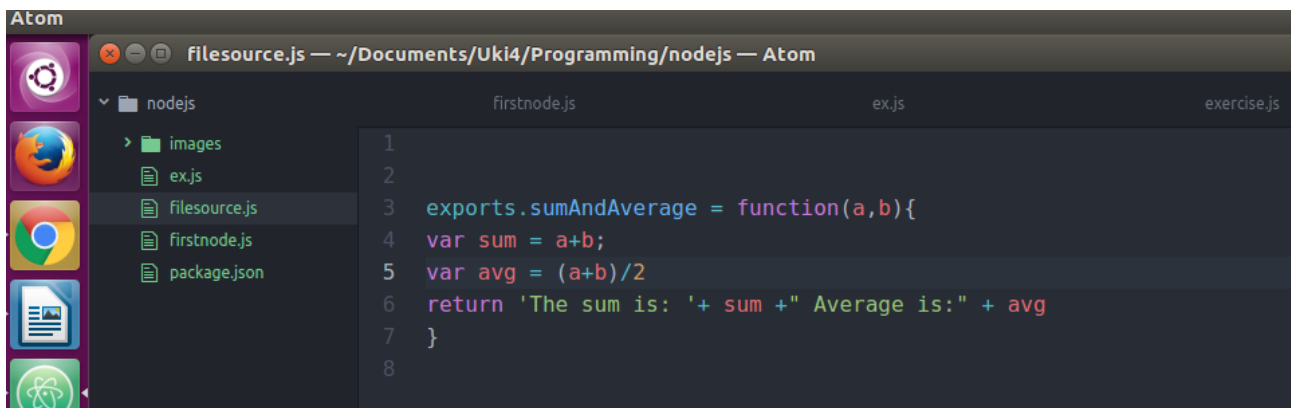


1. Create a custom module which returns the sum and average of any two numbers passed into it. Require the module and run the server by passing 123 and 321 so that the server prints out the sum and average.

```
Code: exports.sumAndAverage = function(a,b){  
var sum = a+b;  
var avg = a+b/2  
return 'The sum is: ' + sum + " Average is:" + avg  
}  
var http = require('http');  
var sums = require('./filesource')  
http.createServer(function (req, res) {  
res.writeHead(200, {'Content-Type': 'text/html'});  
res.write('Sum of two numbers is :' + sums.sumAndAverage(123, 321));  
res.end();  
}).listen(8000);
```



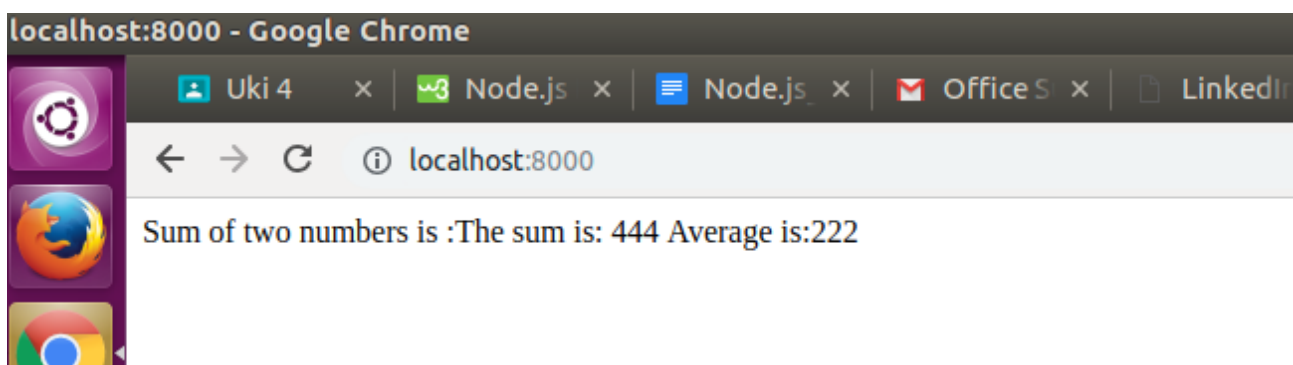
A screenshot of the Atom text editor. The left sidebar shows a file explorer with a folder named 'nodejs' containing files 'ex.js', 'filesource.js', 'firstnode.js', and 'package.json'. The main editor area has 'filesource.js' open, displaying the following JavaScript code:

```
1  
2  
3 exports.sumAndAverage = function(a,b){  
4   var sum = a+b;  
5   var avg = (a+b)/2  
6   return 'The sum is: ' + sum + " Average is:" + avg  
7 }  
8
```



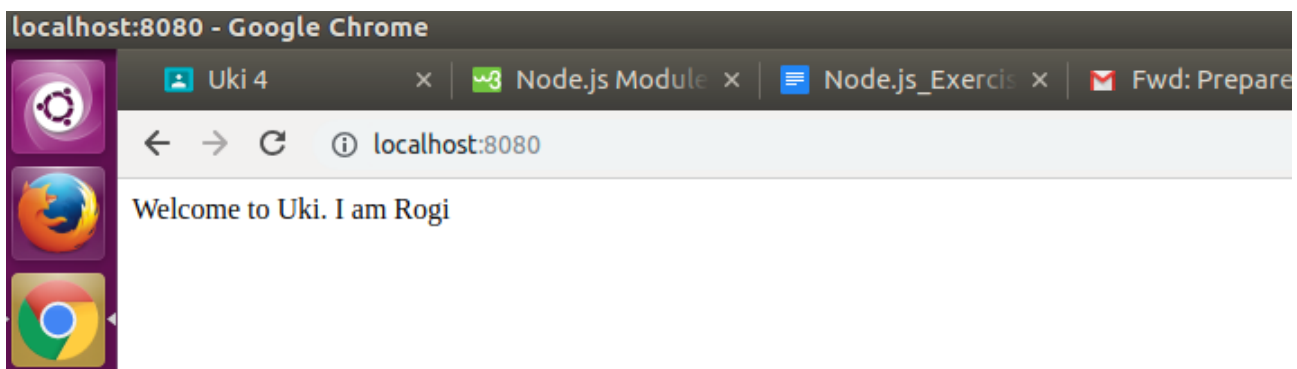
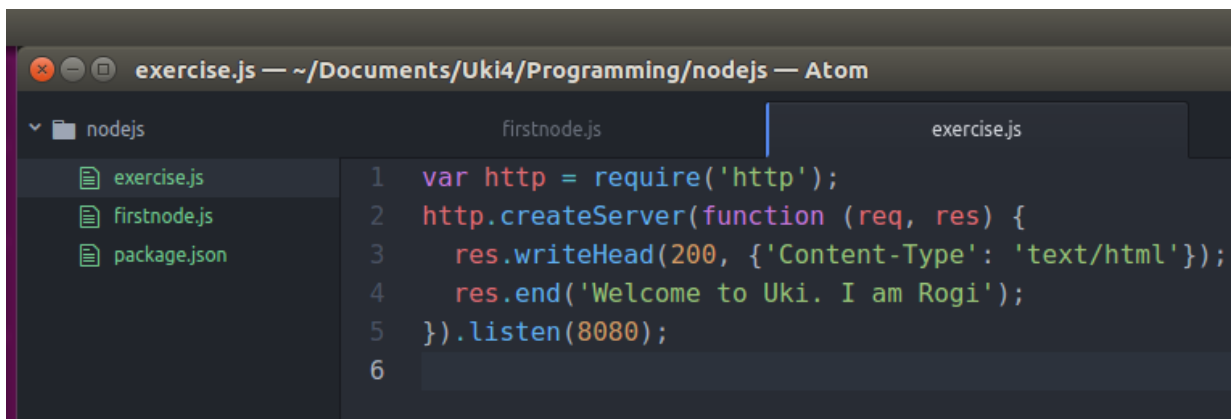
A screenshot of the Atom text editor. The left sidebar shows the same file explorer. The main editor area has 'ex.js' open, displaying the following JavaScript code:

```
1 var http = require('http');  
2 var sums = require('./filesource')  
3 http.createServer(function (req, res) {  
4   res.writeHead(200, {'Content-Type': 'text/html'});  
5   res.write('Sum of two numbers is :' + sums.sumAndAverage(123, 321));  
6   res.end();  
7 }).listen(8000);  
8
```



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)

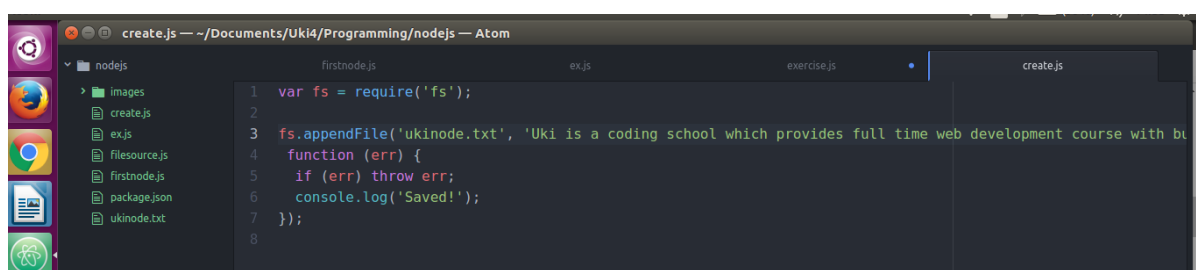
```
code: var http = require('http');
      http.createServer(function (req, res) {
        res.writeHead(200, {'Content-Type': 'text/html'});
        res.end('Welcome to Uki. I am Rogi');
      }).listen(8080);
```



3. Using the file system module create a new file called ukinode.txt

3.1 Write a paragraph about Uki into that file

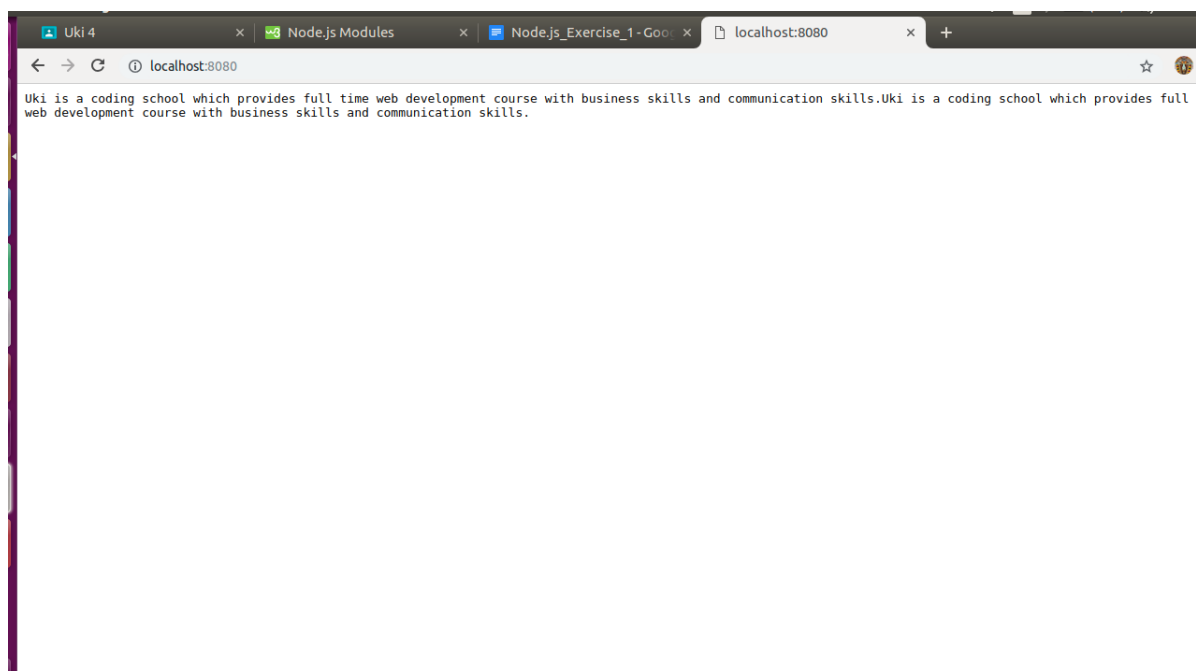
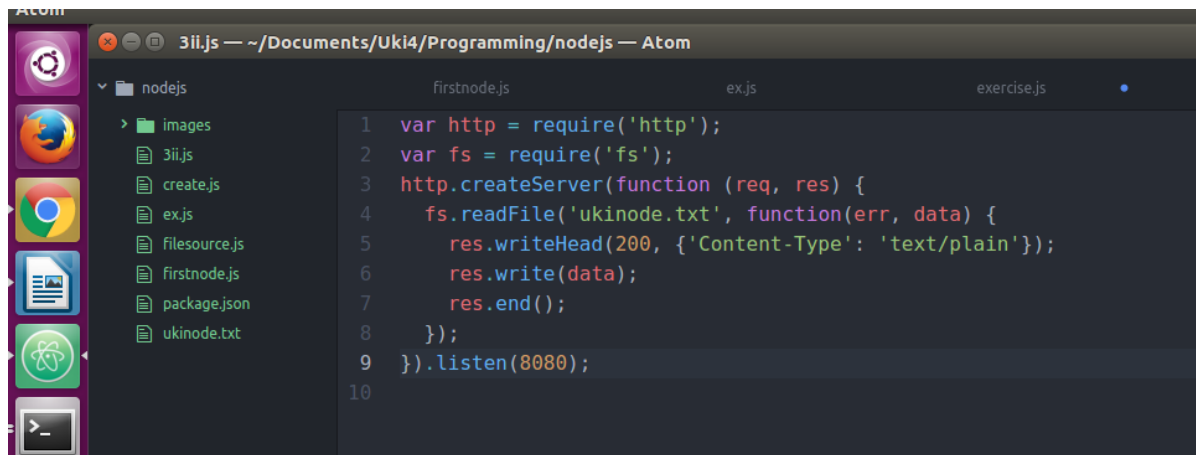
```
code: var fs = require('fs');
fs.appendFile('ukinode.txt', 'Uki is a coding school which provides full
time web development course with business skills and communication skills.',
function (err) {
  if (err) throw err;
  console.log('Saved!');
});
```



```
ukistu03@ukipc03: ~/Documents/Uki4/Programming/nodejs
ukistu03@ukipc03:~/Documents/Uki4/Programming/nodejs$ node create.js
Saved!
ukistu03@ukipc03:~/Documents/Uki4/Programming/nodejs$
```

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

```
var http = require('http');
var fs = require('fs');
http.createServer(function (req, res) {
  fs.readFile('ukinode.txt', function(err, data) {
    res.writeHead(200, {'Content-Type': 'text/plain'});
    res.write(data);
    res.end();
  });
}).listen(8080);
```



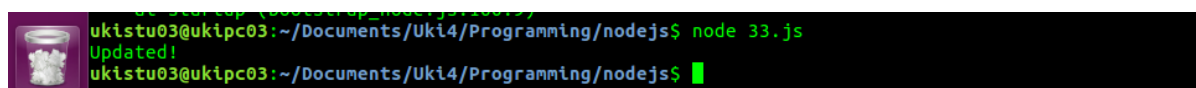
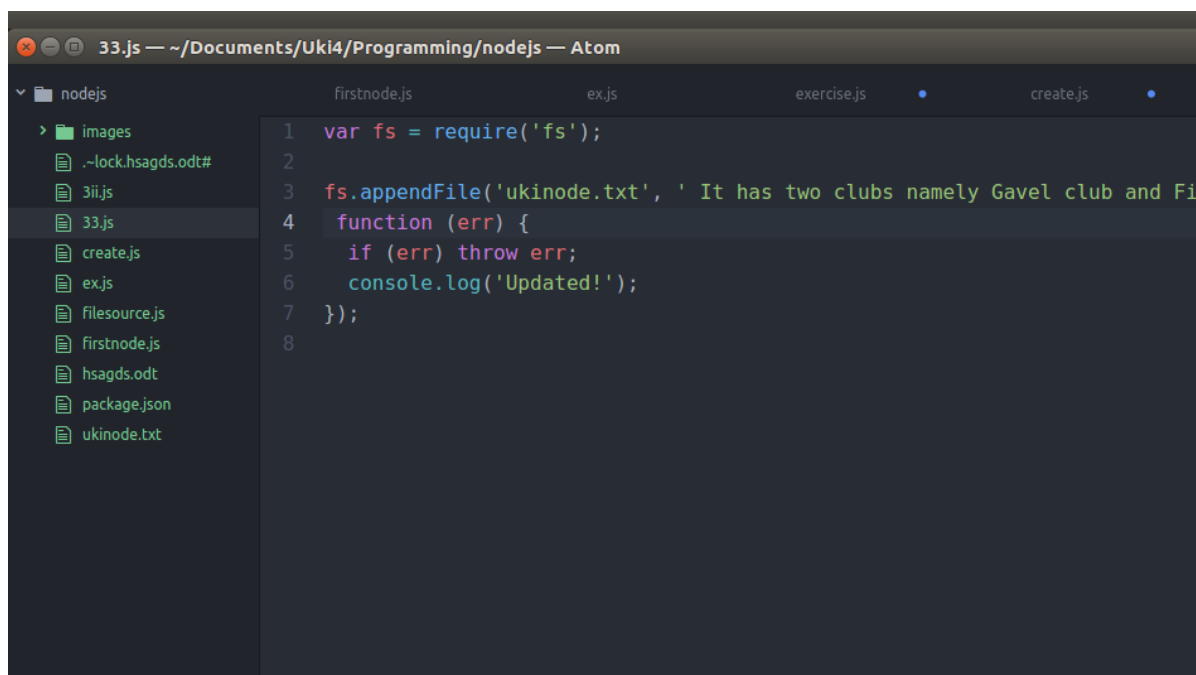
3.3 Append another paragraph about Uki and now serve the new file

```
code: var fs = require('fs');
```

```
fs.appendFile('ukinode.txt', ' It has two clubs namely Gavel club and  
Fitness club.Gavel club develops the student communication skills in a  
formal manner and fitness club improve the fitness of the students and  
celebrate some festival with uki family.',
```

```
function (err) {  
  if (err) throw err;  
  console.log('Updated!');
```

```
});
```

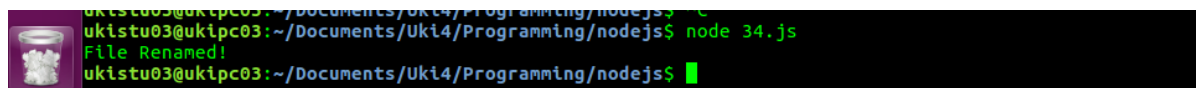


3.4 Rename the file as ukinodejsexercise1.txt

```
code: var fs = require('fs');
```

```
fs.rename('ukinode.txt', 'ukinodejsexercise1.txt', function (err) {  
  if (err) throw err;  
  console.log('File Renamed!');
```

```
});
```

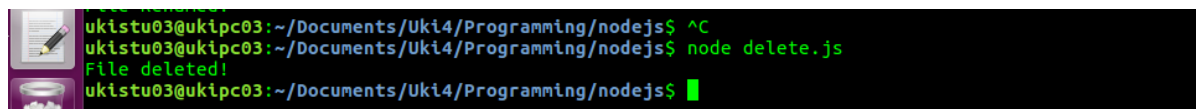




3.5 Delete the file you created

code: `var fs = require('fs');`

```
fs.unlink('ukinodejsexercise1.txt', function (err) {
  if (err) throw err;
  console.log('File deleted!');
});
```



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.

Code: `console.log("Request for " + pathname + " received.");`

```
// Read the requested file content from file system
fs.readFile(pathname.substr(1), function (err, data) {
  if (err) {
    console.log(err);

    // HTTP Status: 404 : NOT FOUND
    // Content Type: text/plain
```

```

        response.writeHead(404, {'Content-Type': 'text/html'});
    } else {
        //Page found
        // HTTP Status: 200 : OK
        // Content Type: text/plain
        response.writeHead(200, {'Content-Type': 'text/html'});

        // Write the content of the file to response body
        response.write(data.toString());
    }

    // Send the response body
    response.end();
});
}).listen(8080);

// Console will print the message
console.log('Server running at http://127.0.0.1:8080/');
```

```

new.js — ~/Documents/Uki4/Programming/nodejs — Atom
11 console.log("Request for " + pathname + " received.");
12
13 // Read the requested file content from file system
14 fs.readFile(pathname.substr(1), function (err, data) {
15     if (err) {
16         console.log(err);
17     }
18     // HTTP Status: 404 : NOT FOUND
19     // Content Type: text/plain
20     response.writeHead(404, {'Content-Type': 'text/html'});
21 } else {
22     //Page found
23     // HTTP Status: 200 : OK
24     // Content Type: text/plain
25     response.writeHead(200, {'Content-Type': 'text/html'});
26
27     // Write the content of the file to response body
28     response.write(data.toString());
29 }
30
31 // Send the response body
32 response.end();
33 });
34 }).listen(8080);
35
36 // Console will print the message
37 console.log('Server running at http://127.0.0.1:8080/');
```



This localhost page can't be found

No webpage was found for the web address: **http://localhost:8080/**

HTTP ERROR 404

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

```
code:var http = require('http');
```

```
// Options to be used by request
```

```
var options = {  
  host: 'localhost',  
  port: '8080',  
  path: '/head.html'  
};
```

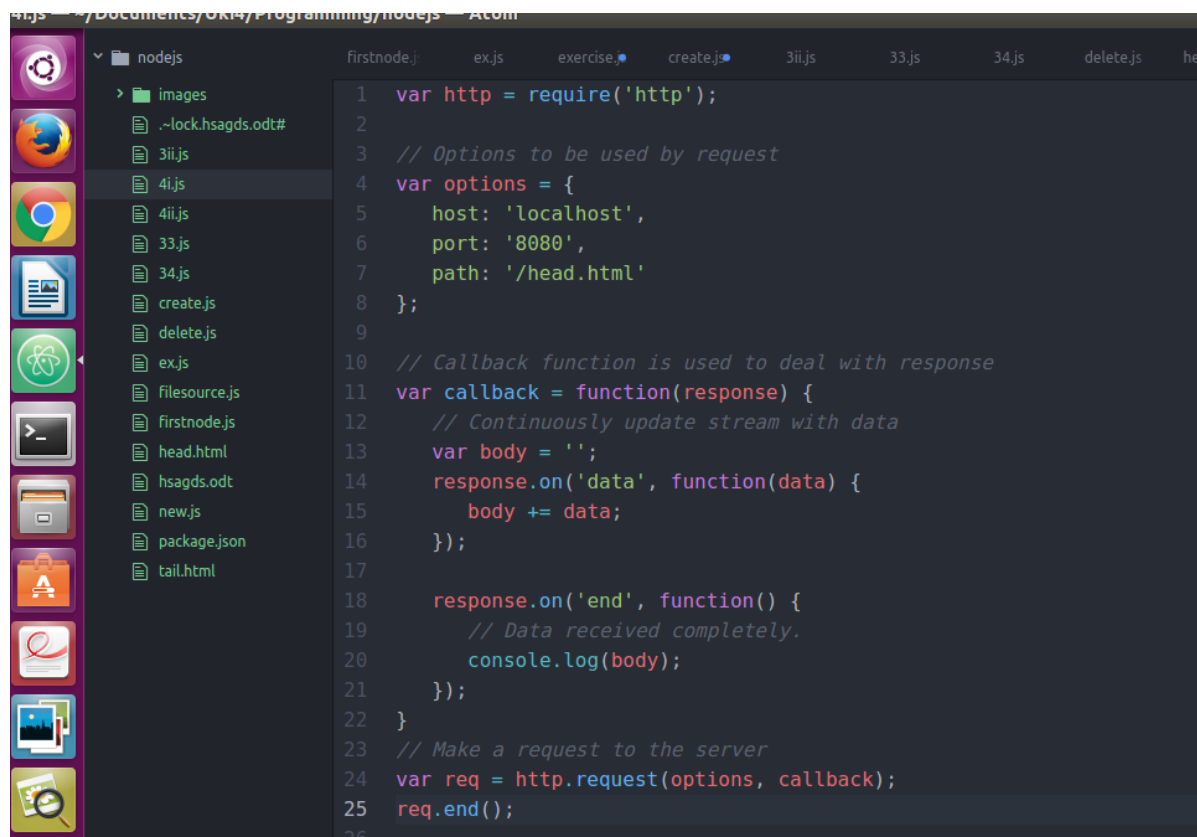
```
// Callback function is used to deal with response
```

```
var callback = function(response) {  
  // Continuously update stream with data  
  var body = '';  
  response.on('data', function(data) {  
    body += data;  
  });
```

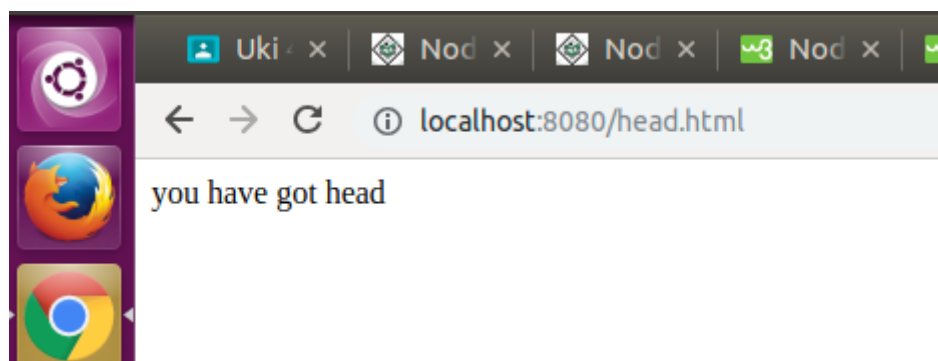
```
  response.on('end', function() {  
    // Data received completely.  
    console.log(body);  
  });  
}
```

```
// Make a request to the server
```

```
var req = http.request(options, callback);  
req.end();
```



```
1 var http = require('http');
2
3 // Options to be used by request
4 var options = {
5   host: 'localhost',
6   port: '8080',
7   path: '/head.html'
8 };
9
10 // Callback function is used to deal with response
11 var callback = function(response) {
12   // Continuously update stream with data
13   var body = '';
14   response.on('data', function(data) {
15     body += data;
16   });
17
18   response.on('end', function() {
19     // Data received completely.
20     console.log(body);
21   });
22 }
23 // Make a request to the server
24 var req = http.request(options, callback);
25 req.end();
26
```



<http://localhost:8080/tail.html> -> You have got tail
code:var http = require('http');

// Options to be used by request

```
var options = {
  host: 'localhost',
  port: '8080',
  path: '/tail.html'
};
```

// Callback function is used to deal with response

```
var callback = function(response) {
  // Continuously update stream with data
```



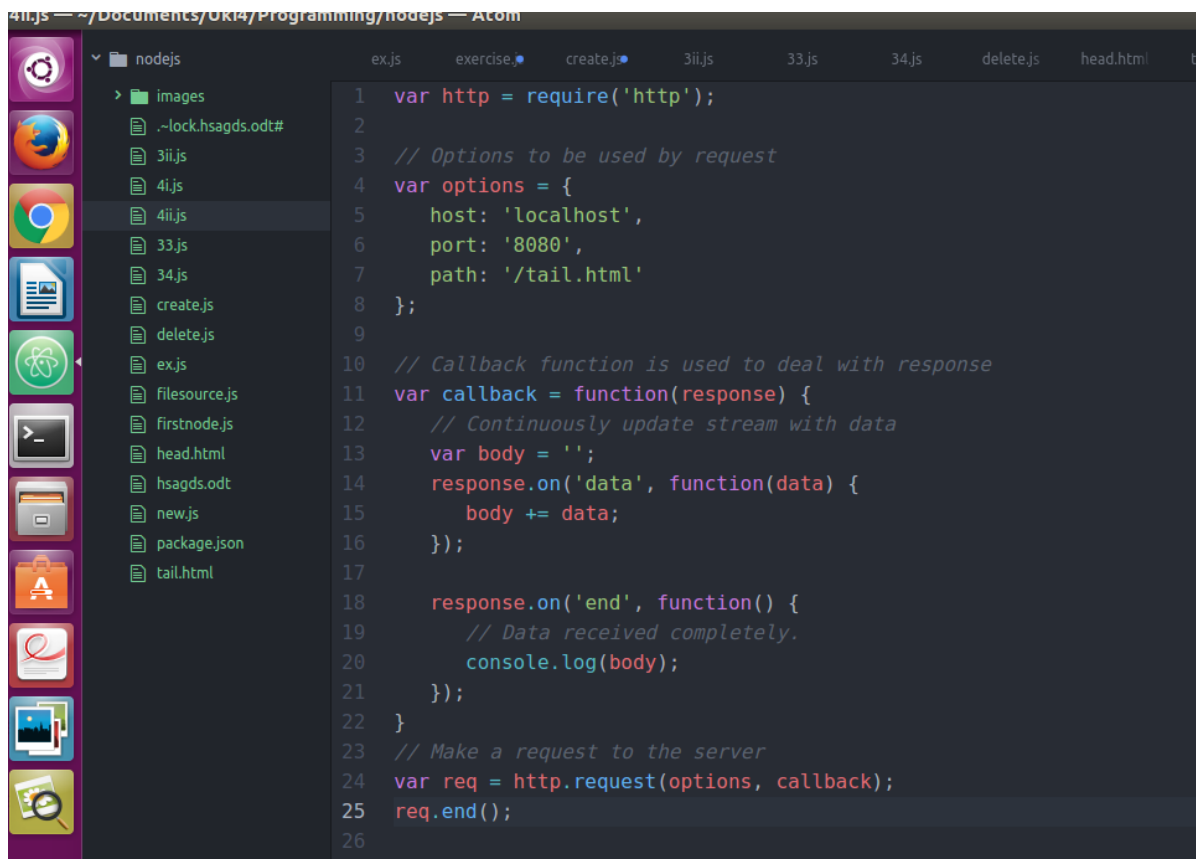
```

var body = '';
response.on('data', function(data) {
    body += data;
});

response.on('end', function() {
    // Data received completely.
    console.log(body);
});
}

// Make a request to the server
var req = http.request(options, callback);
req.end();

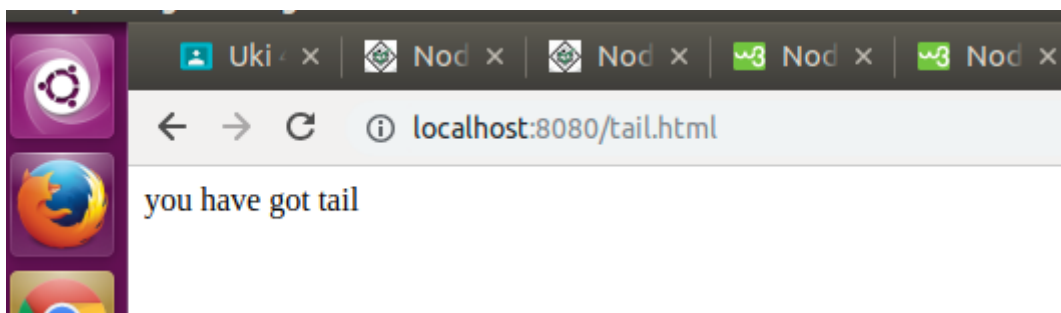
```



```

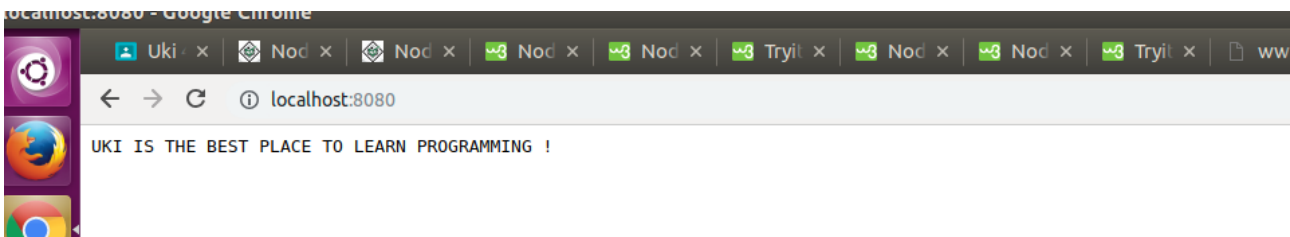
1  var http = require('http');
2
3  // Options to be used by request
4  var options = {
5      host: 'localhost',
6      port: '8080',
7      path: '/tail.html'
8  };
9
10 // Callback function is used to deal with response
11 var callback = function(response) {
12     // Continuously update stream with data
13     var body = '';
14     response.on('data', function(data) {
15         body += data;
16     });
17
18     response.on('end', function() {
19         // Data received completely.
20         console.log(body);
21     });
22 }
23 // Make a request to the server
24 var req = http.request(options, callback);
25 req.end();
26

```



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.

```
Code:var http = require('http');
var uc = require('upper-case');
http.createServer(function (req, res) {
  res.writeHead(200, {'Content-Type': 'text/plain'});
  res.write(uc("Uki is the best place to learn programming !"));
  res.end();
}).listen(8080);
```



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

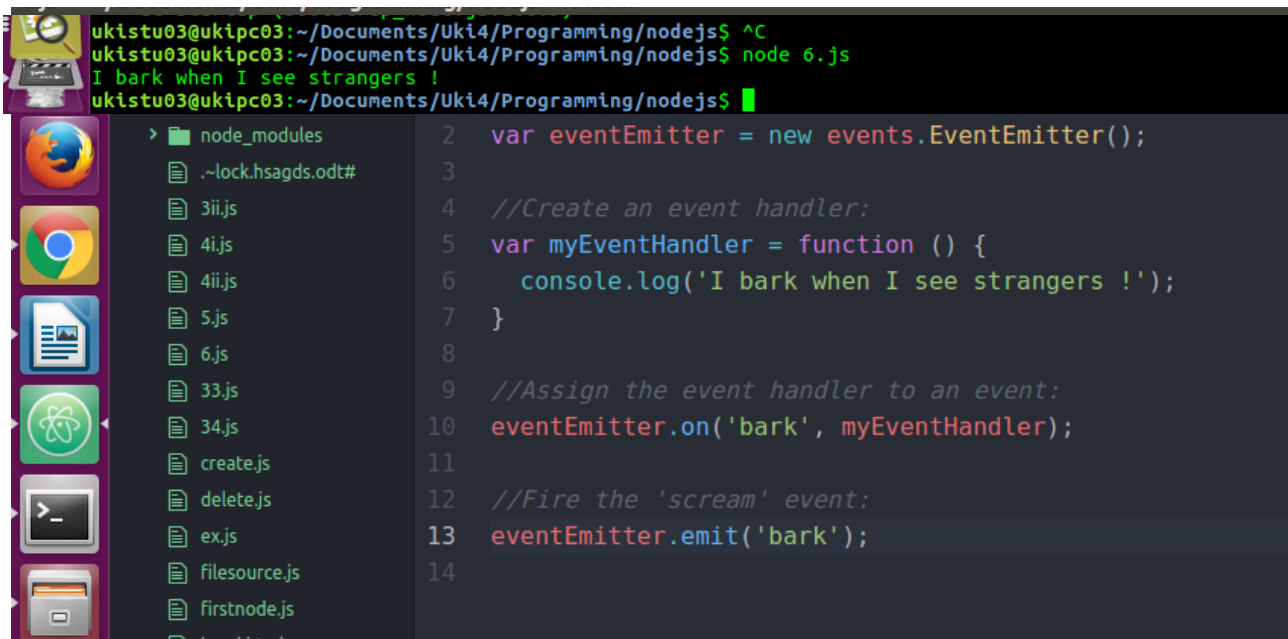
```
Code:var events = require('events');
var EventEmitter = new events.EventEmitter();
```

```
//Create an event handler:
var myEventHandler = function () {
  console.log('I bark when I see strangers !');
}
```

```
//Assign the event handler to an event:
eventEmitter.on('bark', myEventHandler);
```

```
//Fire the 'scream' event:
```

```
eventEmitter.emit('bark');
```



The screenshot shows a terminal window at the top and a file explorer below it. The terminal shows the user running 'node 6.js' in the directory ~/Documents/Uki4/Programming/nodejs, which outputs 'I bark when I see strangers !'. The file explorer shows a directory with various files, including 3ii.js, 4i.js, 4ii.js, 5.js, 6.js, 33.js, 34.js, create.js, delete.js, ex.js, filesource.js, firstnode.js, and head.html. The file 6.js is selected, and its contents are displayed in the right pane.

```
ukistu03@ukipc03:~/Documents/Uki4/Programming/nodejs$ ^C
ukistu03@ukipc03:~/Documents/Uki4/Programming/nodejs$ node 6.js
I bark when I see strangers !
ukistu03@ukipc03:~/Documents/Uki4/Programming/nodejs$
```

```
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
```

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

```
var http = require('http');
```

```
var formidable = require('formidable');
```

```
var fs = require('fs');
```

```
http.createServer(function (req, res) {
```

```
  if (req.url === '/fileupload') {
```

```
    var form = new formidable.IncomingForm();
```

```
    form.parse(req, function (err, fields, files) {
```

```
      var oldpath = files.fileupload.path;
```

```
      var newpath = '/home/ukistu03/Pictures/ima/raw_2F3f88b367-d27f-4b26-  
ad61-5e7f89a09204_2Fyeti-jquery (1).png/' + files.fileupload.name;
```

```
      fs.rename(oldpath, newpath, function (err) {
```

```
        if (err) throw err;
```

```
        res.write('File uploaded and moved!');
```

```
        res.end();
```

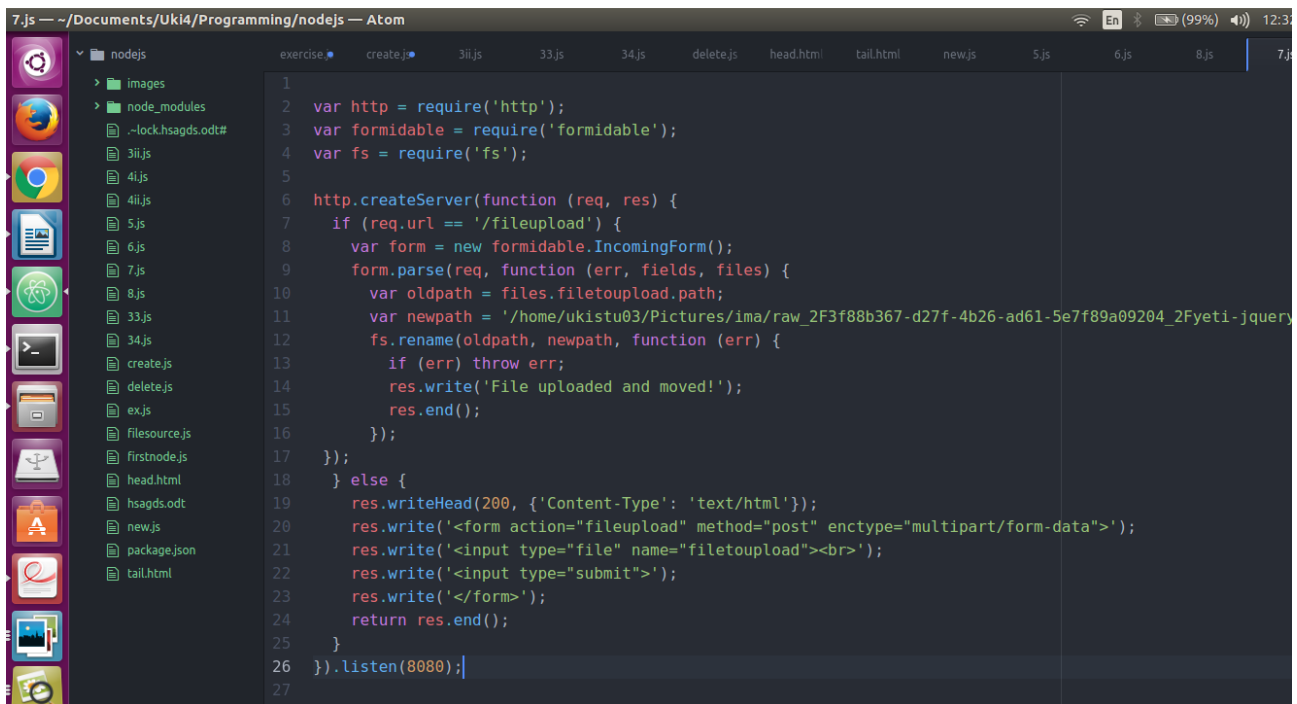
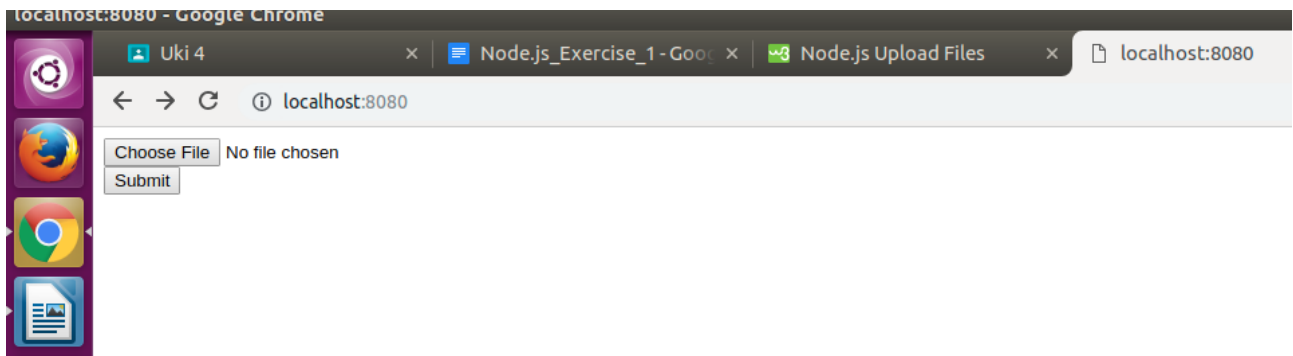
```
      });
```

```
    });
```

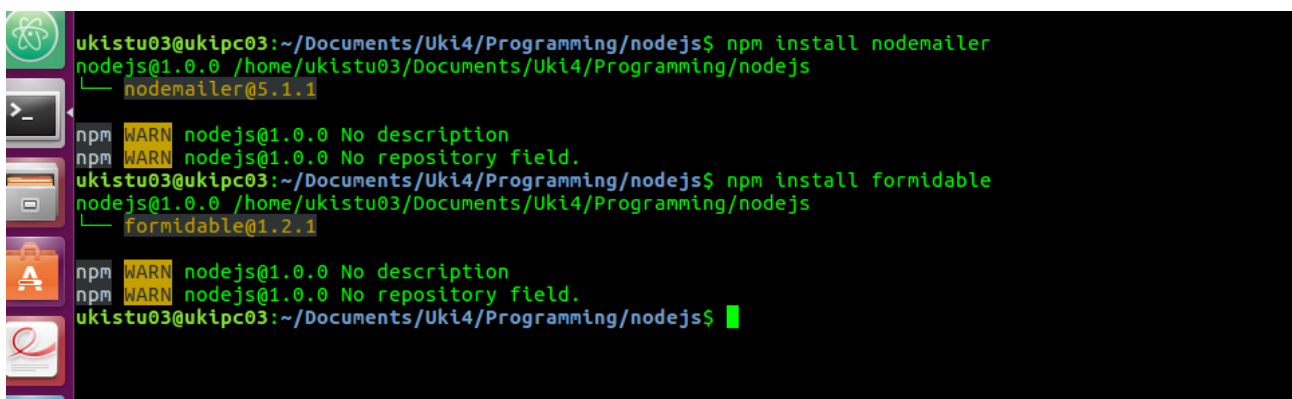
```
  } else {
```

```
res.writeHead(200, {'Content-Type': 'text/html'});
res.write('<form action="fileupload" method="post"
enctype="multipart/form-data">');
res.write('<input type="file" name="filetoupload"><br>');
res.write('<input type="submit">');
res.write('</form>');
return res.end();
}
}).listen(8080);
```

A terminal window with a dark background and green text. The window title is 'ukistu03@ukipc03: ~/Documents/Uki4/Programming/nodejs'. The terminal shows the following commands and output:
1. Command: `npm install upper-case -g`
Output: `/home/ukistu03/.npm/versions/node/v6.16.0/lib`
2. Command: `npm install upper-case`
Output: `nodejs@1.0.0 /home/ukistu03/Documents/Uki4/Programming/nodejs`
3. Command: `npm install formidable -g`
Output: `/home/ukistu03/.npm/versions/node/v6.16.0/lib`
The terminal also shows two warning messages: `npm WARN nodejs@1.0.0 No description` and `npm WARN nodejs@1.0.0 No repository field.`
The prompt `ukistu03@ukipc03:~/Documents/Uki4/Programming/nodejs$` is visible at the end of the last line.



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 !”



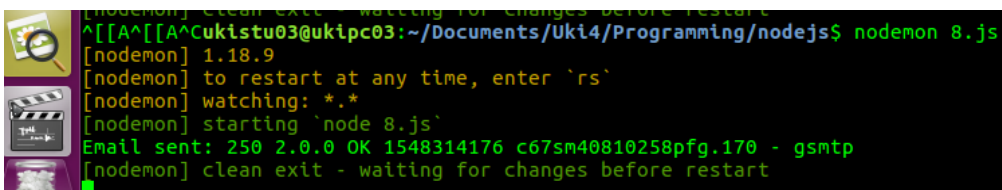
code: var nodemailer = require('nodemailer');

var transporter = nodemailer.createTransport({

```
service: 'gmail',  
auth: {  
  user: 'soruby1298@gmail.com',  
  pass: '*****'  
}  
});
```

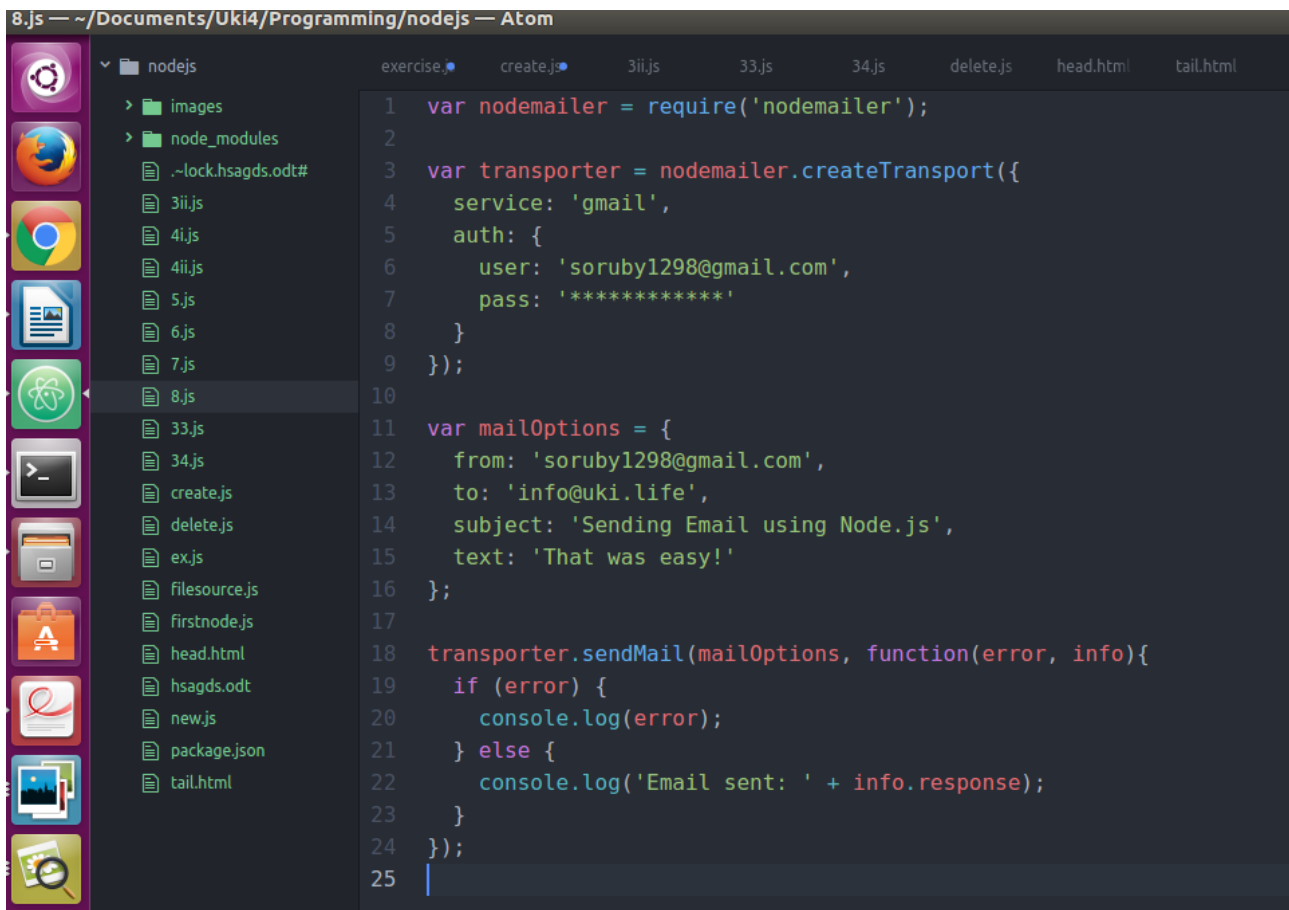
```
var mailOptions = {  
  from: 'soruby1298@gmail.com',  
  to: 'info@uki.life',  
  subject: 'Sending Email using Node.js',  
  text: 'That was easy!'  
};
```

```
transporter.sendMail(mailOptions, function(error, info){  
  if (error) {  
    console.log(error);  
  } else {  
    console.log('Email sent: ' + info.response);  
  }  
});
```



A terminal window with a dark background and green text. The prompt shows the user is in the directory ~/Documents/Uki4/Programming/nodejs. The command 'nodemon 8.js' has been executed. The output shows nodemon version 1.18.9, watching for file changes, and starting node 8.js. A green line of output indicates 'Email sent: 250 2.0.0 OK 1548314176 c67sm40810258pfg.170 - gsmtip'. The terminal ends with 'nodemon] clean exit - waiting for changes before restart'.

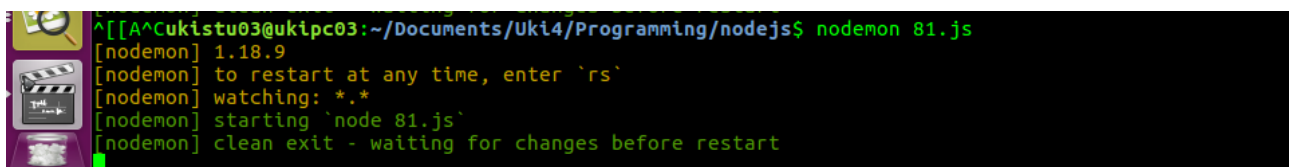
```
[nodemon] clean exit - waiting for changes before restart  
^[[A^[[A^Cukistu03@ukipc03:~/Documents/Uki4/Programming/nodejs$ nodemon 8.js  
[nodemon] 1.18.9  
[nodemon] to restart at any time, enter `rs`  
[nodemon] watching: *.*  
[nodemon] starting `node 8.js`  
Email sent: 250 2.0.0 OK 1548314176 c67sm40810258pfg.170 - gsmtip  
[nodemon] clean exit - waiting for changes before restart
```

A screenshot of the Atom text editor. The left sidebar shows a file explorer with a folder named 'nodejs' containing several files: images, node_modules, .lock.hsagds.odt#, 3ii.js, 4i.js, 4ii.js, 5.js, 6.js, 7.js, 8.js (selected), 33.js, 34.js, create.js, delete.js, ex.js, filesource.js, firstnode.js, head.html, hsagds.odt, new.js, package.json, and tail.html. The main editor area shows the code for 8.js, which uses nodemailer to send an email via Gmail. The code is as follows:

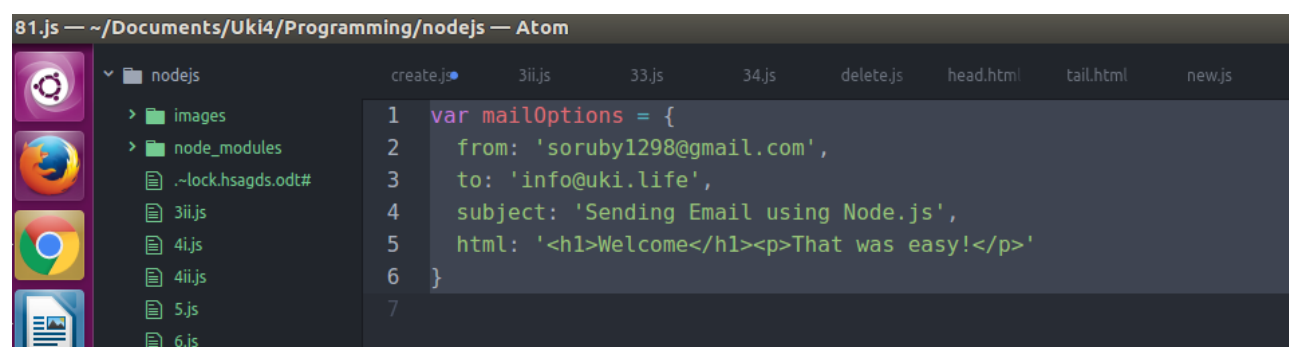
```
1 var nodemailer = require('nodemailer');
2
3 var transporter = nodemailer.createTransport({
4   service: 'gmail',
5   auth: {
6     user: 'soruby1298@gmail.com',
7     pass: '*****'
8   }
9 });
10
11 var mailOptions = {
12   from: 'soruby1298@gmail.com',
13   to: 'info@uki.life',
14   subject: 'Sending Email using Node.js',
15   text: 'That was easy!'
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
```

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

Code:var mailOptions = {
 from: 'soruby1298@gmail.com',
 to: 'info@uki.life',
 subject: 'Sending Email using Node.js',
 html: '<h1>Welcome</h1><p>That was easy!</p>'
}

A screenshot of a terminal window showing the output of the nodemon command. The prompt is '^[[A^Cukistu03@ukipc03:~/Documents/Uki4/Programming/nodejs\$'. The output is:

```
nodemon 1.18.9
to restart at any time, enter `rs`
watching: *.*
starting `node 81.js`
clean exit - waiting for changes before restart
```

A screenshot of the Atom text editor. The left sidebar shows the same file explorer as before, but with 'new.js' added to the list. The main editor area shows the code for 81.js, which is similar to 8.js but uses the 'html' property for the email content. The code is as follows:

```
1 var mailOptions = {
2   from: 'soruby1298@gmail.com',
3   to: 'info@uki.life',
4   subject: 'Sending Email using Node.js',
5   html: '<h1>Welcome</h1><p>That was easy!</p>'
6 }
7
```

