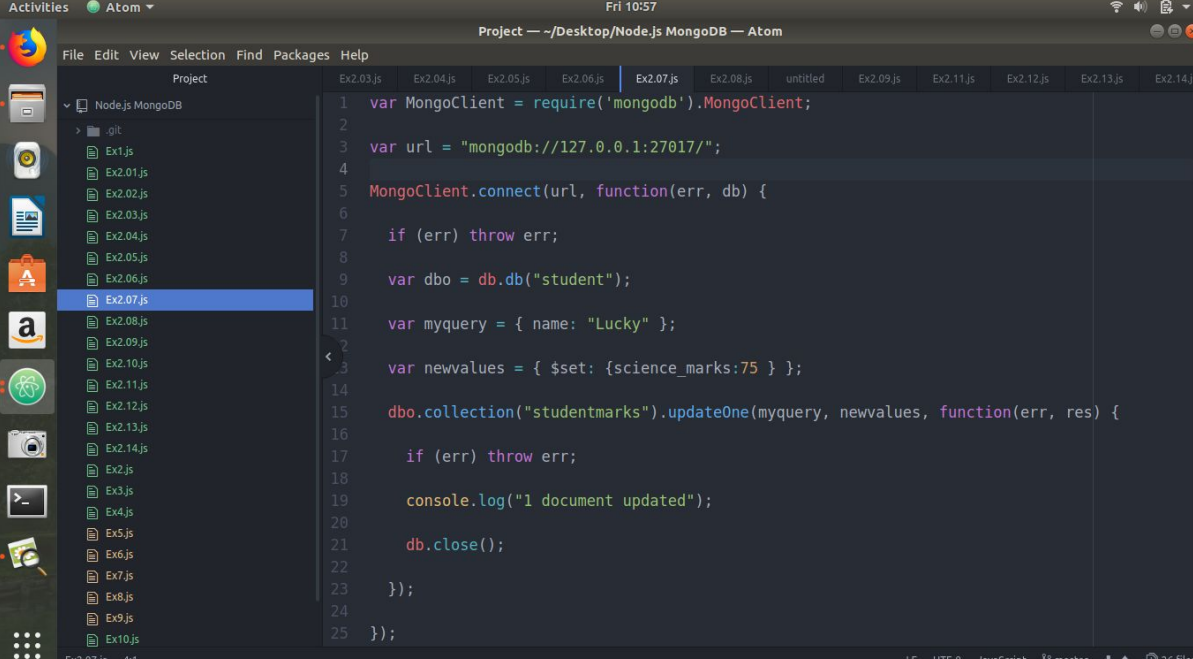
6)Add a new column(field) for Average for all students.

A screenshot of the Atom text editor interface. The title bar shows 'Fri 10:57' and 'Project - ~/Desktop/Node.js MongoDB - Atom'. The menu bar includes File, Edit, View, Selection, Find, Packages, and Help. The left sidebar shows a file explorer for 'Node.js MongoDB' with a list of files from Ex1.js to Ex10.js. Ex2.06.js is selected and highlighted in blue. The main editor area displays the code for Ex2.06.js, which connects to a MongoDB instance at 127.0.0.1:27017, creates a 'student' database, and updates the 'average' field for all documents in the 'studentmarks' collection. The code is as follows:

```
1 var MongoClient = require('mongodb').MongoClient;
2
3 var url = "mongodb://127.0.0.1:27017/";
4
5 MongoClient.connect(url, function(err, db) {
6
7     if (err) throw err;
8
9     var dbo = db.db("student");
10
11     var myquery = { };
12
13     var newvalues = { $set: {average: " " } };
14
15     dbo.collection("studentmarks").updateOne(myquery, newvalues, function(err, res) {
16
17     if (err) throw err;
18
19     console.log("1 document updated");
20
21     db.close();
22
23 });
24
25 });
26
```

The status bar at the bottom indicates 'LF UTF-8 JavaScript master' and '26 files'.

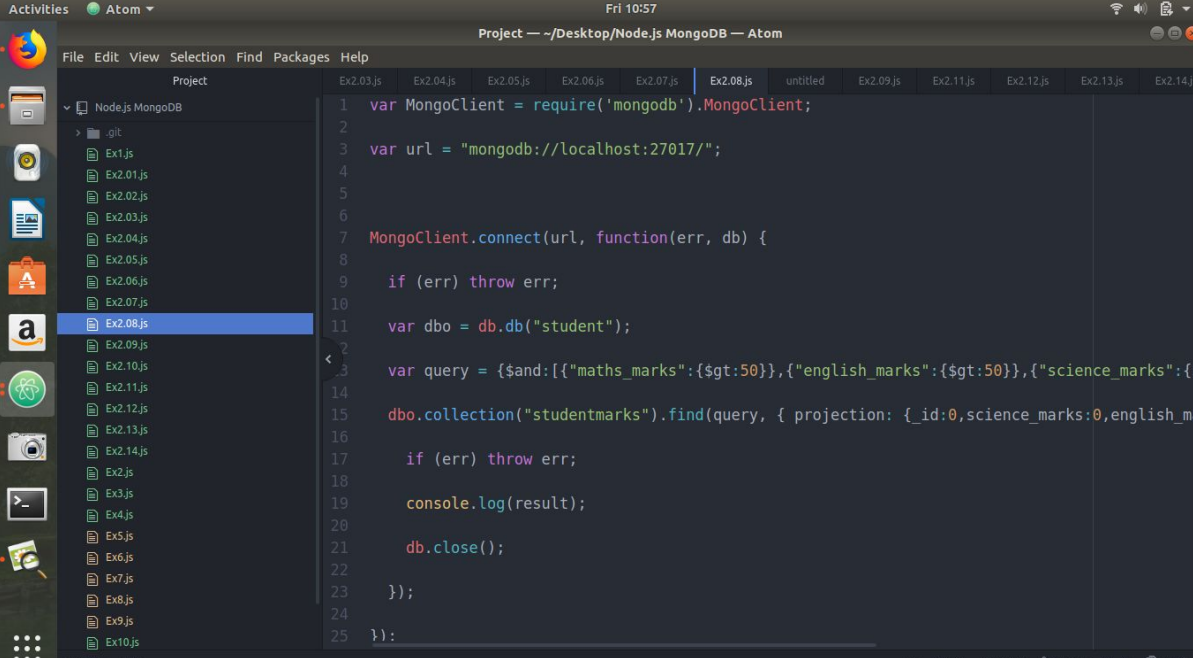
7) Update Marks\_Science=75 to Lucky .



The screenshot shows the Atom text editor with a project named "Node.js MongoDB". The file explorer on the left lists files from Ex1.js to Ex10.js, with Ex2.07.js selected. The main editor displays the code in Ex2.07.js, which connects to a MongoDB instance at 127.0.0.1:27017, updates a document in the 'studentmarks' collection where the name is 'Lucky' by setting 'science\_marks' to 75, and logs the update.

```
1 var MongoClient = require('mongodb').MongoClient;
2
3 var url = "mongodb://127.0.0.1:27017/";
4
5 MongoClient.connect(url, function(err, db) {
6
7     if (err) throw err;
8
9     var dbo = db.db("student");
10
11     var myquery = { name: "Lucky" };
12
13     var newvalues = { $set: { science_marks: 75 } };
14
15     dbo.collection("studentmarks").updateOne(myquery, newvalues, function(err, res) {
16
17         if (err) throw err;
18
19         console.log("1 document updated");
20
21         db.close();
22     });
23
24 });
25
```

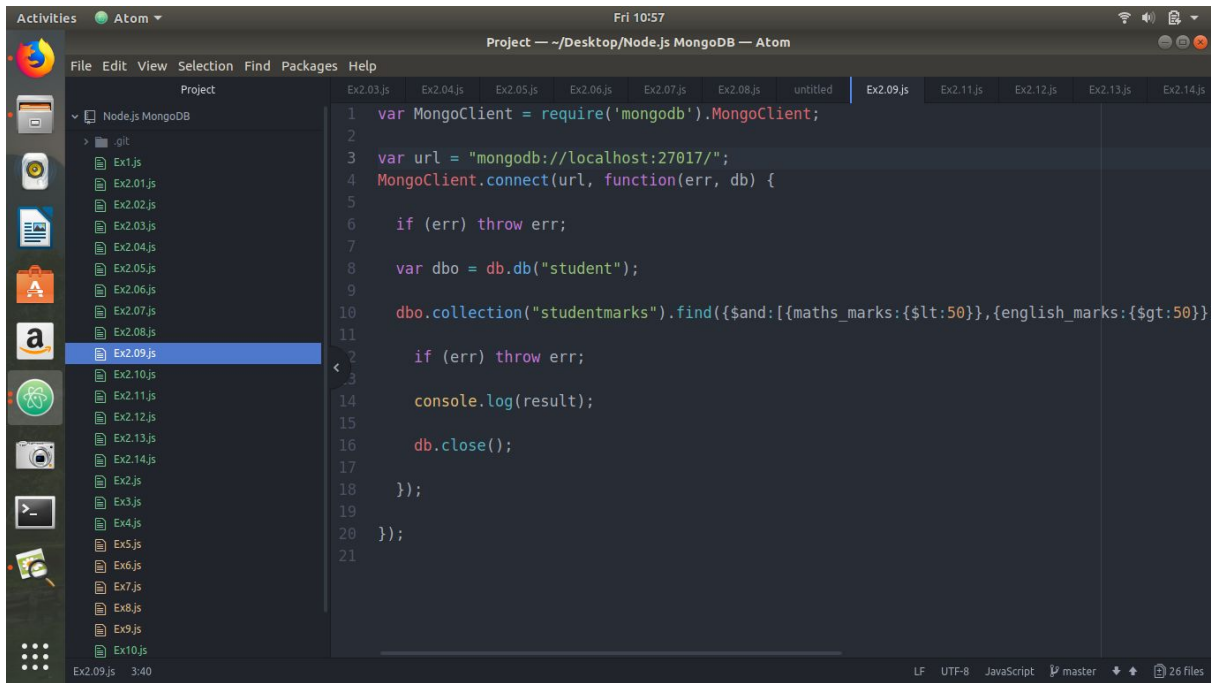
8) List the names who got more than 50 marks in all subjects.



The screenshot shows the Atom text editor with the same project. The file explorer now has Ex2.08.js selected. The main editor displays the code in Ex2.08.js, which connects to a MongoDB instance at localhost:27017, finds documents in the 'studentmarks' collection where 'maths\_marks', 'english\_marks', and 'science\_marks' are all greater than 50, and logs the results.

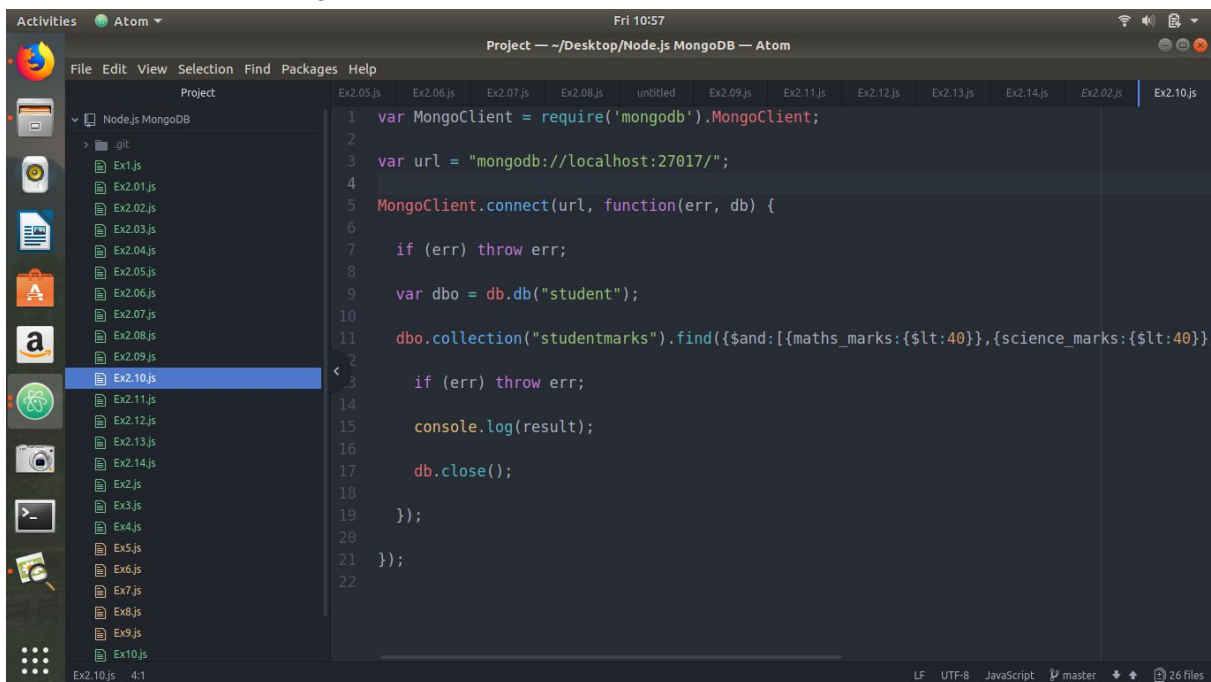
```
1 var MongoClient = require('mongodb').MongoClient;
2
3 var url = "mongodb://localhost:27017/";
4
5 MongoClient.connect(url, function(err, db) {
6
7     if (err) throw err;
8
9     var dbo = db.db("student");
10
11     var query = { $and: [{ "maths_marks": { $gt: 50 } }, { "english_marks": { $gt: 50 } }, { "science_marks": { $gt: 50 } } ] };
12
13     dbo.collection("studentmarks").find(query, { projection: { _id: 0, science_marks: 0, english_marks: 0 } }, function(err, result) {
14
15         if (err) throw err;
16
17         console.log(result);
18
19         db.close();
20     });
21
22 });
23
24
25
```

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
3 var url = "mongodb://localhost:27017/";
4 MongoClient.connect(url, function(err, db) {
5
6   if (err) throw err;
7
8   var dbo = db.db("student");
9
10  dbo.collection("studentmarks").find({$and:[{maths_marks:{<:50}},{english_marks:{>:50}}]
11
12    if (err) throw err;
13
14    console.log(result);
15
16    db.close();
17  });
18
19 });
20
21
```

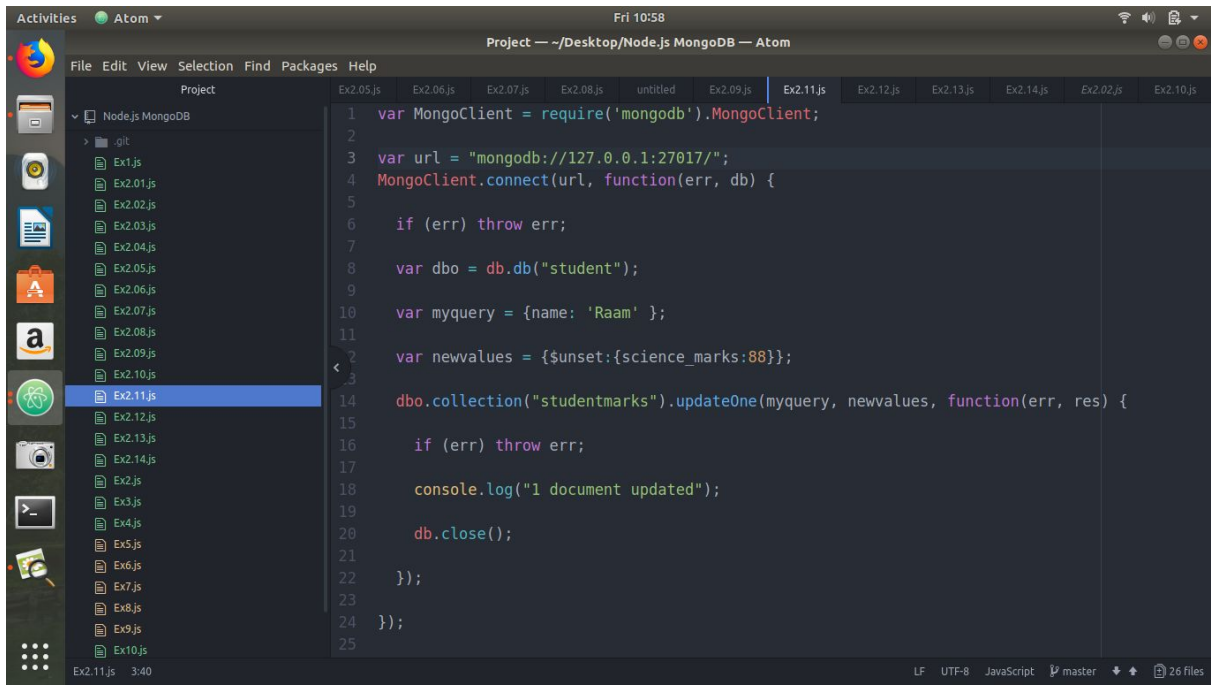
10) List the names who got less than 40 in both Maths and Science.



```
1 var MongoClient = require('mongodb').MongoClient;
2
3 var url = "mongodb://localhost:27017/";
4 MongoClient.connect(url, function(err, db) {
5
6   if (err) throw err;
7
8   var dbo = db.db("student");
9
10  dbo.collection("studentmarks").find({$and:[{maths_marks:{<:40}},{science_marks:{<:40}}]
11
12    if (err) throw err;
13
14    console.log(result);
15
16    db.close();
17  });
18
19 });
20
21
22
```

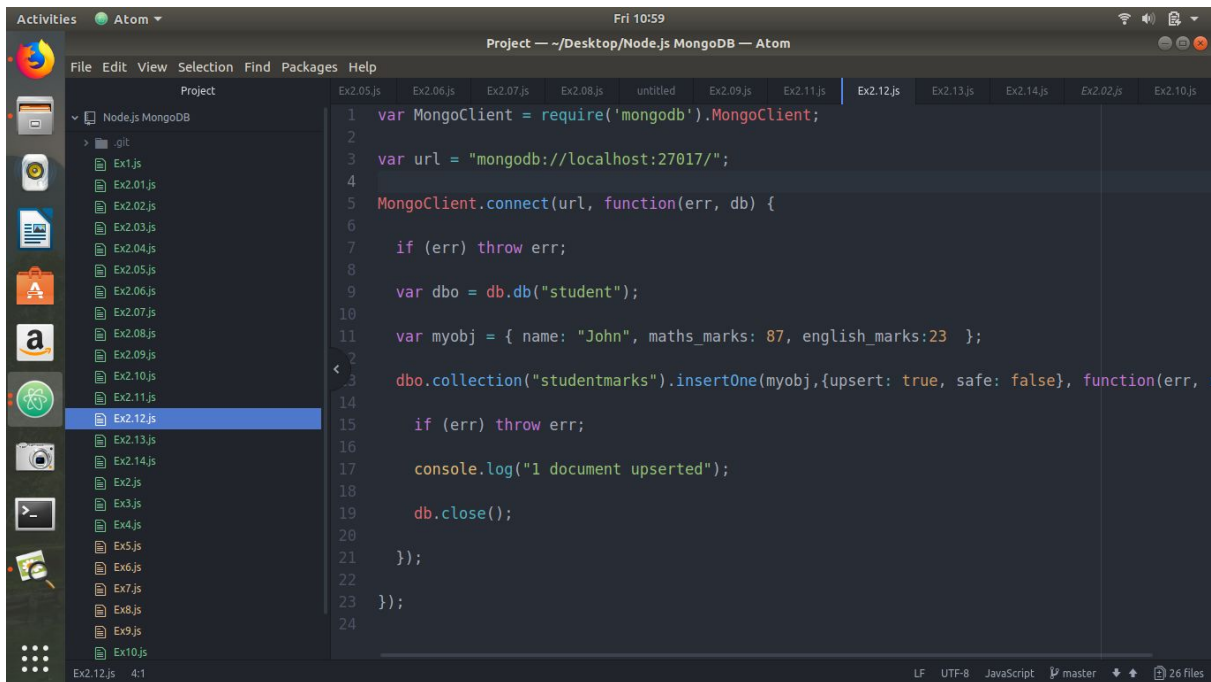
11) Remove Science column/field for Raam



A screenshot of the Atom text editor interface. The top bar shows 'Fri 10:58' and the project path 'Project - ~/Desktop/Node.js MongoDB - Atom'. The left sidebar displays a file explorer for the 'Node.js MongoDB' project, listing files from Ex1.js to Ex10.js, with Ex2.11.js selected. The main editor area shows the code for Ex2.11.js, which connects to a MongoDB instance at 'mongodb://127.0.0.1:27017/' and updates a document in the 'studentmarks' collection. The code includes error handling and a console log.

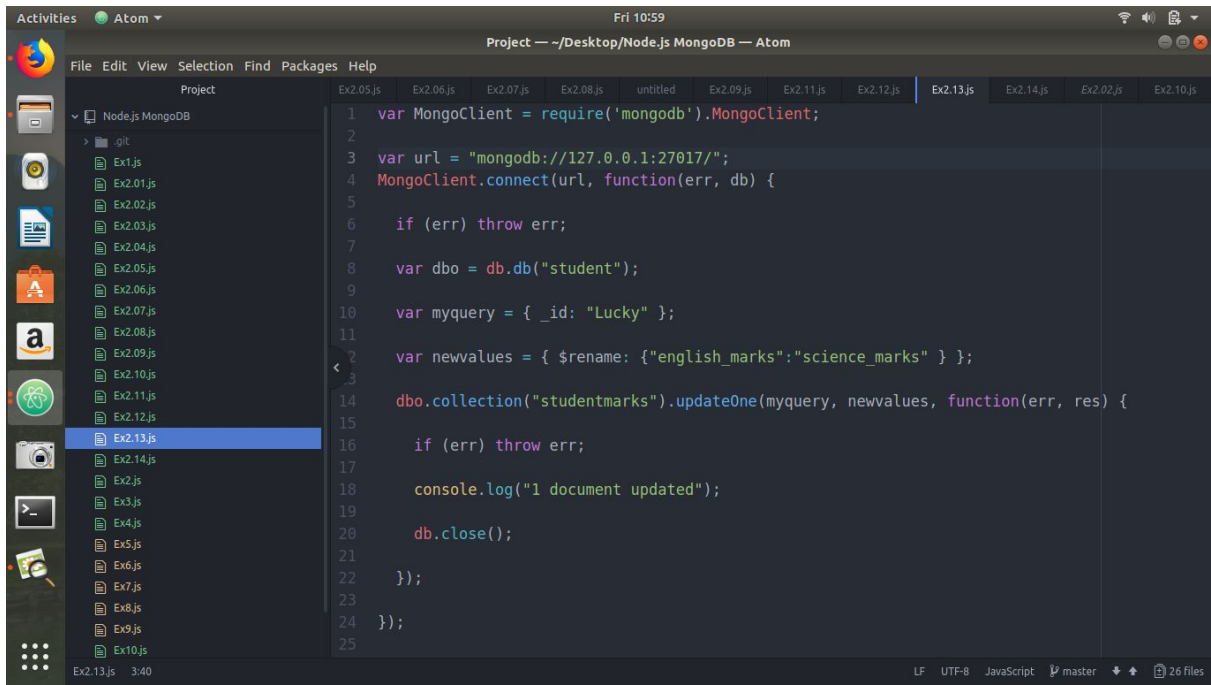
```
1 var MongoClient = require('mongodb').MongoClient;
2
3 var url = "mongodb://127.0.0.1:27017/";
4 MongoClient.connect(url, function(err, db) {
5
6     if (err) throw err;
7
8     var dbo = db.db("student");
9
10    var myquery = {name: 'Raam' };
11
12    var newvalues = {$unset:{science_marks:88}};
13
14    dbo.collection("studentmarks").updateOne(myquery, newvalues, function(err, res) {
15
16        if (err) throw err;
17
18        console.log("1 document updated");
19
20        db.close();
21
22    });
23
24 });
25
```

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

A screenshot of the Atom text editor interface. The top bar shows 'Fri 10:59' and the project path 'Project - ~/Desktop/Node.js MongoDB - Atom'. The left sidebar shows the file explorer with Ex2.12.js selected. The main editor area shows the code for Ex2.12.js, which connects to a MongoDB instance at 'mongodb://localhost:27017/' and inserts a new document into the 'studentmarks' collection. The code includes error handling and a console log.

```
1 var MongoClient = require('mongodb').MongoClient;
2
3 var url = "mongodb://localhost:27017/";
4
5 MongoClient.connect(url, function(err, db) {
6
7     if (err) throw err;
8
9     var dbo = db.db("student");
10
11    var myobj = { name: "John", maths_marks: 87, english_marks: 23 };
12
13    dbo.collection("studentmarks").insertOne(myobj,{upsert: true, safe: false}, function(err,
14
15        if (err) throw err;
16
17        console.log("1 document upserted");
18
19        db.close();
20
21    });
22
23 });
24
```

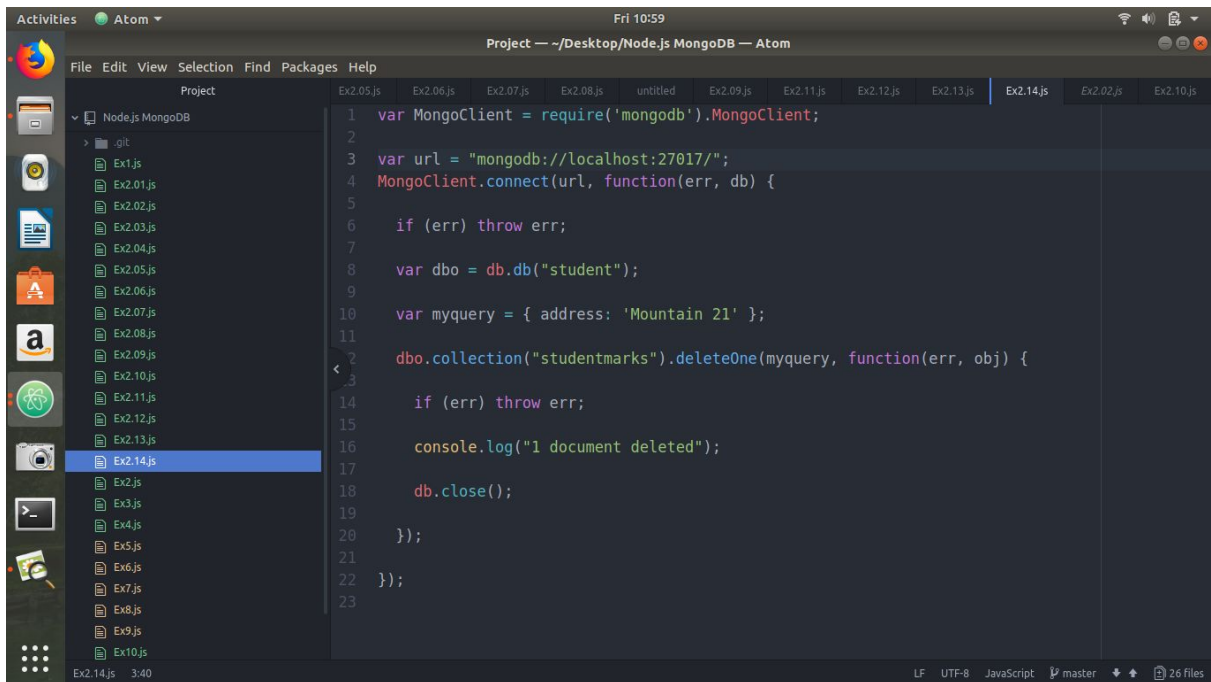
13) Rename the english\_marks column/field for John to science\_marks



The screenshot shows the Atom editor interface with a project named "Node.js MongoDB". The file explorer on the left lists files from Ex1.js to Ex10.js. The main editor window displays the code for Ex2.13.js, which connects to a MongoDB instance at 127.0.0.1:27017 and updates a document in the 'studentmarks' collection. The code is as follows:

```
1 var MongoClient = require('mongodb').MongoClient;
2
3 var url = "mongodb://127.0.0.1:27017/";
4 MongoClient.connect(url, function(err, db) {
5
6     if (err) throw err;
7
8     var dbo = db.db("student");
9
10    var myquery = { _id: "Lucky" };
11
12    var newvalues = { $rename: { "english_marks": "science_marks" } };
13
14    dbo.collection("studentmarks").updateOne(myquery, newvalues, function(err, res) {
15
16        if (err) throw err;
17
18        console.log("1 document updated");
19
20        db.close();
21
22    });
23
24 });
25
```

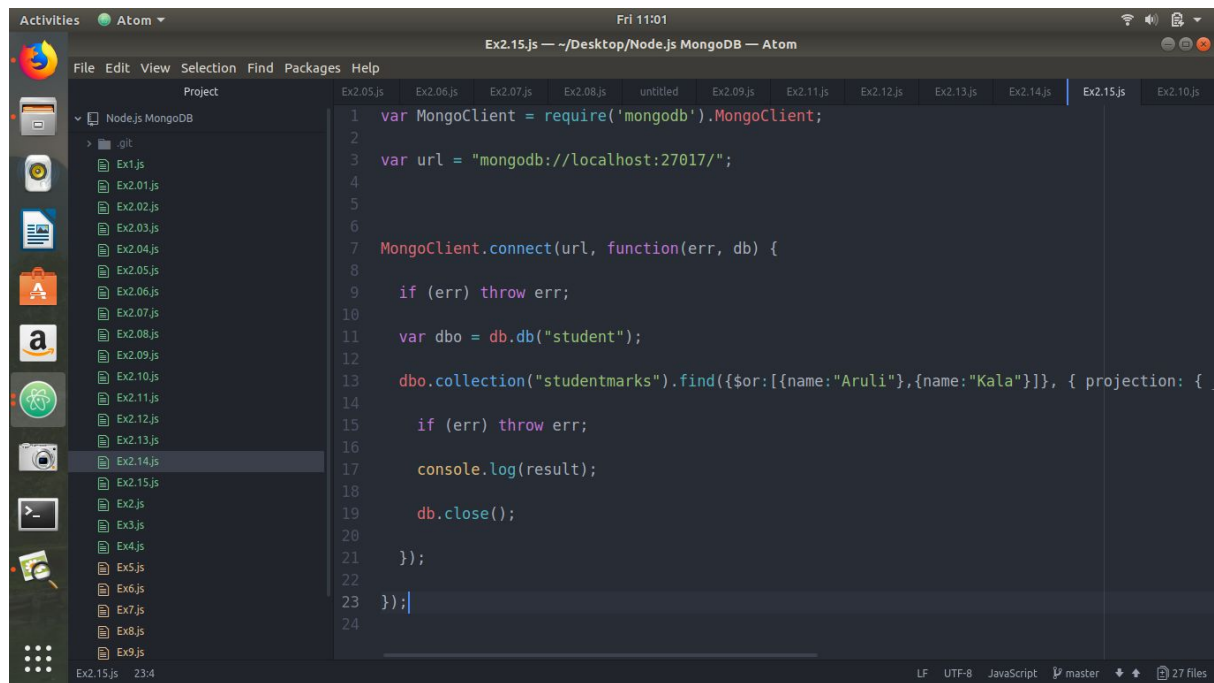
#### 14) Remove Kumaran's document from collection



The screenshot shows the Atom editor interface with the same project. The file explorer now lists files up to Ex10.js, with Ex2.14.js selected. The main editor window displays the code for Ex2.14.js, which connects to a MongoDB instance at localhost:27017 and deletes a document from the 'studentmarks' collection. The code is as follows:

```
1 var MongoClient = require('mongodb').MongoClient;
2
3 var url = "mongodb://localhost:27017/";
4 MongoClient.connect(url, function(err, db) {
5
6     if (err) throw err;
7
8     var dbo = db.db("student");
9
10    var myquery = { address: 'Mountain 21' };
11
12    dbo.collection("studentmarks").deleteOne(myquery, function(err, obj) {
13
14        if (err) throw err;
15
16        console.log("1 document deleted");
17
18        db.close();
19
20    });
21
22 });
23
```

## 15) Find Kala's or Aruli's math\_marks and science\_marks



```
1 var MongoClient = require('mongodb').MongoClient;
2
3 var url = "mongodb://localhost:27017/";
4
5
6
7 MongoClient.connect(url, function(err, db) {
8
9   if (err) throw err;
10
11   var dbo = db.db("student");
12
13   dbo.collection("studentmarks").find({$or:[{name:"Aruli"},{name:"Kala"}]}).project({
14
15     if (err) throw err;
16
17     console.log(result);
18
19     db.close();
20
21   });
22
23 });
```