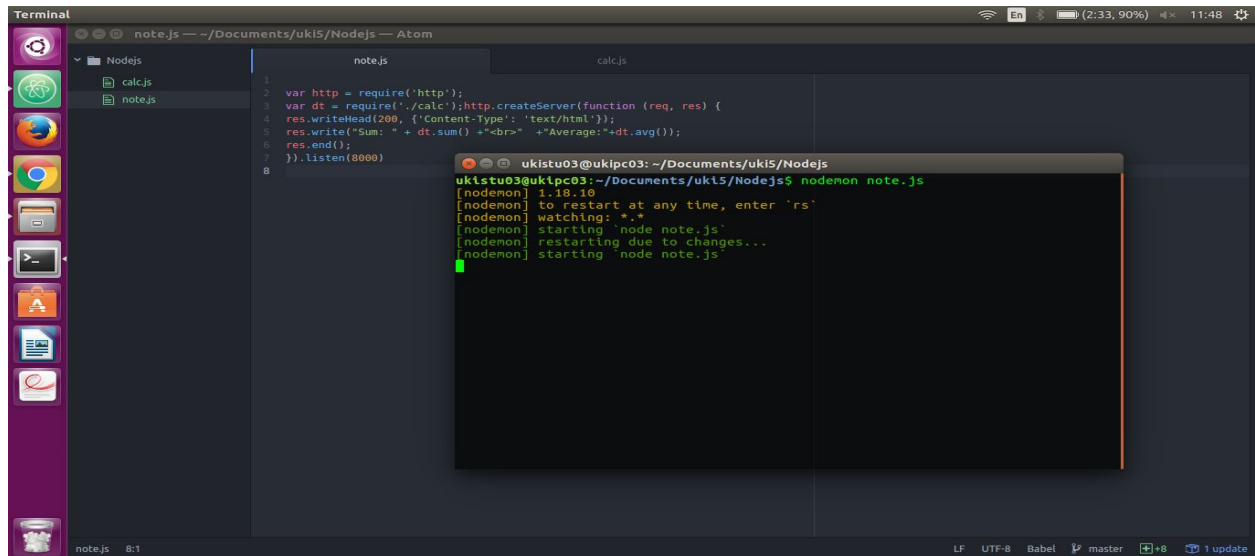


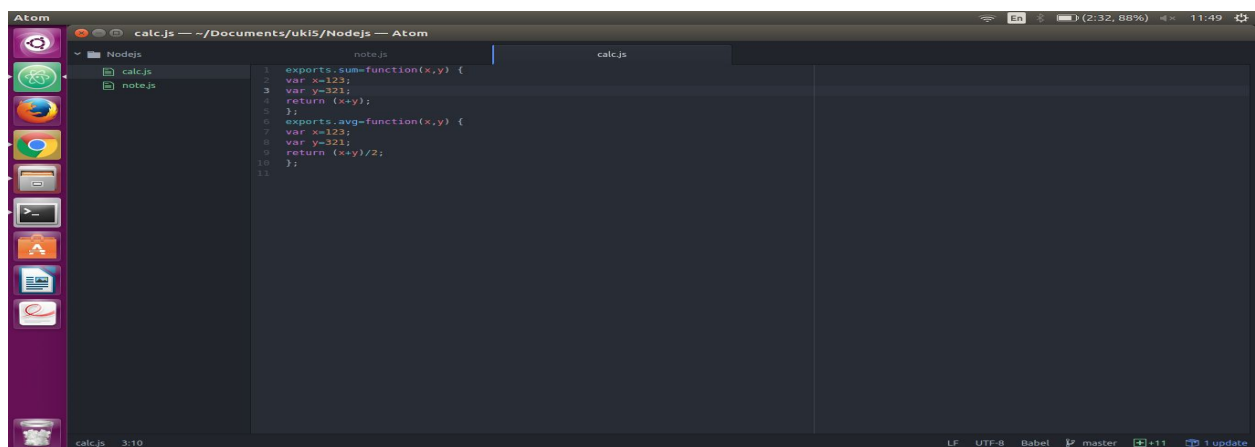
1. Create a custom module which returns the sum and average of any two numbers passed into it. Require the module and run the server by passing 123 and 321 so that the server prints out the sum and average.



The terminal window shows the Atom editor with two files: `note.js` and `calc.js`. The `note.js` file contains the following code:

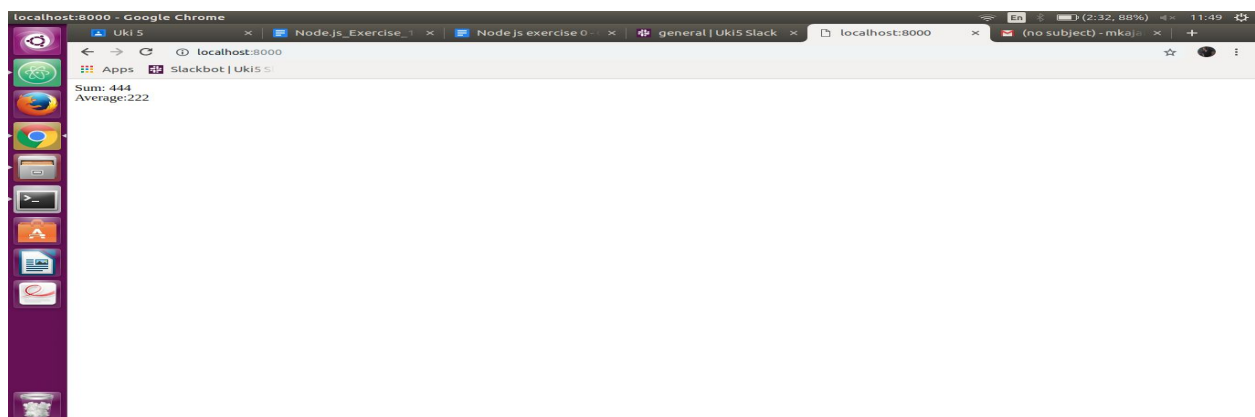
```
1 var http = require('http');
2 var fs = require('fs');
3 var dt = require('./calc');
4 http.createServer(function (req, res) {
5   res.writeHead(200, {'Content-Type': 'text/html'});
6   res.write("Sum: " + dt.sum() + "<br>" + "Average: " + dt.avg());
7   res.end();
8 }).listen(8000)
```

A terminal window is open, showing the command `nodemon note.js` being executed. The output shows the server starting and listening on port 8000.

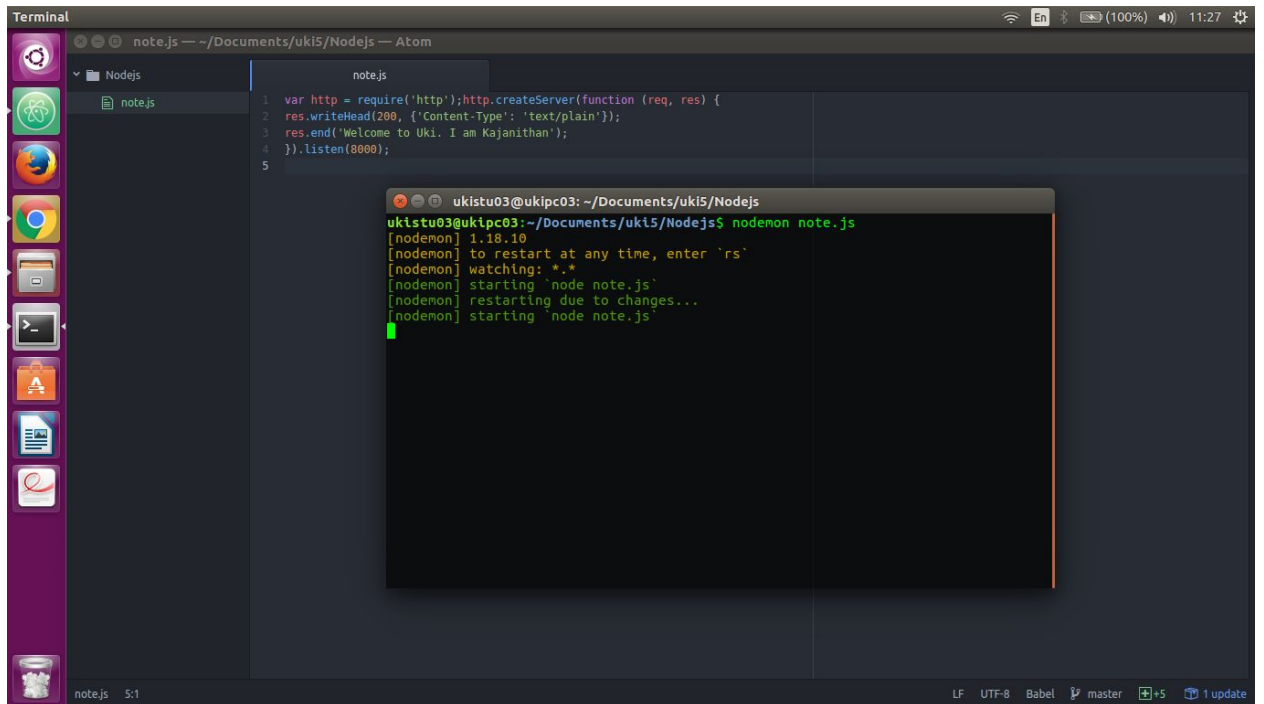


The Atom editor shows the `calc.js` file with the following code:

```
1 exports.sum=function(x,y) {
2   var x=123;
3   var y=321;
4   return (x+y);
5 };
6 exports.avg=function(x,y) {
7   var x=123;
8   var y=321;
9   return (x+y)/2;
10 };
11
```



2. Create a simple http server and print “Welcome to Uki. I am **yourname**” when a request is sent to your server via the port 8000. (Note - Change different port numbers and check)

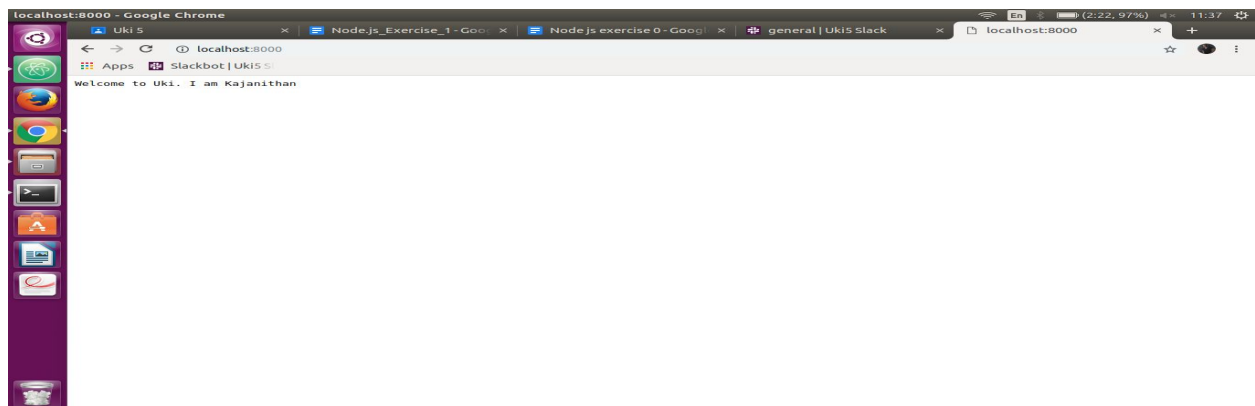


The screenshot shows an Atom editor window with a file named `note.js` open. The code in the editor is as follows:

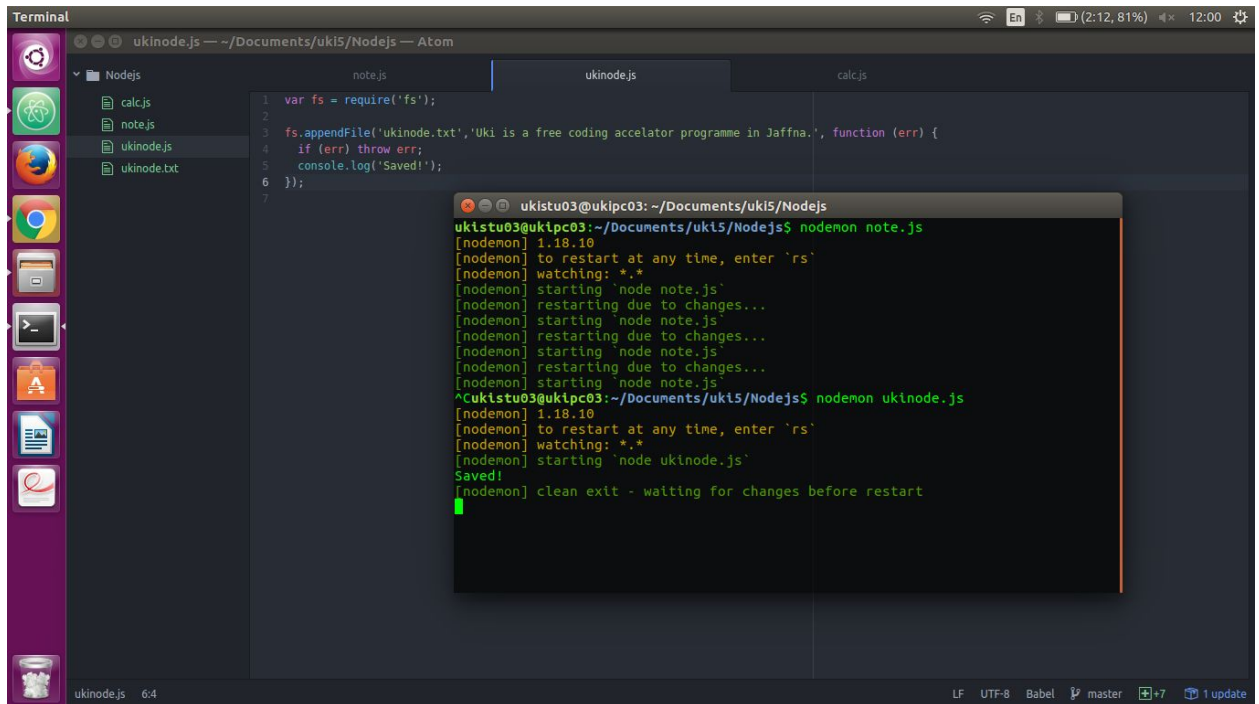
```
1 var http = require('http'); http.createServer(function (req, res) {
2   res.writeHead(200, {'Content-Type': 'text/plain'});
3   res.end('Welcome to Uki. I am Kajanithan');
4 }).listen(8000);
5
```

Below the editor, a terminal window is open, showing the command `nodemon note.js` being executed. The terminal output is:

```
ukistu03@ukipc03: ~/Documents/uki5/Nodejs
[nodemon] 1.18.10
[nodemon] to restart at any time, enter `rs`
[nodemon] watching: *.*
[nodemon] starting 'node note.js'
[nodemon] restarting due to changes...
[nodemon] starting 'node note.js'
```



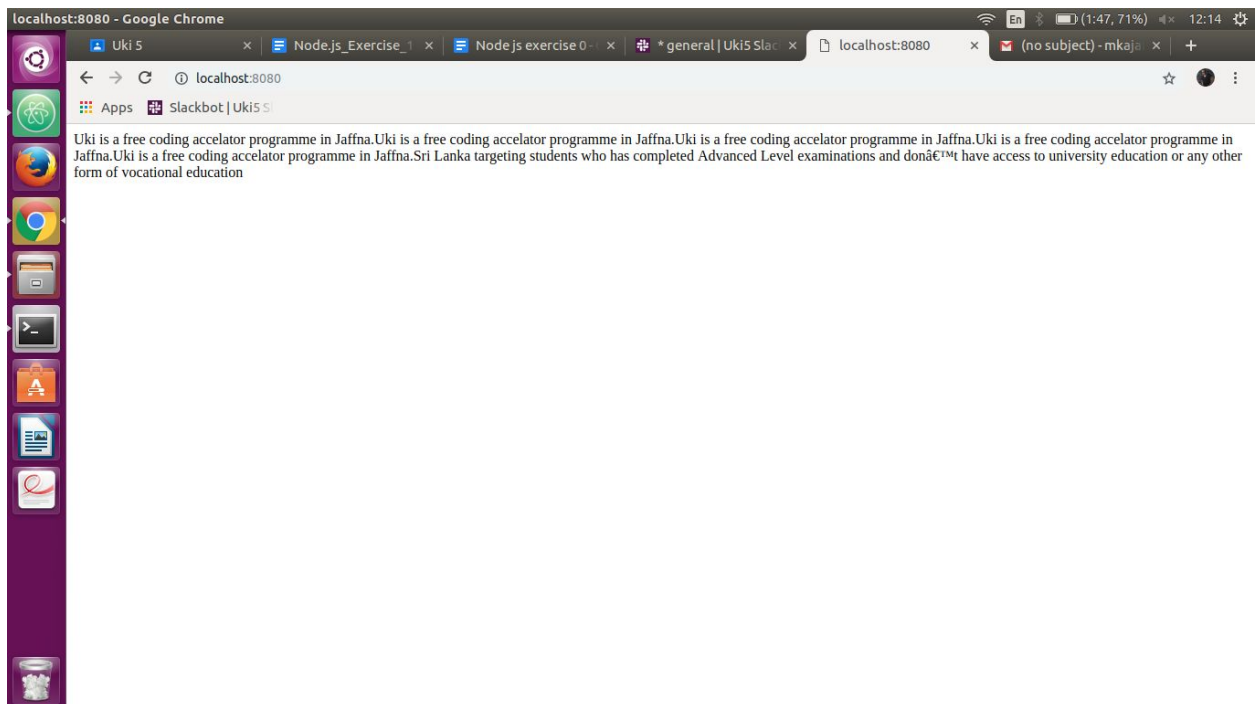
3. Using the file system module create a new file called ukinode.txt
- 3.1 Write a paragraph about Uki into that file



The screenshot shows the Atom editor interface with a file explorer on the left showing a directory named 'Nodejs' containing 'calc.js', 'note.js', 'ukinode.js', and 'ukinode.txt'. The main editor area shows the code in 'ukinode.js':

```
1 var fs = require('fs');
2
3 fs.appendFile('ukinode.txt', 'Uki is a free coding accelator programme in Jaffna.', function (err) {
4   if (err) throw err;
5   console.log('Saved!');
6 });
7
```

An inset terminal window shows the command prompt where the user runs 'nodemon note.js' and 'nodemon ukinode.js'. The output shows the file being saved and the message 'Saved!' being printed to the console.



3.2 Serve that file to the client (Read File) over your server

The screenshot shows the Atom editor with a file explorer on the left containing files: calc.js, ex3.2.js, ex3.3.js, note.js, ukinode.js, and ukinode.txt. The main editor displays the code for ex3.2.js, which is a simple HTTP server that reads a file and serves it to the client. The code is as follows:

```
1 var http = require('http');
2 var fs = require('fs');
3 http.createServer(function (req, res) {
4   fs.readFile('ukinode.txt', function(err, data) {
5     res.writeHead(200, {'Content-Type': 'text/html'});
6     res.write(data);
7     res.end();
8   });
9 }).listen(8080);
```

Below the editor, a terminal window shows the command `node ex3.2.js` being executed. The terminal output indicates that the server is running on port 1.18.10 and is watching for changes.

```
ukistu03@ukipc03: ~/Documents/uki5/Nodejs
ukistu03@ukipc03:~/Documents/uki5/Nodejs$ node ex3.2.js
[nodemon] 1.18.10
[nodemon] to restart at any time, enter 'rs'
[nodemon] watching: *.*
[nodemon] starting 'node ex3.2.js'
```

3.3 Append another paragraph about Uki and now serve the new file

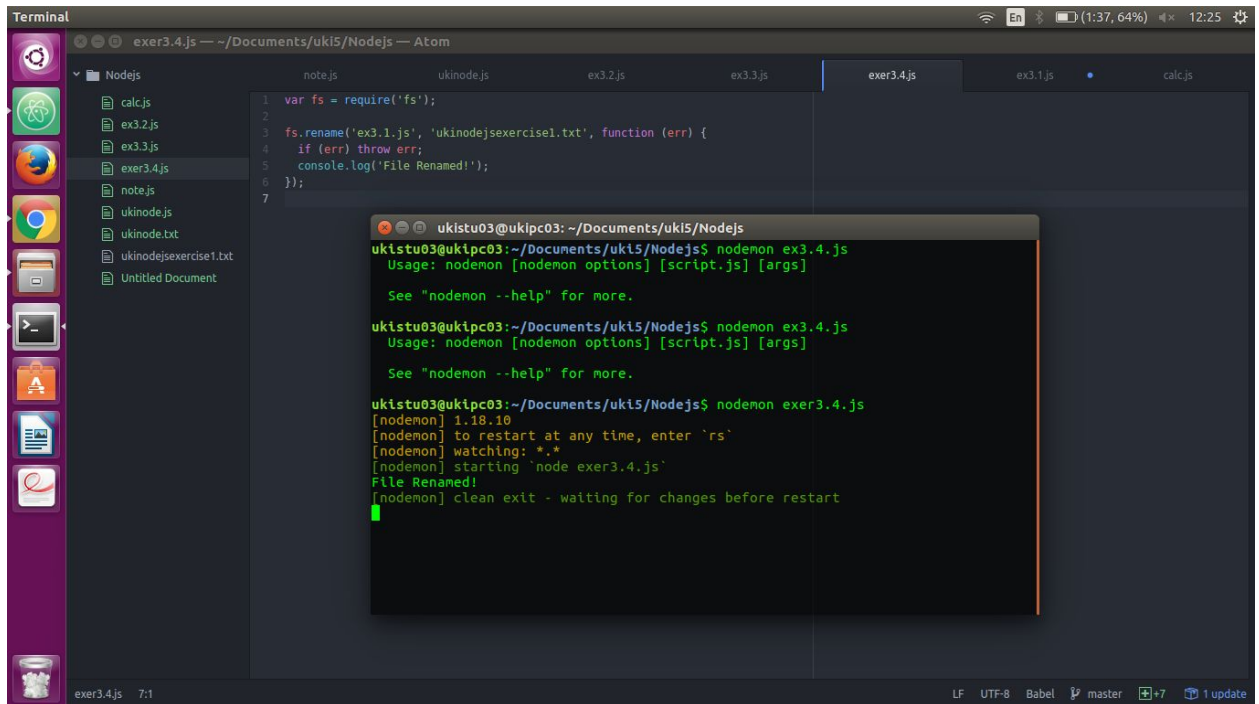
The screenshot shows the Atom editor with the same file explorer. The main editor displays the code for ex3.3.js, which is an updated version of the server that appends a new paragraph to the file. The code is as follows:

```
1 var fs = require('fs');
2
3 fs.appendFile('ukinode.txt', 'Sri Lanka targeting students who has completed Advanced Level examinations and don't have access to university educat
4 if (err) throw err;
5 console.log('Saved!');
6 });
7
```

Below the editor, a terminal window shows the command `node ex3.3.js` being executed. The terminal output indicates that the server is running on port 1.18.10 and is watching for changes. It also shows the message "Saved!" and "clean exit - waiting for changes before restart".

```
ukistu03@ukipc03: ~/Documents/uki5/Nodejs
ukistu03@ukipc03:~/Documents/uki5/Nodejs$ node ex3.3.js
[nodemon] 1.18.10
[nodemon] to restart at any time, enter 'rs'
[nodemon] watching: *.*
[nodemon] starting 'node ex3.3.js'
^Cukistu03@ukipc03:~/Documents/uki5/Nodejs$ node ex3.3.js
[nodemon] 1.18.10
[nodemon] to restart at any time, enter 'rs'
[nodemon] watching: *.*
[nodemon] starting 'node ex3.3.js'
Saved!
[nodemon] clean exit - waiting for changes before restart
```

3.4 Rename the file as ukinodejsexercise1.txt



The screenshot shows the Atom editor with a file named `exer3.4.js` open. The file contains the following JavaScript code:

```
1 var fs = require('fs');
2
3 fs.rename('ex3.1.js', 'ukinodejsexercise1.txt', function (err) {
4   if (err) throw err;
5   console.log('File Renamed!');
6 });
7
```

A terminal window is open in the foreground, showing the execution of the script using `nodemon`:

```
ukistu03@uklpc03: ~/Documents/ukl5/Nodejs
ukistu03@uklpc03:~/Documents/ukl5/Nodejs$ nodemon ex3.4.js
Usage: nodemon [nodemon options] [script.js] [args]

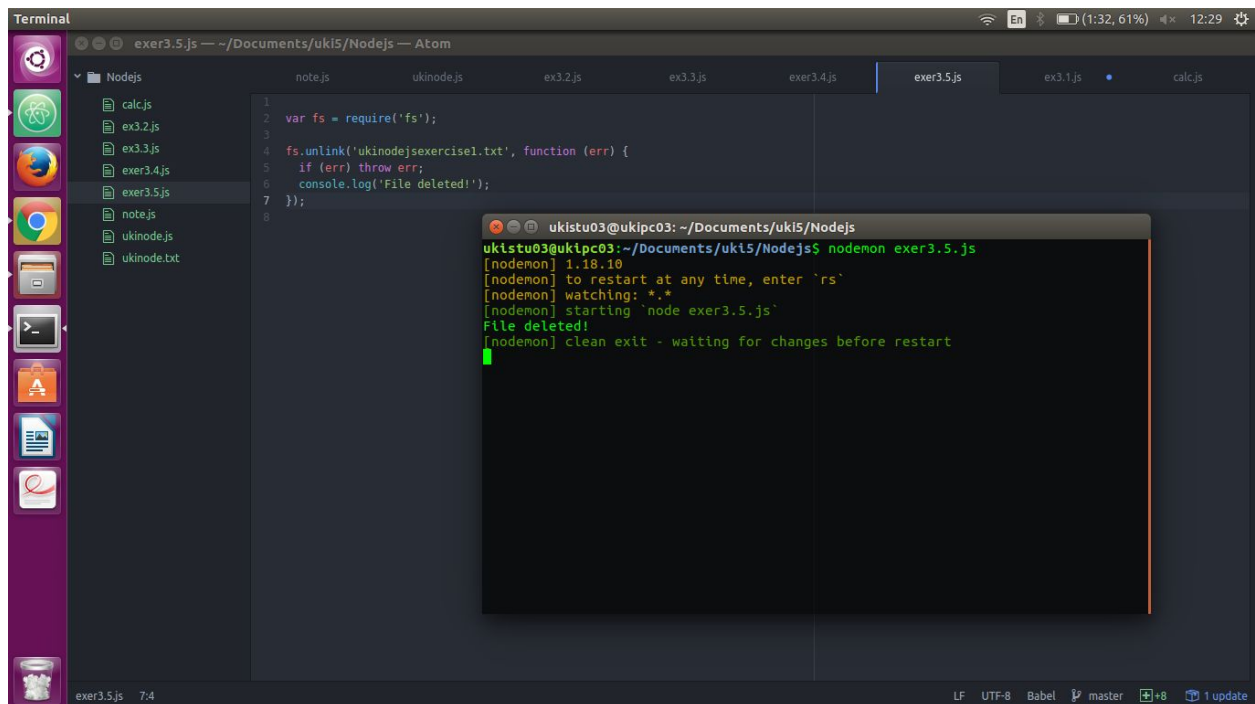
See "nodemon --help" for more.

ukistu03@uklpc03:~/Documents/ukl5/Nodejs$ nodemon ex3.4.js
Usage: nodemon [nodemon options] [script.js] [args]

See "nodemon --help" for more.

ukistu03@uklpc03:~/Documents/ukl5/Nodejs$ nodemon ex3.4.js
[nodemon] 1.18.10
[nodemon] to restart at any time, enter 'rs'
[nodemon] watching: *.*
[nodemon] starting 'node ex3.4.js'
File Renamed!
[nodemon] clean exit - waiting for changes before restart
```

3.5 Delete the file you created



The screenshot shows the Atom editor with a file named `exer3.5.js` open. The file contains the following JavaScript code:

```
1 var fs = require('fs');
2
3
4 fs.unlink('ukinodejsexercise1.txt', function (err) {
5   if (err) throw err;
6   console.log('File deleted!');
7 });
8
```

A terminal window is open in the foreground, showing the execution of the script using `nodemon`:

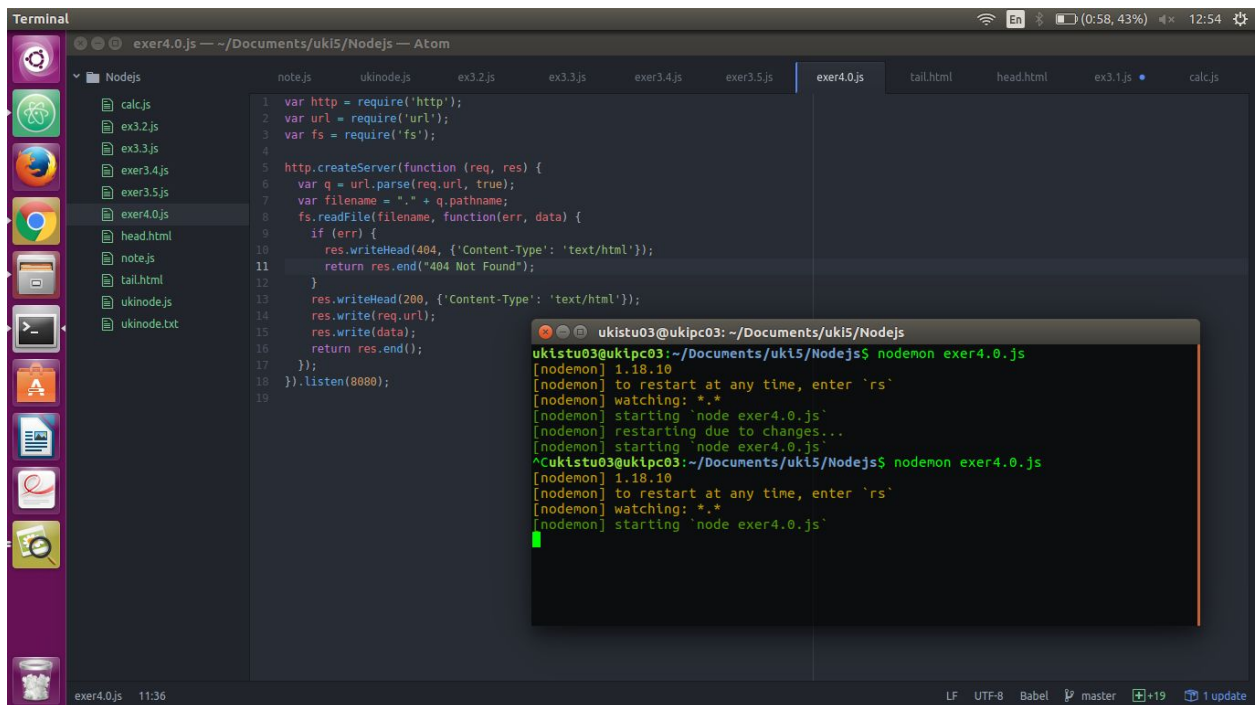
```
ukistu03@uklpc03: ~/Documents/ukl5/Nodejs
ukistu03@uklpc03:~/Documents/ukl5/Nodejs$ nodemon ex3.5.js
[nodemon] 1.18.10
[nodemon] to restart at any time, enter 'rs'
[nodemon] watching: *.*
[nodemon] starting 'node ex3.5.js'
File deleted!
[nodemon] clean exit - waiting for changes before restart
```

4. Create two html files called head.html which is a web page which says 'you have got head' and tail.html which is a web page which says 'you have got tail' and save them in the same folder as your node.js files. Create a Node.js file that opens the requested file and returns the content to the client. If anything goes wrong, throw a 404 error.

If you have followed the correct steps you should see two different results when opening these two addresses:

http://localhost:8080/head.html -> You have got head

http://localhost:8080/tail.html -> You have got tail



The screenshot shows an Atom editor window with a file explorer on the left and a code editor in the center. The file explorer shows a directory named 'Nodejs' containing files: calc.js, ex3.2.js, ex3.3.js, ex3.4.js, ex3.5.js, ex4.0.js, head.html, note.js, tail.html, ukinode.js, and ukinode.txt. The code editor shows the content of 'ex4.0.js', which is a Node.js server script. The script uses the 'http' and 'fs' modules to create a server that listens on port 8080. It checks if the requested file exists and returns its content as a text/html response. If the file is not found, it returns a 404 error. The terminal window shows the command 'nodemon ex4.0.js' being executed, and the output indicates that the server is running and watching for changes.

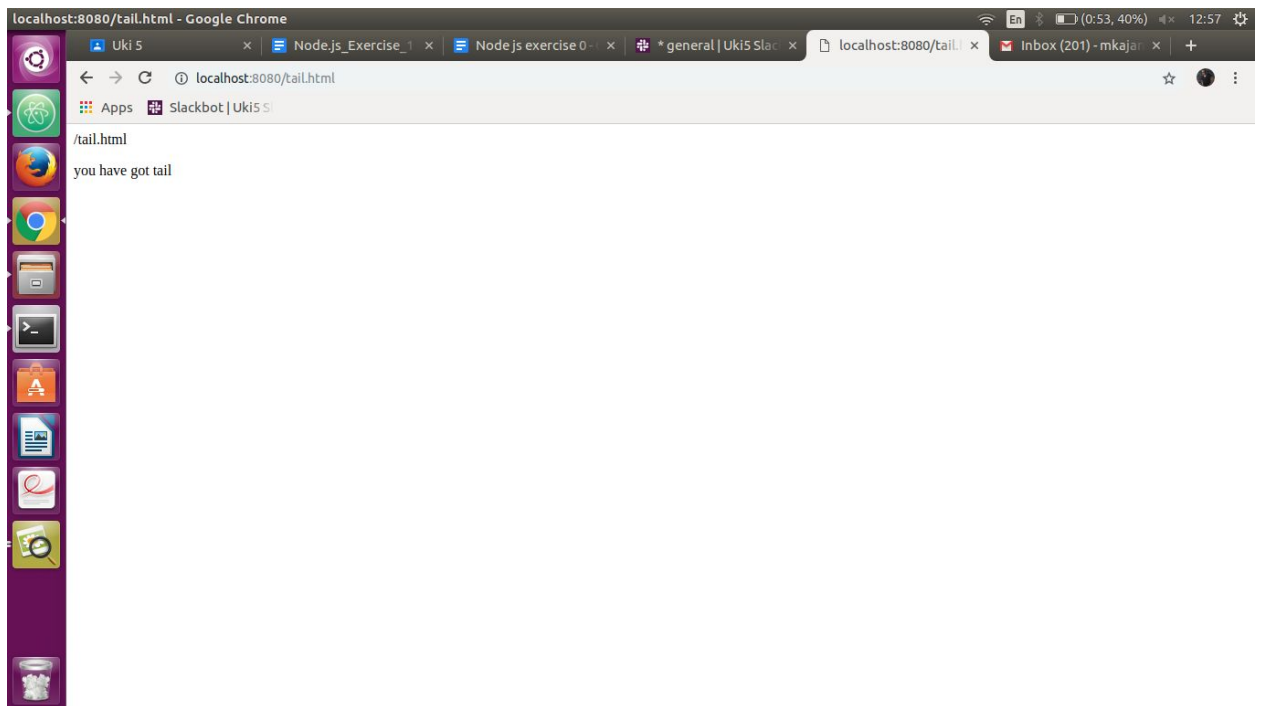
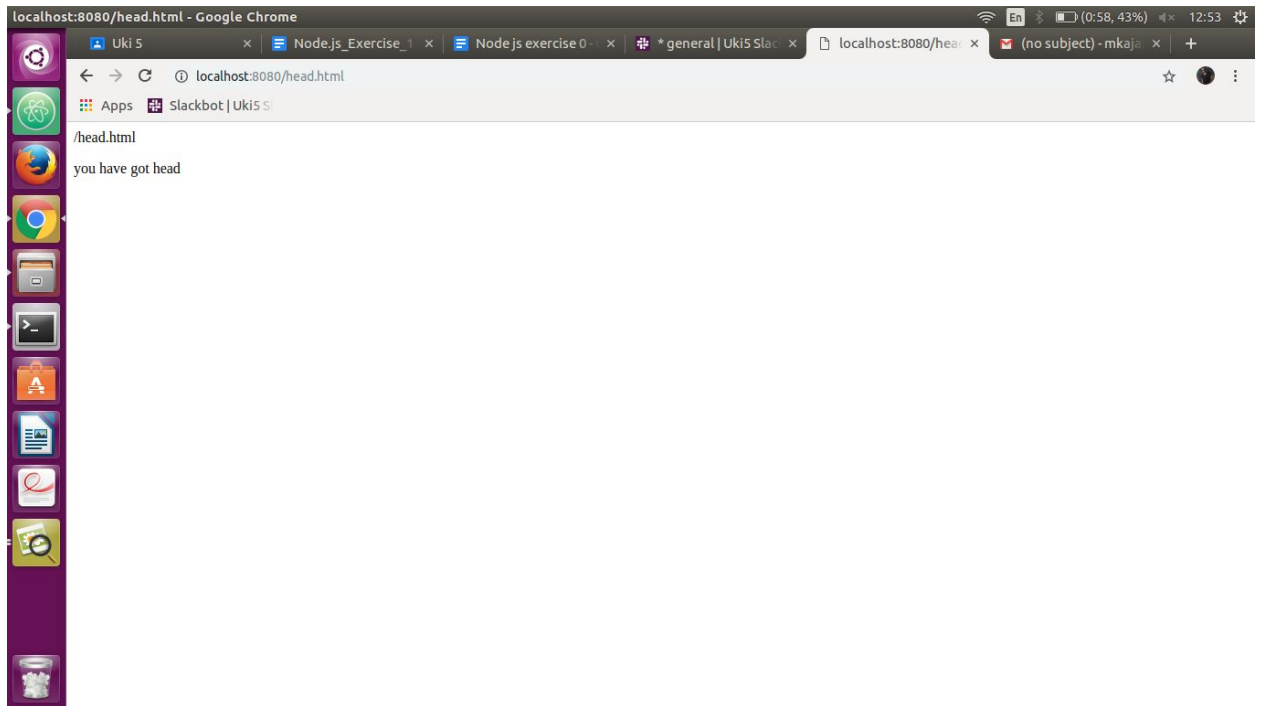
```
Terminal
ex4.0.js -- ~/Documents/ukl5/Nodejs -- Atom

Nodejs
calc.js
ex3.2.js
ex3.3.js
ex3.4.js
ex3.5.js
ex4.0.js
head.html
note.js
tail.html
ukinode.js
ukinode.txt

1 var http = require('http');
2 var url = require('url');
3 var fs = require('fs');
4
5 http.createServer(function (req, res) {
6   var q = url.parse(req.url, true);
7   var filename = "." + q.pathname;
8   fs.readFile(filename, function(err, data) {
9     if (err) {
10      res.writeHead(404, {'Content-Type': 'text/html'});
11      return res.end("404 Not Found");
12    }
13    res.writeHead(200, {'Content-Type': 'text/html'});
14    res.write(req.url);
15    res.write(data);
16    return res.end();
17  });
18 }).listen(8080);
19

ukl5tu03@ukl5pc03: ~/Documents/ukl5/Nodejs
ukl5tu03@ukl5pc03:~/Documents/ukl5/Nodejs$ nodemon ex4.0.js
[nodemon] 1.18.10
[nodemon] to restart at any time, enter `rs`
[nodemon] watching: *.*
[nodemon] starting `node ex4.0.js`
[nodemon] restarting due to changes...
[nodemon] starting `node ex4.0.js`
^Cukl5tu03@ukl5pc03:~/Documents/ukl5/Nodejs$ nodemon ex4.0.js
[nodemon] 1.18.10
[nodemon] to restart at any time, enter `rs`
[nodemon] watching: *.*
[nodemon] starting `node ex4.0.js`

ex4.0.js 11:36
LF UTF-8 Babel master +19 1 update
```



Atom

head.html — ~/Documents/uki5/Nodejs — Atom

- Nodejs
 - calc.js
 - ex3.2.js
 - ex3.3.js
 - exer3.4.js
 - exer3.5.js
 - exer4.0.js
 - head.html
 - note.js
 - tail.html
 - ukinode.js
 - ukinode.txt

```
1 <html>
2 <head>
3
4 </head>
5 <body>
6   <p>you have got head</p>
7 </body>
8 </html>
9
```

head.html 6:23

LF UTF-8 HTML master +9 1 update

Atom

tail.html — ~/Documents/uki5/Nodejs — Atom

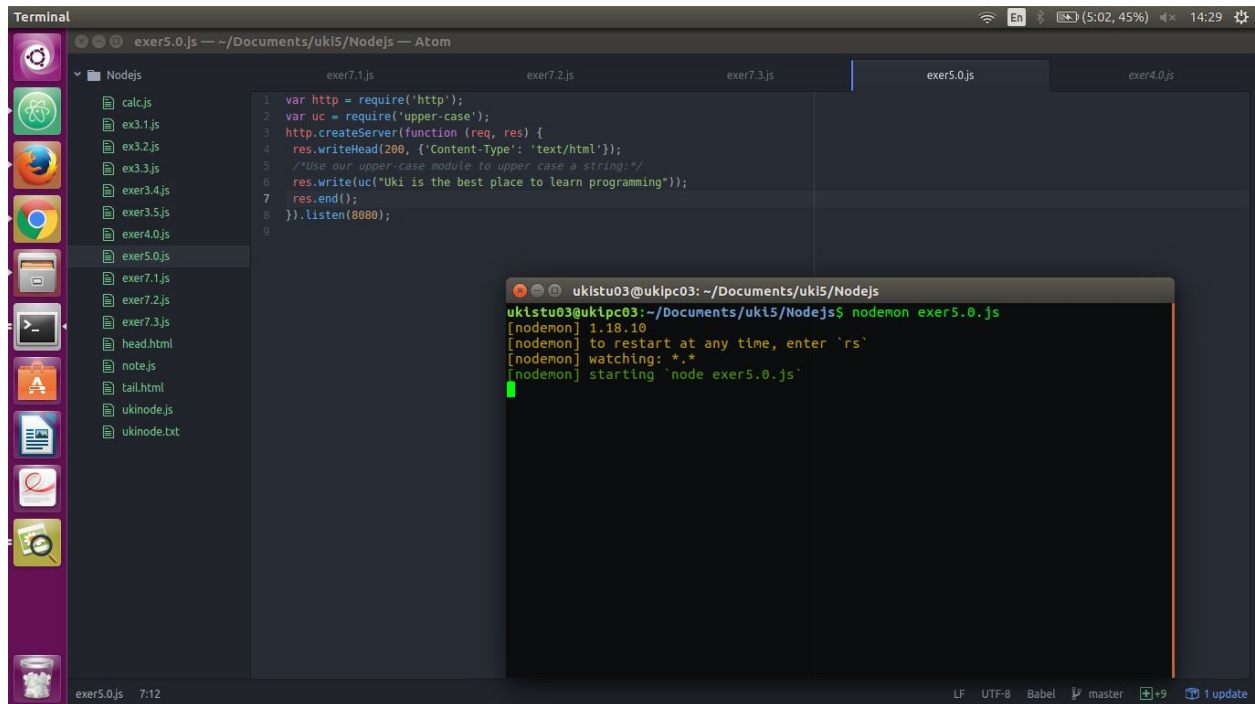
- Nodejs
 - calc.js
 - ex3.2.js
 - ex3.3.js
 - exer3.4.js
 - exer3.5.js
 - exer4.0.js
 - head.html
 - note.js
 - tail.html
 - ukinode.js
 - ukinode.txt

```
1 <html>
2 <head>
3
4 </head>
5 <body>
6   <p>you have got tail</p>
7 </body>
8 </html>
9
```

tail.html 2:7

LF UTF-8 HTML master +9 1 update

5. Install the package “upper-case” using NPM and create a Node.js file that will convert the output "Uki is the best place to learn programming !" into upper-case letters.

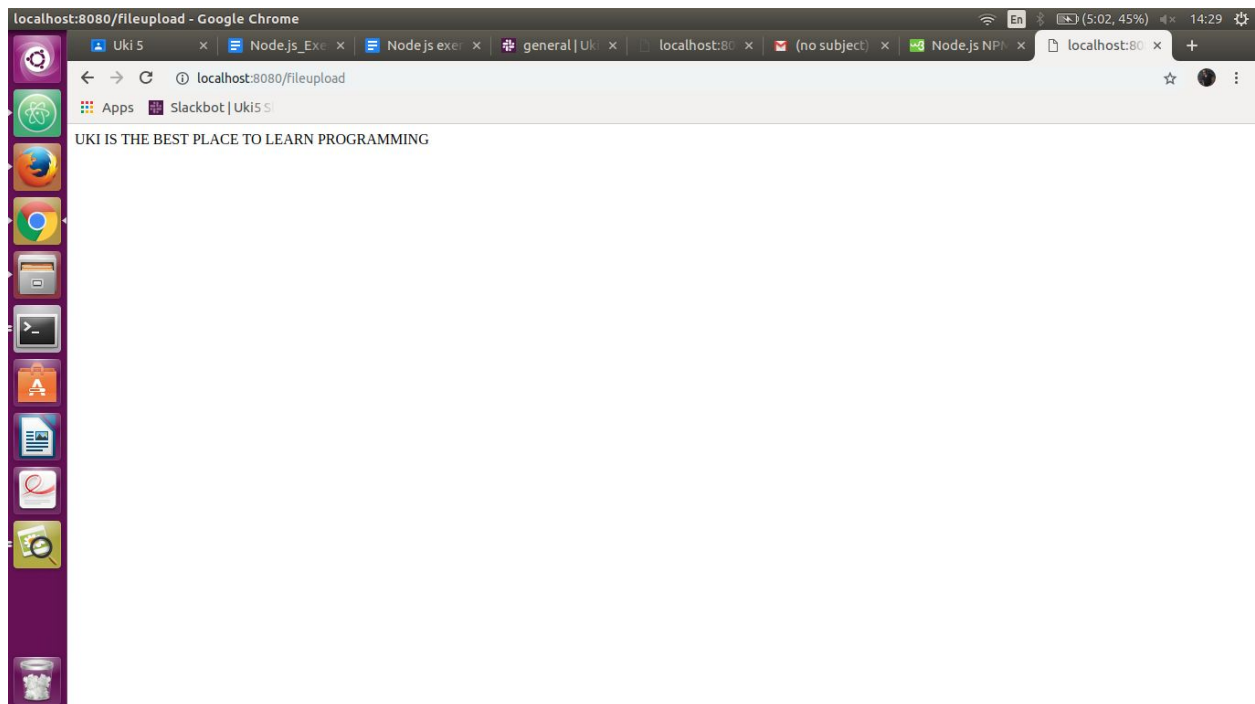


The screenshot shows the Atom editor interface. The left sidebar displays a file explorer with a directory named 'Nodejs' containing several files: calc.js, ex3.1.js, ex3.2.js, ex3.3.js, ex3.4.js, ex3.5.js, ex4.0.js, ex5.0.js (selected), ex7.1.js, ex7.2.js, ex7.3.js, head.html, note.js, tail.html, ukinode.js, and ukinode.txt. The main editor area shows the content of 'ex5.0.js':

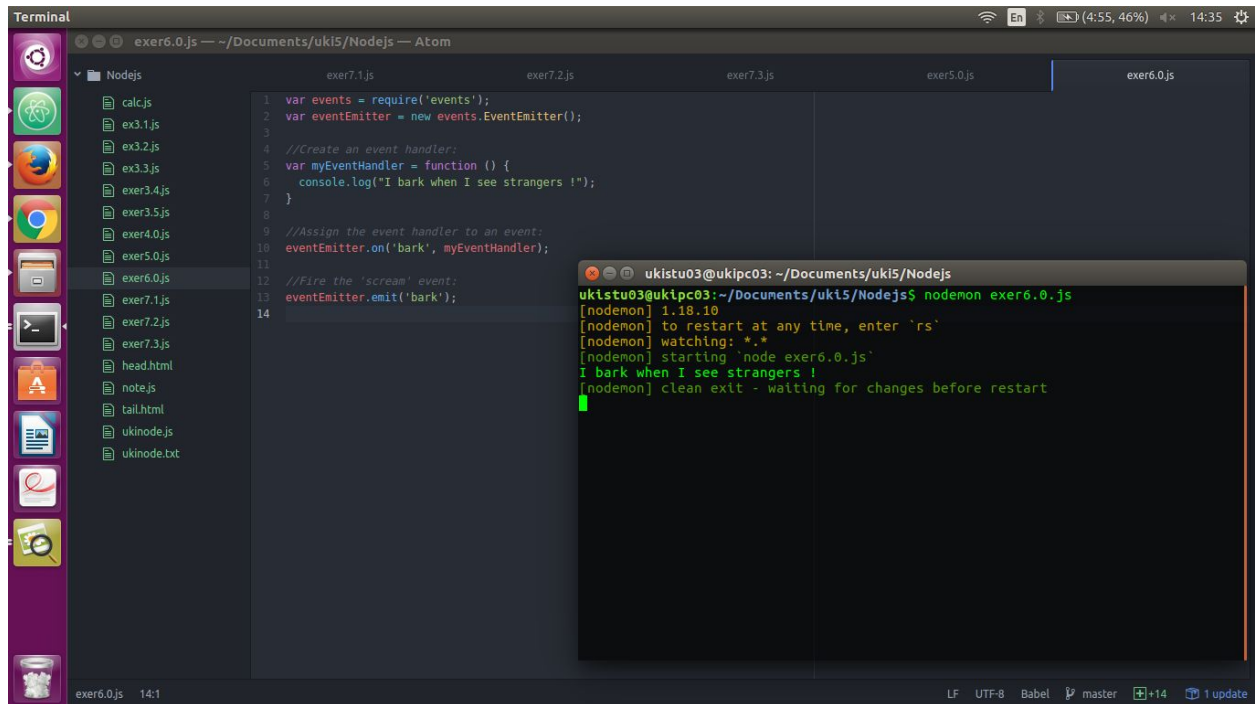
```
1 var http = require('http');
2 var uc = require('upper-case');
3 http.createServer(function (req, res) {
4   res.writeHead(200, {'Content-Type': 'text/html'});
5   /*Use our upper-case module to upper case a string:*/
6   res.write(uc("Uki is the best place to learn programming"));
7   res.end();
8 }).listen(8080);
9
```

A terminal window is open in the foreground, showing the command 'nodemon ex5.0.js' being executed. The terminal output is:

```
ukistu03@ukipc03: ~/Documents/uki5/Nodejs
ukistu03@ukipc03:~/Documents/uki5/Nodejs$ nodemon ex5.0.js
[nodemon] 1.18.10
[nodemon] to restart at any time, enter 'rs'
[nodemon] watching: *.*
[nodemon] starting 'node ex5.0.js'
```



6. Create an event handler function that will say "I bark when I see strangers !" when a "bark" event is fired.



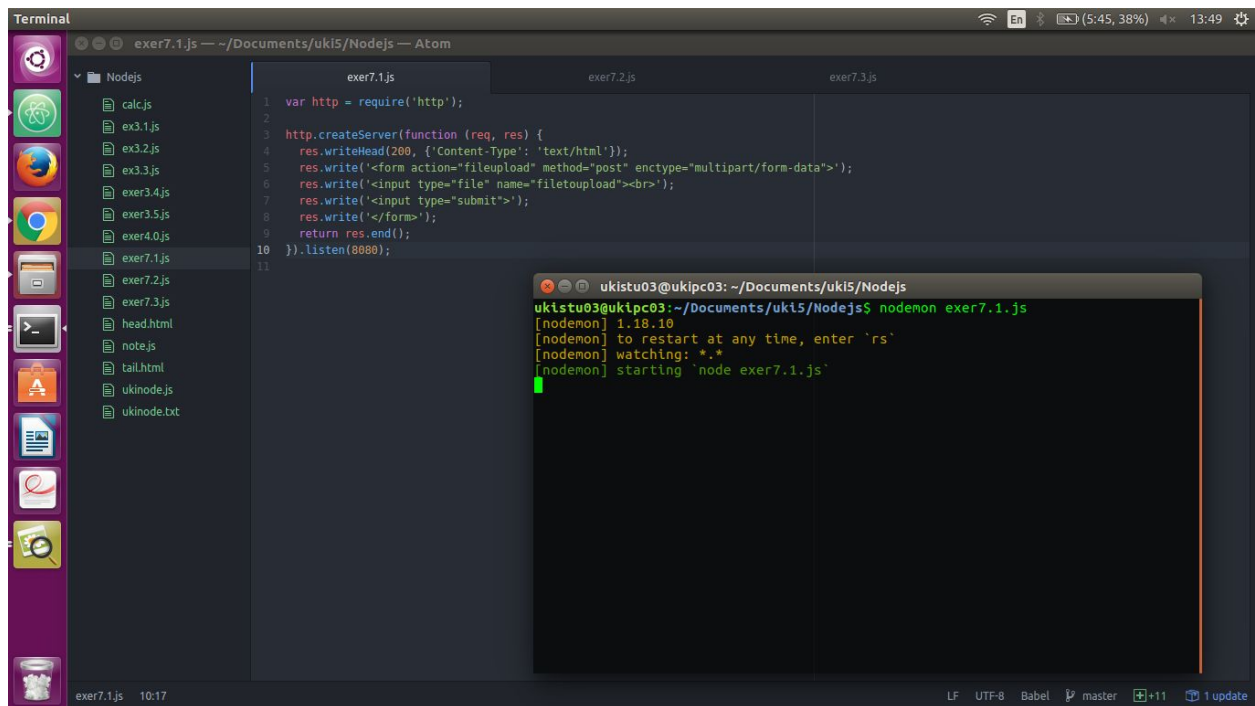
The screenshot shows the Atom text editor with a file explorer on the left and a terminal window in the foreground. The file explorer lists files in a directory named 'Nodejs', including 'ex3.1.js' through 'ex7.3.js', 'head.html', 'note.js', 'tail.html', 'ukinode.js', and 'ukinode.txt'. The editor displays the code for 'exer6.0.js'.

```
1 var events = require('events');
2 var EventEmitter = new events.EventEmitter();
3
4 //Create an event handler:
5 var myEventHandler = function () {
6   console.log("I bark when I see strangers !");
7 }
8
9 //Assign the event handler to an event:
10 EventEmitter.on('bark', myEventHandler);
11
12 //Fire the 'scream' event:
13 EventEmitter.emit('bark');
14
```

The terminal window, titled 'ukistu03@ukipc03: ~/Documents/uki5/Nodejs', shows the output of running 'nodemon exer6.0.js'. The output includes the IP address '1.18.10', a message to restart with 'rs', the file being watched, and the execution of 'node exer6.0.js'. The terminal output matches the console.log statement in the code: 'I bark when I see strangers !'. It also shows 'nodemon' messages for clean exit and waiting for changes.

7. Install “formidable” module using npm and make a web page in Node.js that lets the user upload files to your computer.

7.1 Save that uploaded file into your Documents directory.



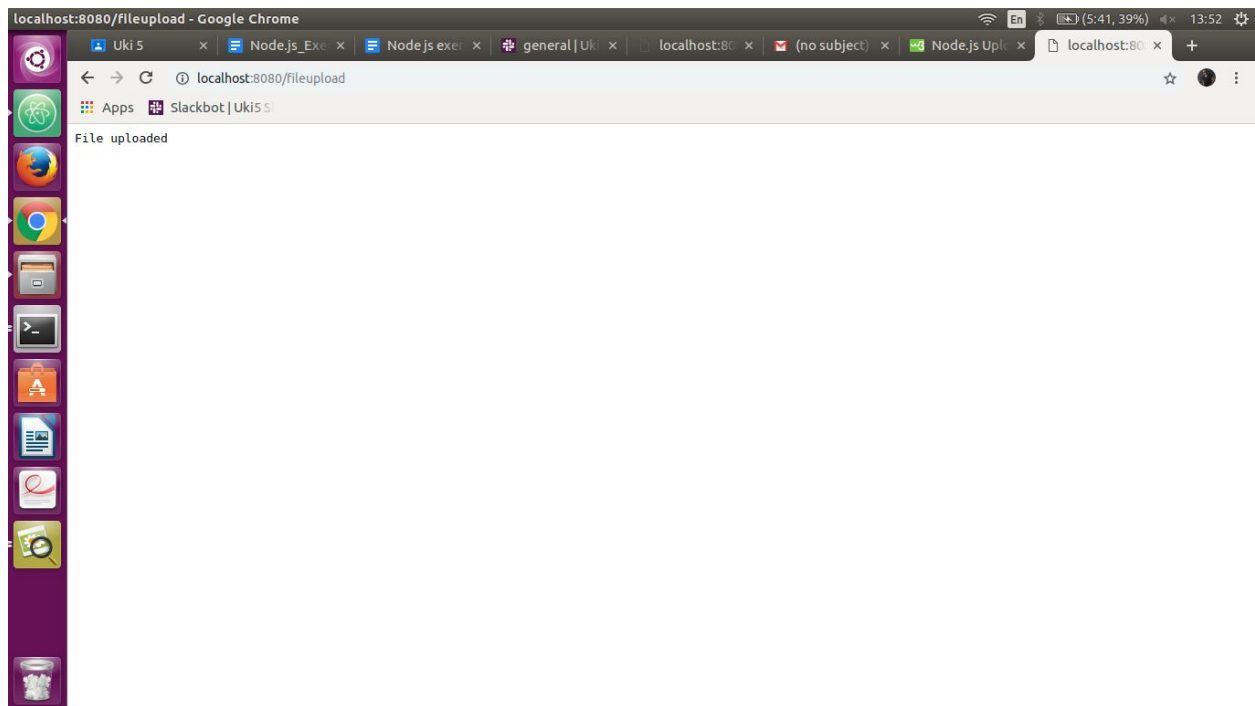
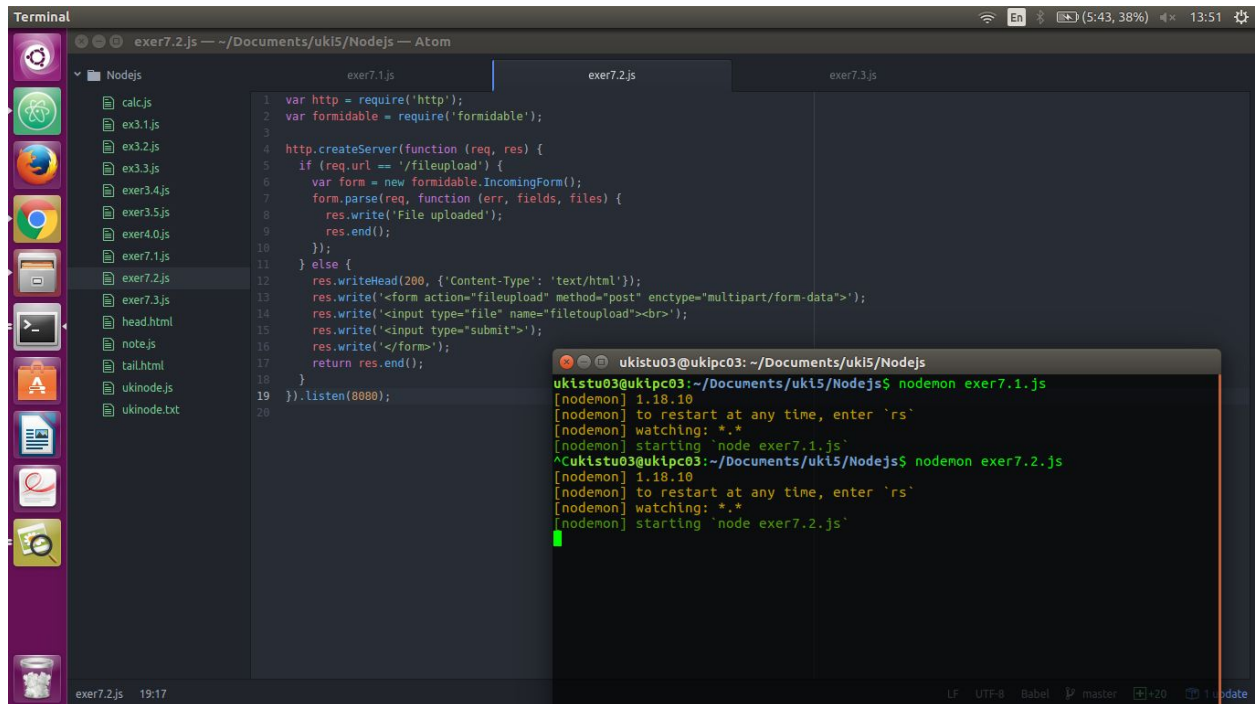
The screenshot shows the Atom text editor with a file explorer on the left and a terminal window in the foreground. The file explorer lists files in a directory named 'Nodejs', including 'calc.js', 'ex3.1.js', 'ex3.2.js', 'ex3.3.js', 'exer3.4.js', 'exer3.5.js', 'exer4.0.js', 'exer7.1.js', 'exer7.2.js', 'exer7.3.js', 'head.html', 'note.js', 'tail.html', 'ukinode.js', and 'ukinode.txt'. The 'exer7.1.js' file is selected. The main editor pane shows the code for 'exer7.1.js':

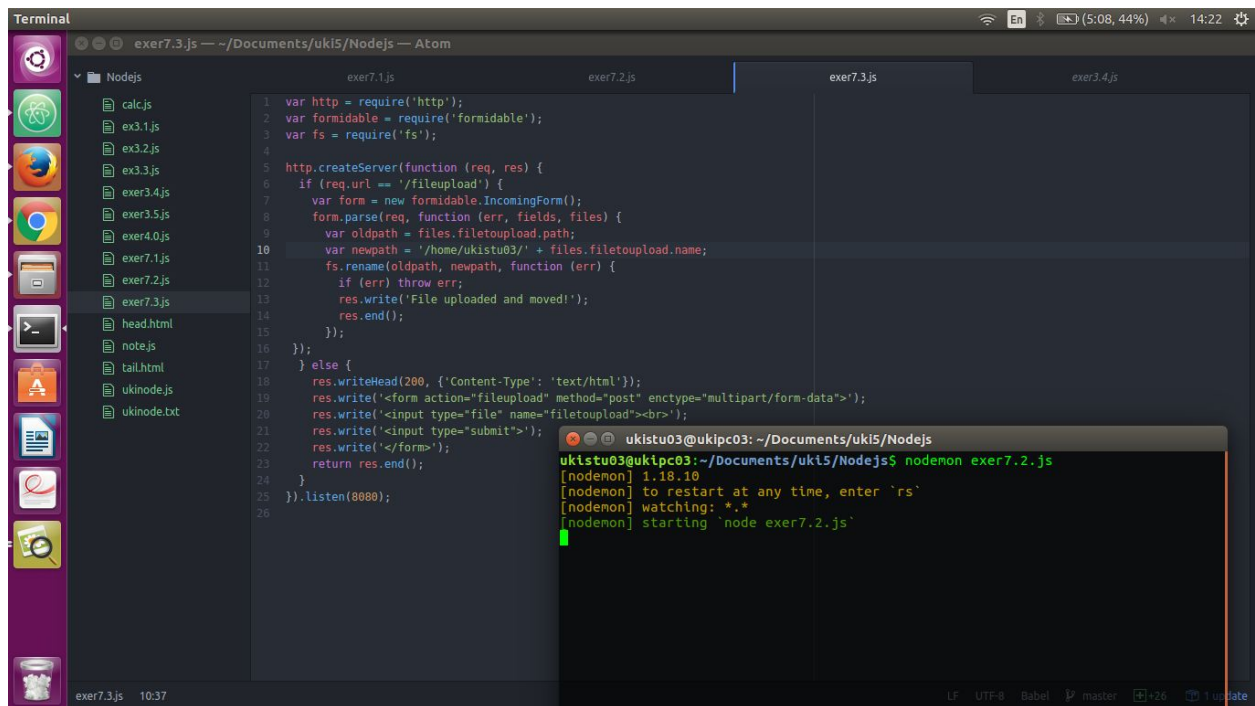
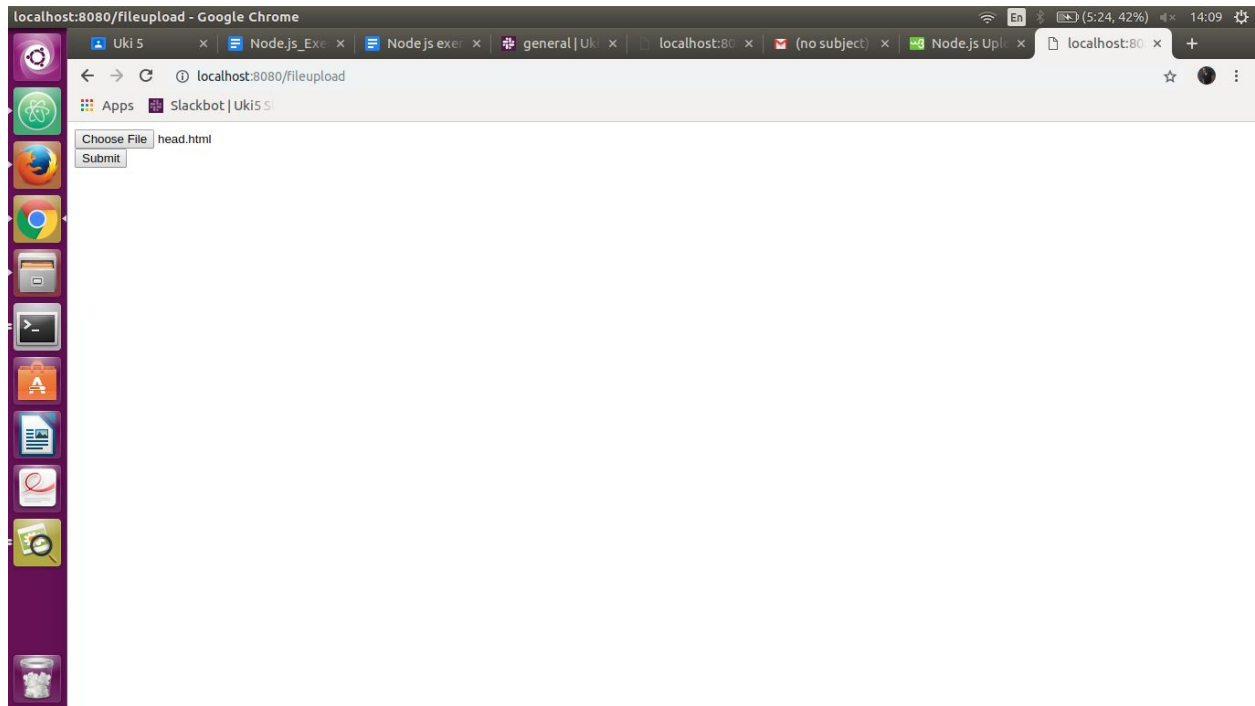
```
1 var http = require('http');
2
3 http.createServer(function (req, res) {
4   res.writeHead(200, {'Content-Type': 'text/html'});
5   res.write('<form action="fileupload" method="post" enctype="multipart/form-data">');
6   res.write('<input type="file" name="fileupload"><br>');
7   res.write('<input type="submit">');
8   res.write('</form>');
9   return res.end();
10 }).listen(8080);
11
```

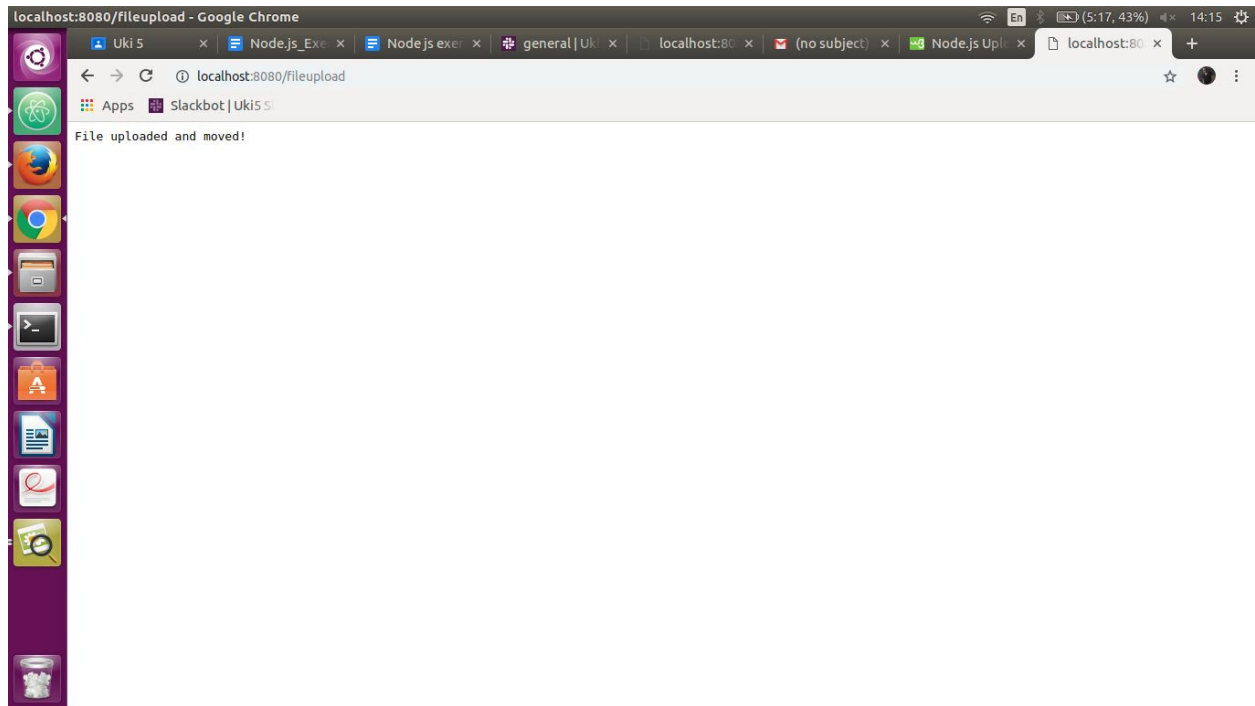
The terminal window, titled 'ukistu03@uklpc03: ~/Documents/uki5/Nodejs', shows the command 'nodemon exer7.1.js' being executed. The output is:

```
ukistu03@uklpc03:~/Documents/uki5/Nodejs$ nodemon exer7.1.js
[nodemon] 1.18.10
[nodemon] to restart at any time, enter `rs`
[nodemon] watching: *.*
[nodemon] starting `node exer7.1.js`
```

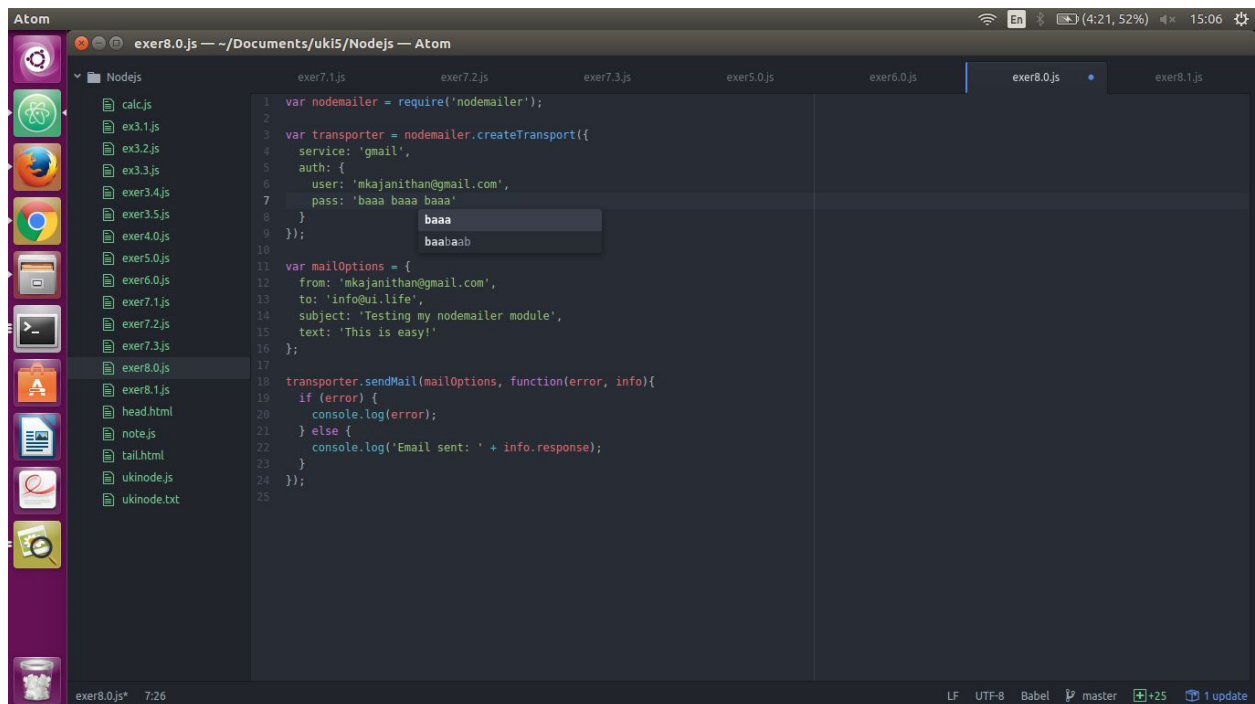
The status bar at the bottom of the Atom editor shows 'exer7.1.js 10:17' and 'LF UTF-8 Babel master +11 1 update'.







8. Using the Nodemailer module create a server and send a mail to `info@uki.life` with the subject : “Testing my nodemailer module” , text: “This is easy !”




```

[nodemon] clean exit - waiting for changes before restart
^Cukistu03@ukipc03:~/Documents/uki5/Nodejs$ nodemon exer8.0.js
[nodemon] 1.18.10
[nodemon] to restart at any time, enter `rs`
[nodemon] watching: *.*
[nodemon] starting `node exer8.0.js`
{ Error: queryA EREFUSED smtp.gmail.com
  at errnoException (dns.js:28:10)
  at QueryReqWrap.onresolve [as oncomplete] (dns.js:219:19)
  code: 'EDNS',
  errno: 'EREFUSED',
  syscall: 'queryA',
  hostname: 'smtp.gmail.com',
  command: 'CONN' }
[nodemon] clean exit - waiting for changes before restart
exer8.0.js 7:21

```

8.1 Now instead of text send a basic html formatted mail.

The screenshot shows a code editor with a file explorer on the left and a terminal on the right. The file explorer shows a directory named 'Nodejs' containing several files, including 'exer8.0.js'. The code editor displays the content of 'exer8.0.js', which uses the 'nodemailer' module to send an email. The email options are configured with a 'from' address, a 'to' address, a 'subject', and an 'html' body containing a welcome message. The terminal shows the output of the 'nodemon' command, which starts the application and displays the error message from the previous screenshot.

```

Terminal
ukistu03@ukipc03: ~/Documents/uki5/Nodejs — Atom
Nodejs
  calc.js
  ex3.1.js
  ex3.2.js
  ex3.3.js
  exer3.4.js
  exer3.5.js
  exer4.0.js
  exer5.0.js
  exer6.0.js
  exer7.1.js
  exer7.2.js
  exer7.3.js
  exer8.0.js
  exer8.1.js
  head.html
  note.js
  tail.html
  ukinode.js
  ukinode.txt
1 var nodemailer = require('nodemailer');
2
3 var transporter = nodemailer.createTransport({
4   service: 'gmail',
5   auth: {
6     user: 'mkajaniathan@gmail.com',
7     pass: 'asdf00107'
8   }
9 });
10
11 var mailOptions = {
12   from: 'mkajaniathan@gmail.com',
13   to: 'info@ui.life',
14   subject: 'Testing my nodemailer module',
15   html: '<h1>Welcome</h1><p>This is basic format mail</p>'
16 };
17
18 transporter.sendMail(mailOptions, function(error, info){
19   if (error) {
20     console.log(error);
21   } else {
22     console.log('Email sent: ' + info.response);
23   }
24 });
25
ukistu03@ukipc03:~/Documents/uki5/Nodejs$ nodemon exer8.0.js
[nodemon] 1.18.10
[nodemon] to restart at any time, enter `rs`
[nodemon] watching: *.*
[nodemon] starting `node exer8.0.js`
{ Error: queryA EREFUSED smtp.gmail.com
  at errnoException (dns.js:28:10)
  at QueryReqWrap.onresolve [as oncomplete] (dns.js:219:19)
  code: 'EDNS',
  errno: 'EREFUSED',
  syscall: 'queryA',
  hostname: 'smtp.gmail.com',
  command: 'CONN' }
[nodemon] clean exit - waiting for changes before restart
exer8.0.js 7:21

```