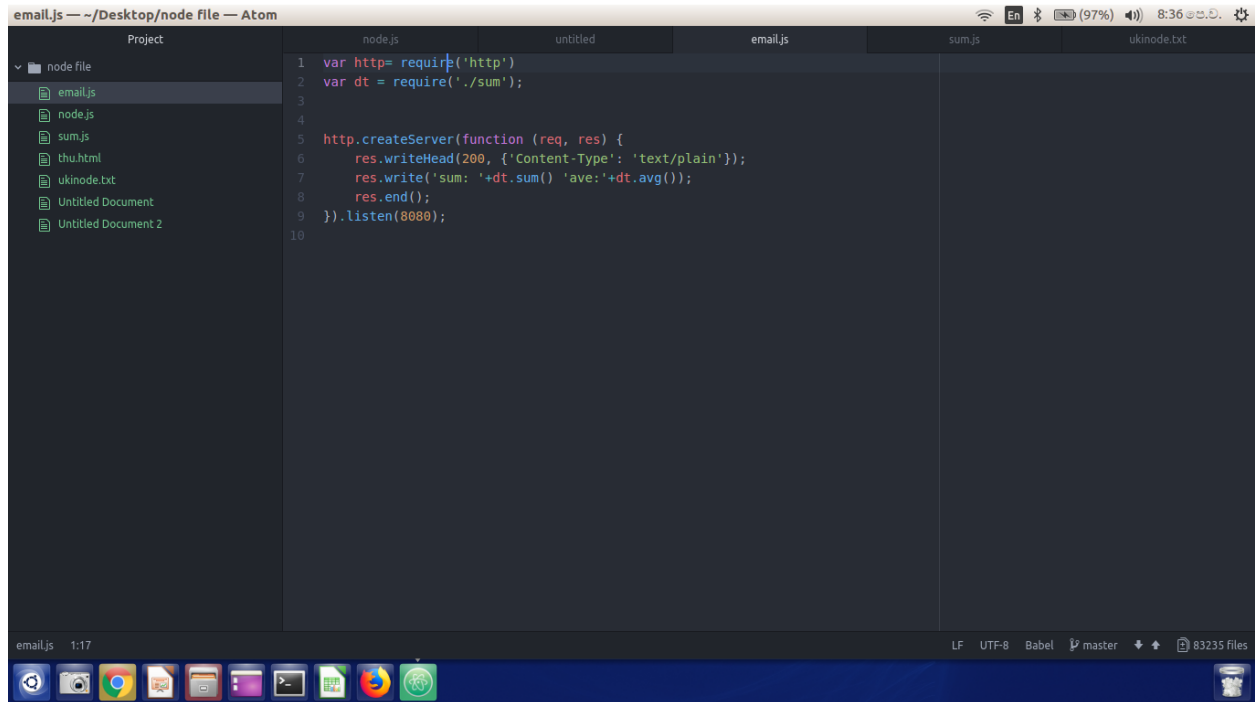


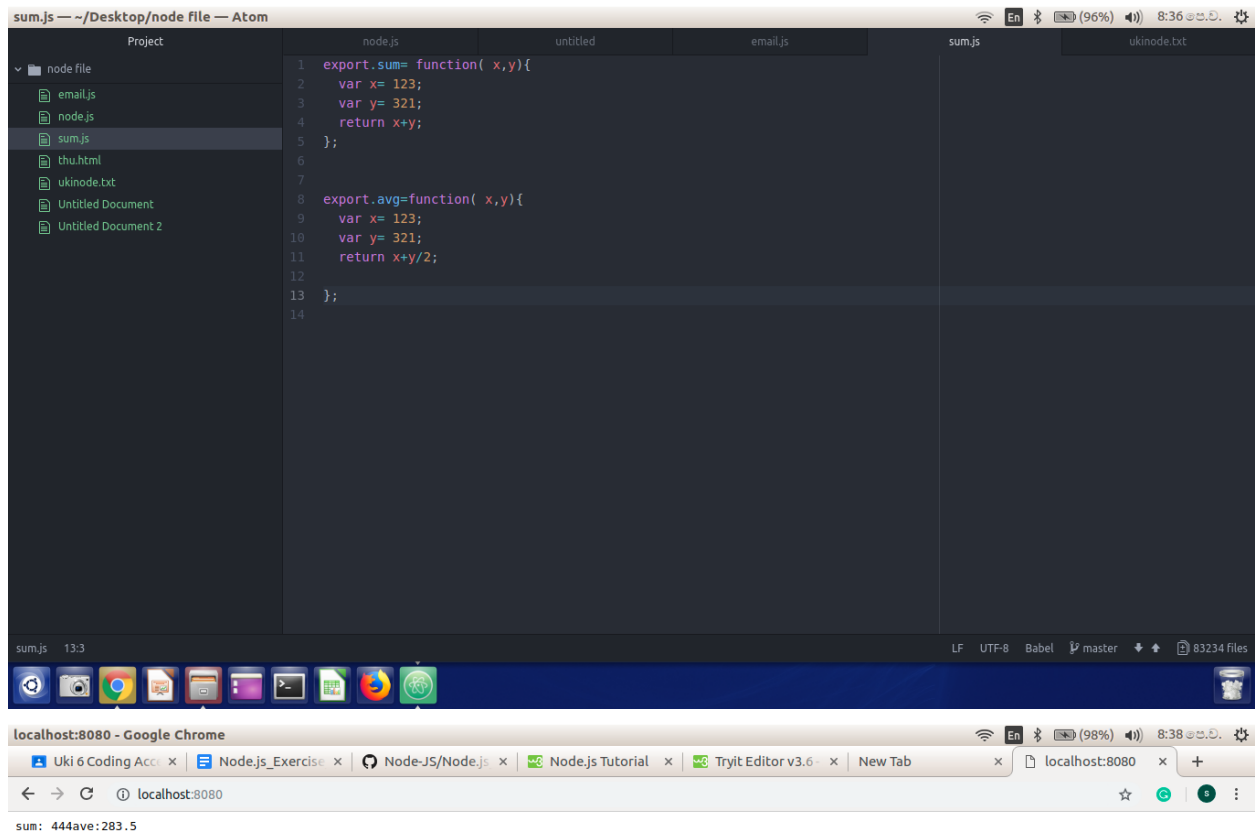
Node.js Exercise 1

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.



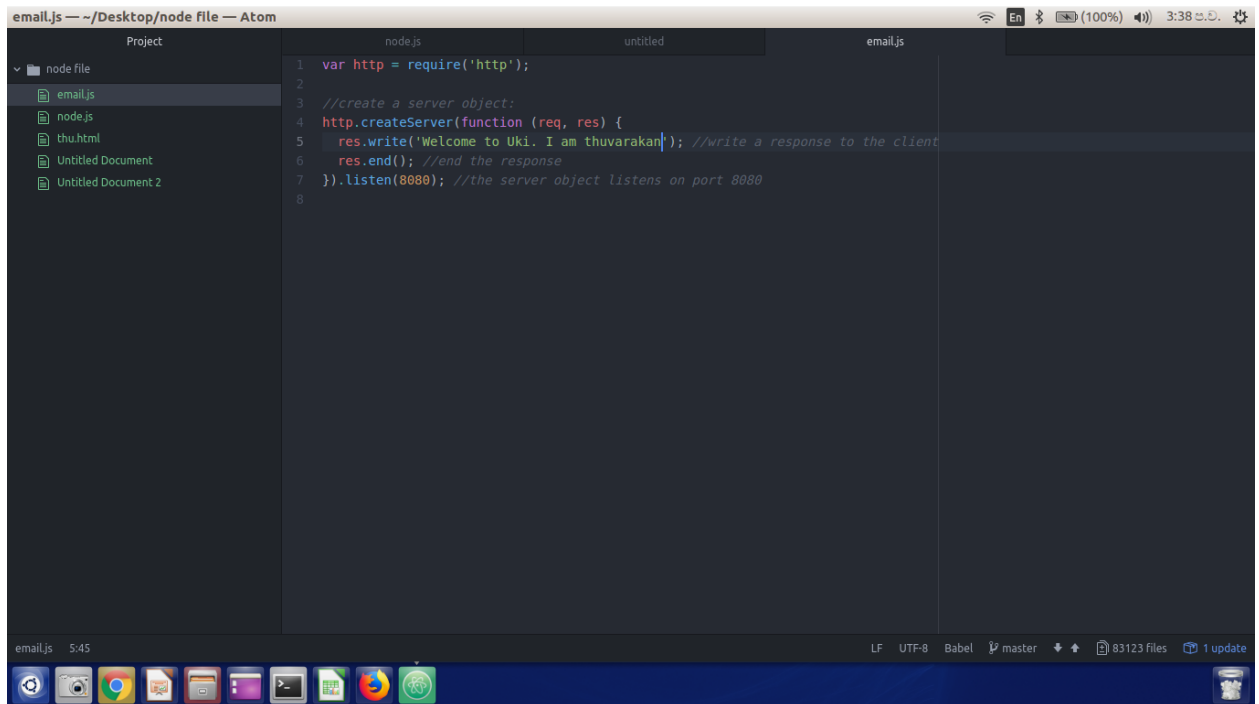
The screenshot shows the Atom text editor interface. The title bar indicates the file is 'email.js' located at '~/.Desktop/node file'. The left sidebar displays a project file explorer with a tree view containing: 'node file' (expanded), 'email.js', 'node.js', 'sum.js', 'thu.html', 'ukinode.txt', 'Untitled Document', and 'Untitled Document 2'. The main editor pane shows the code for 'email.js' with line numbers 1 through 10. The code imports the 'http' module and a local module 'dt' (located at './sum.js'). It then creates an HTTP server that listens on port 8080. The server's response headers are set to 'Content-Type: text/plain'. The response body is formatted as 'sum: ' + dt.sum() + 'ave: ' + dt.avg(). The status bar at the bottom shows the current file is 'email.js', the time is 1:17, and various settings like 'LF', 'UTF-8', 'Babel', and 'master' are visible. The system tray at the very bottom shows icons for various applications including a terminal, file explorer, and web browser.

```
1 var http= require('http')
2 var dt = require('./sum');
3
4
5 http.createServer(function (req, res) {
6   res.writeHead(200, {'Content-Type': 'text/plain'});
7   res.write('sum: '+dt.sum() 'ave: '+dt.avg());
8   res.end();
9 }).listen(8080);
10
```

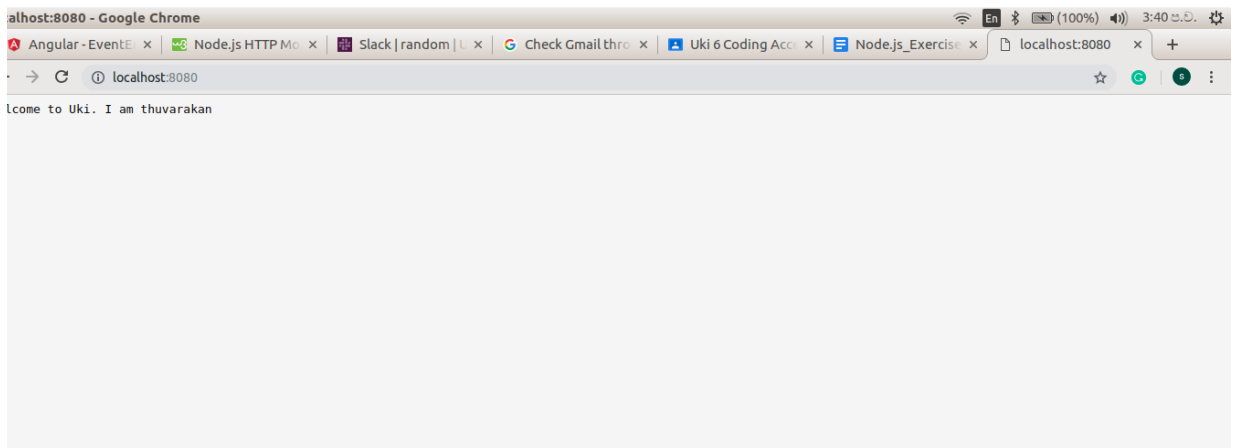


2.Create a simple http server and print “Welcome to Uki. I am **yourname**” when a request is sent to your server via the port 8000. (Note - Change different port numbers

and check)



```
1 var http = require('http');
2
3 //create a server object:
4 http.createServer(function (req, res) {
5   res.write('Welcome to Uki. I am thuvakaran'); //write a response to the client
6   res.end(); //end the response
7 }).listen(8080); //the server object listens on port 8080
8
```



3. Using the file system module create a new file called ukinode.txt
- 3.1 Write a paragraph about Uki into that file
- 3.2 Serve that file to the client (Read File) over your server
- 3.3 Append another paragraph about Uki and now serve the new file
- 3.4 Rename the file as ukinodejsexercise1.txt
- 3.5 Delete the file you created

email.js — ~/Desktop/node file — Atom

Project: node file

- email.js
- node.js
- thu.html
- Untitled Document
- Untitled Document 2

```
1 var fs = require('fs');
2
3 fs.appendFile('ukinode.txt ', 'welcome uki', function (err) {
4   if (err) throw err;
5   console.log('Saved!');
6 });
7
```

email.js 3:43

LF UTF-8 Babel master 83124 files 1 update

email.js — ~/Desktop/node file — Atom

Project: node file

- email.js
- node.js
- thu.html
- ukinode.txt
- Untitled Document
- Untitled Document 2

```
1 var fs = require('fs');
2
3 fs.appendFile('ukinode.txt ', 'welcome to uki', function (err) {
4   if (err) throw err;
5   console.log('update');
6 });
7
```

email.js 3:43

LF UTF-8 Babel master 83203 files 1 update

email.js — ~/Desktop/node file — Atom

Project: node file

- email.js
- node.js
- thu.html
- ukinode.txt
- Untitled Document
- Untitled Document 2

node.js

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

email.js

ukinode.txt

email.js 3:23

LF UTF-8 Babel master 83055 Files 1 update

email.js — ~/Desktop/node file — Atom

Project: node file

- email.js
- node.js
- thu.html
- ukinode.txt
- Untitled Document
- Untitled Document 2

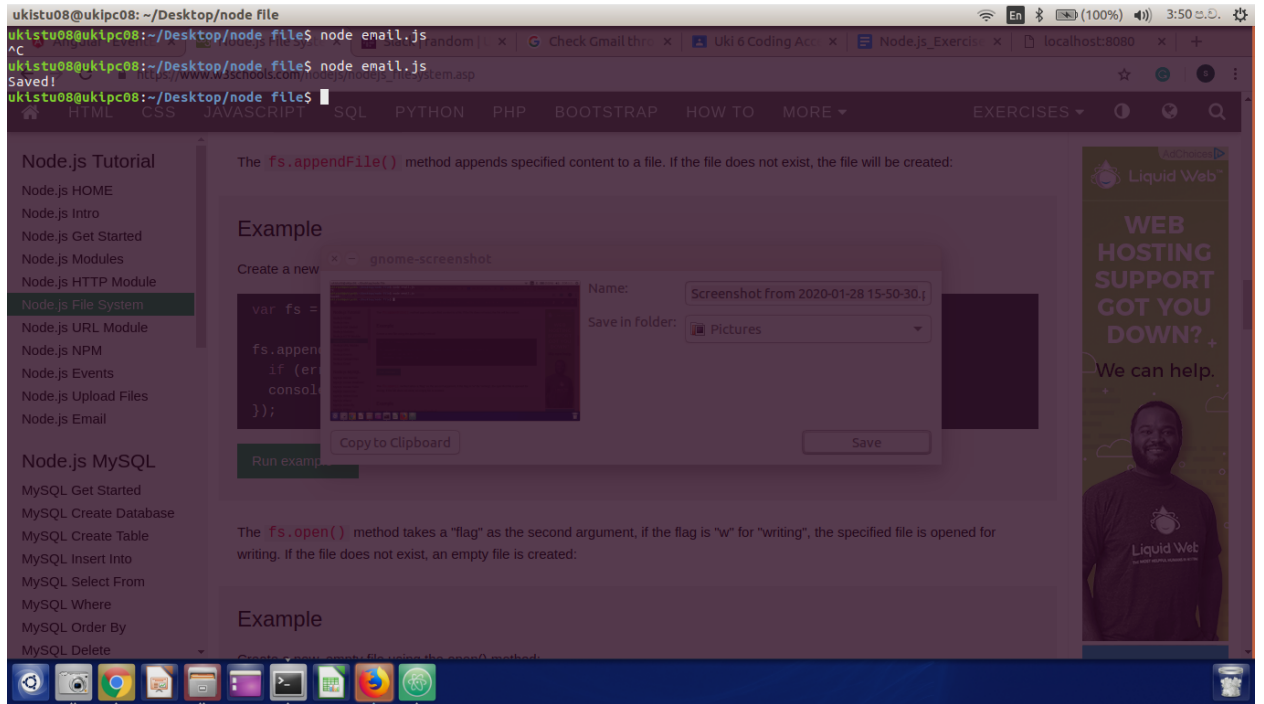
node.js

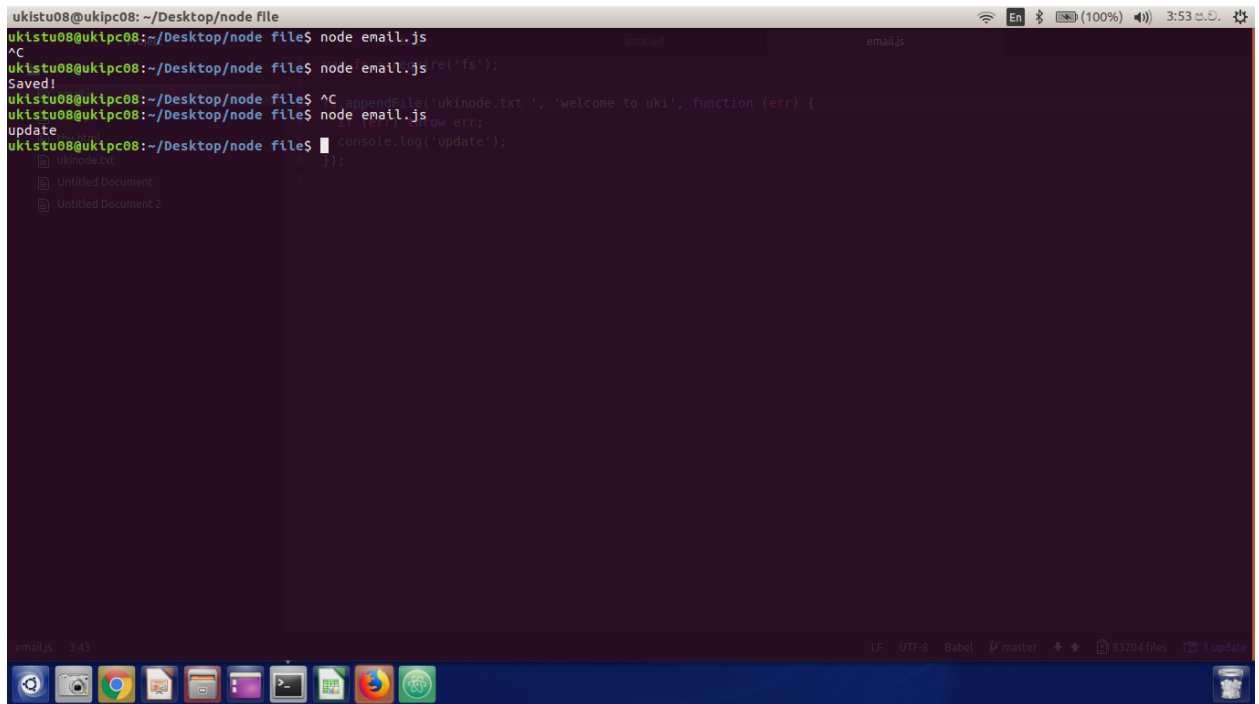
```
1 var fs = require('fs');
2
3 fs.rename('ukinode.txt', 'ukinodejsexercise1.txt', function (err) {
4   if (err) throw err;
5   console.log('File Renamed!');
6 });
7
```

email.js

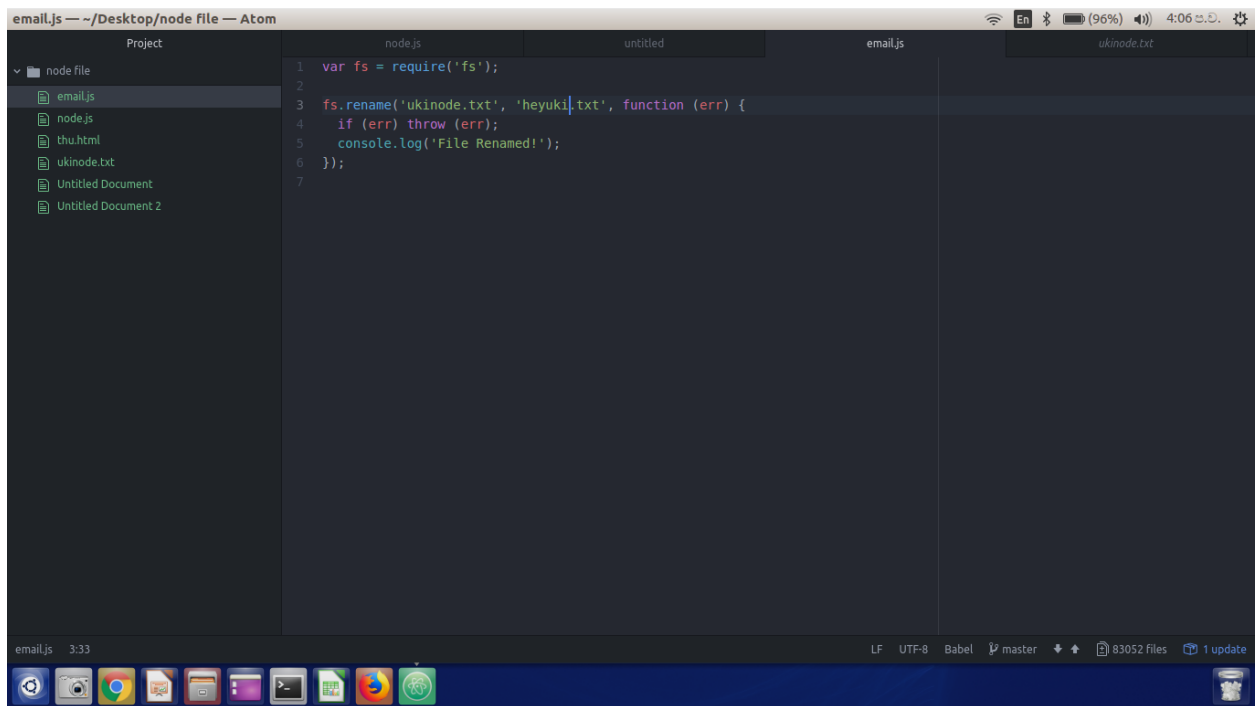
email.js 3:49

LF UTF-8 Babel master 83205 Files 1 update





```
ukistu08@ukipc08: ~/Desktop/node file
ukistu08@ukipc08:~/Desktop/node file$ node email.js
^C
ukistu08@ukipc08:~/Desktop/node file$ node email.js
Saved!
ukistu08@ukipc08:~/Desktop/node file$ node email.js
ukistu08@ukipc08:~/Desktop/node file$ node email.js
update
ukistu08@ukipc08:~/Desktop/node file$ console.log('update');
update
```



```
email.js — ~/Desktop/node file — Atom
Project: node file
  email.js
  node.js
  thu.html
  ukinode.txt
  Untitled Document
  Untitled Document 2

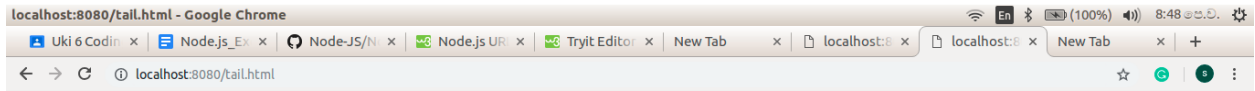
email.js
1 var fs = require('fs');
2
3 fs.rename('ukinode.txt', 'heyuki.txt', function (err) {
4   if (err) throw (err);
5   console.log('File Renamed!');
6 });
7
```

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.

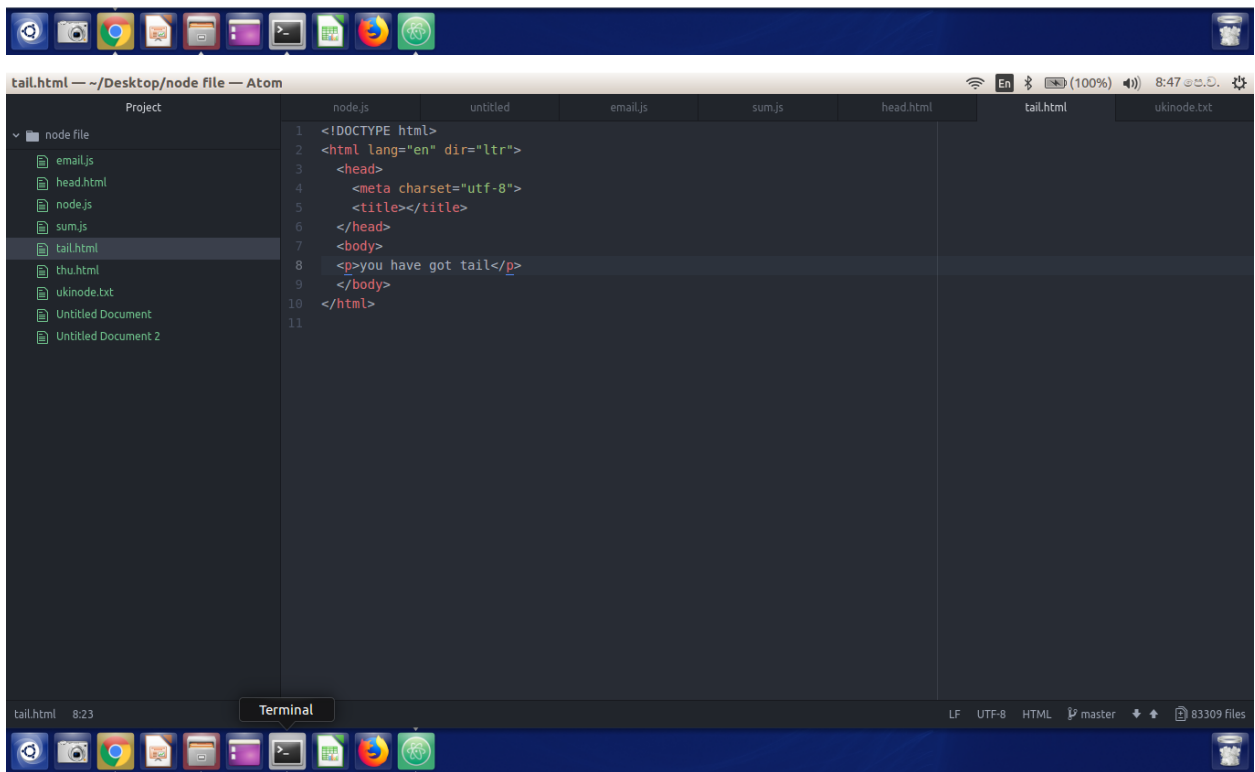
If you have followed the correct steps you should see two different results when opening these two addresses:

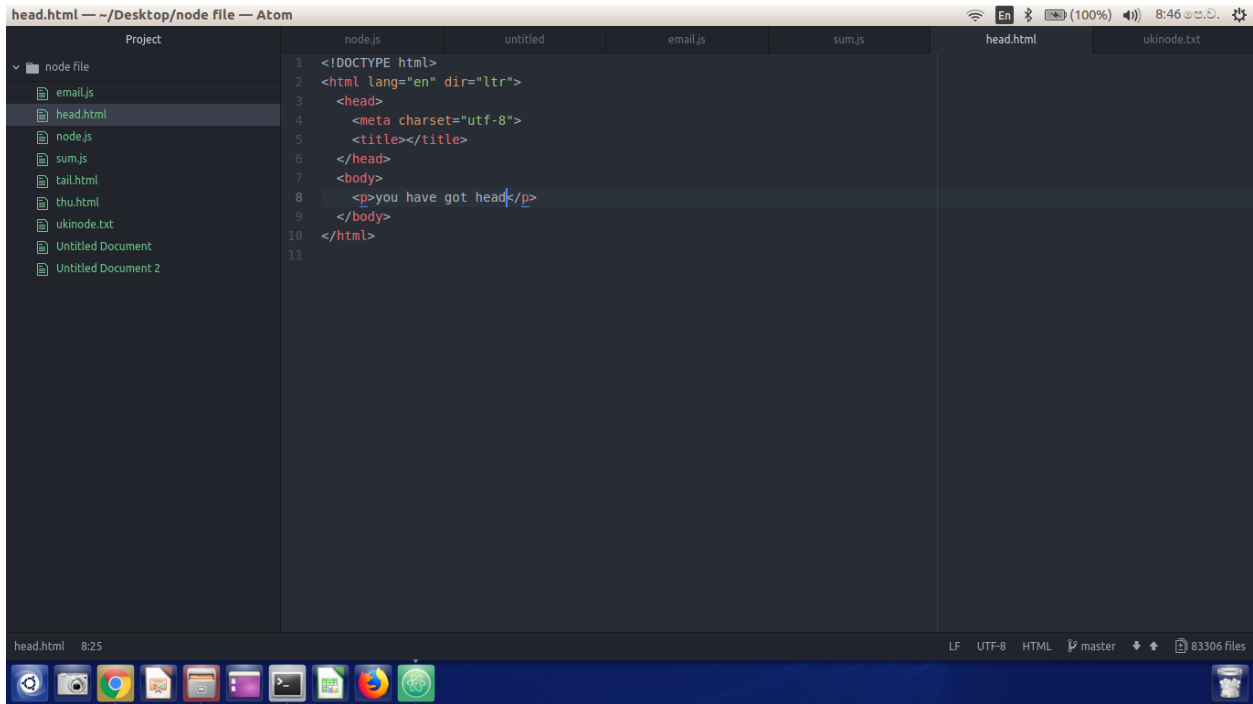
<http://localhost:8080/head.html> - > You have got head

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



you have got tail

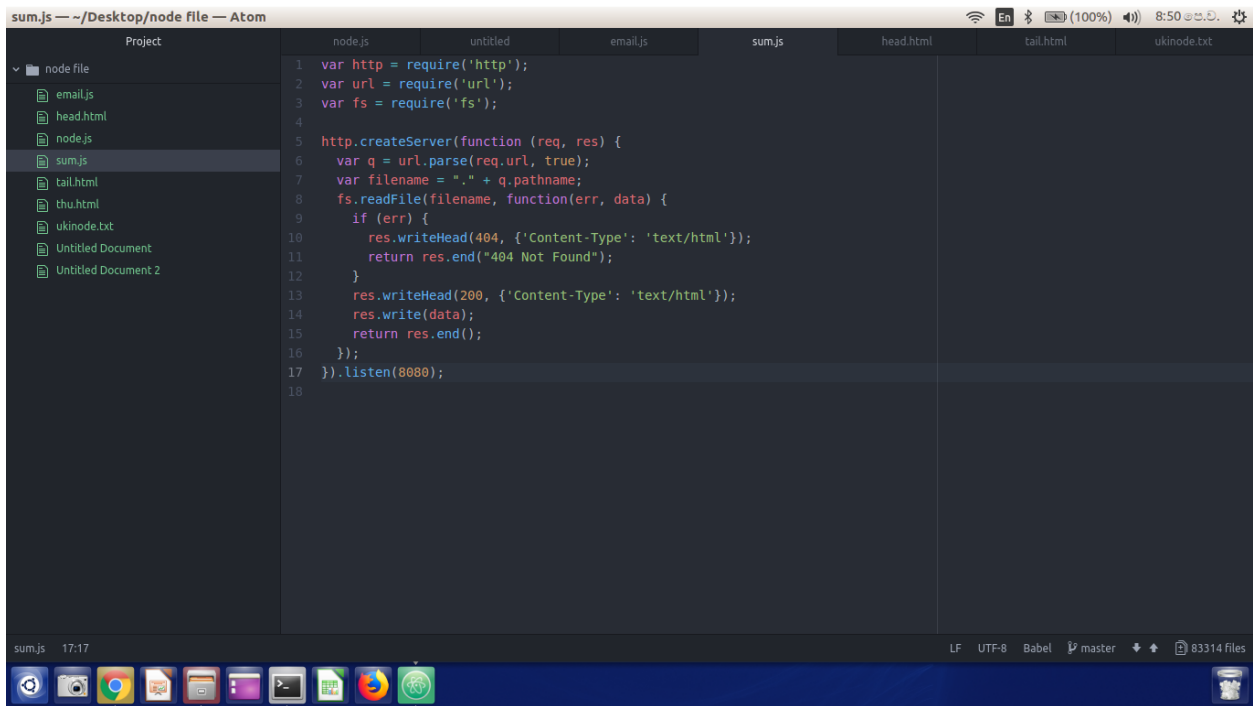




The screenshot shows the Atom editor interface with the file 'head.html' open. The left sidebar displays a project tree with files: node file, email.js, head.html, node.js, sum.js, tail.html, thu.html, ukinode.txt, and two untitled documents. The main editor area shows the following HTML code:

```
1 <!DOCTYPE html>
2 <html lang="en" dir="ltr">
3   <head>
4     <meta charset="utf-8">
5     <title></title>
6   </head>
7   <body>
8     <p>you have got head</p>
9   </body>
10 </html>
11
```

The status bar at the bottom indicates the file is 'head.html', the time is 8:25, and the encoding is UTF-8.

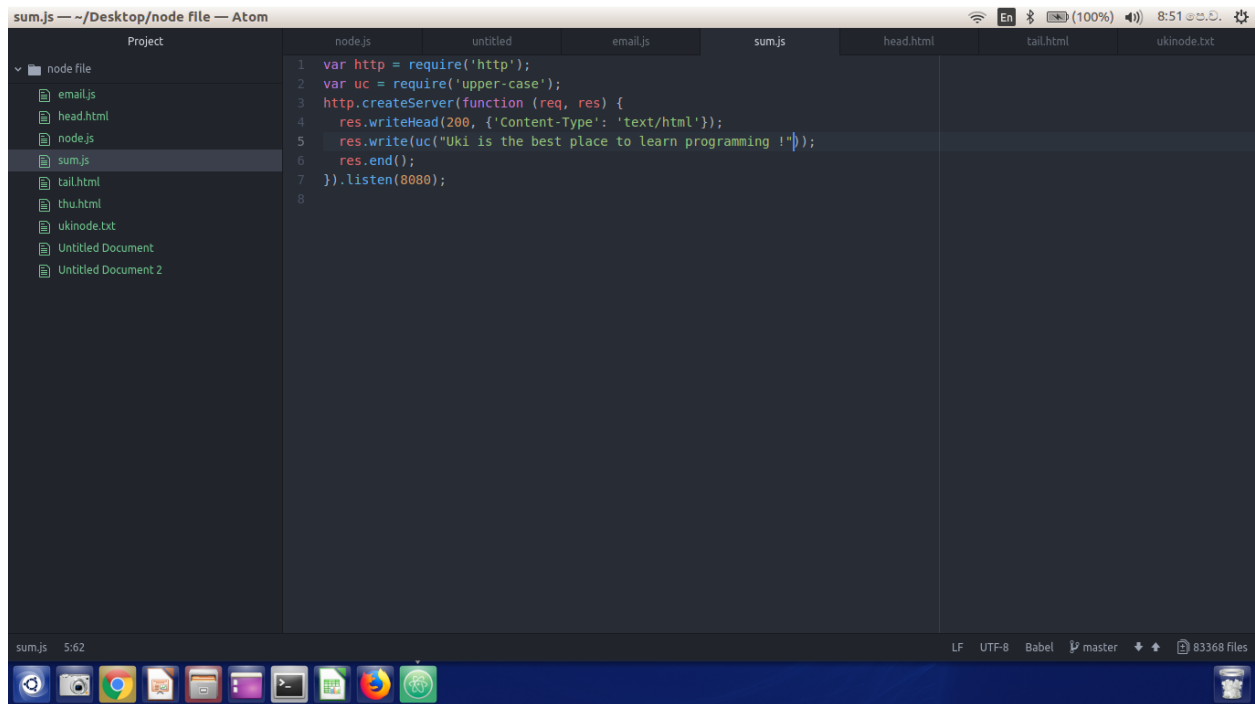


The screenshot shows the Atom editor interface with the file 'sum.js' open. The left sidebar displays a project tree with files: node file, email.js, head.html, node.js, sum.js, tail.html, thu.html, ukinode.txt, and two untitled documents. The main editor area shows the following JavaScript code:

```
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(data);
15    return res.end();
16  });
17 }).listen(8080);
18
```

The status bar at the bottom indicates the file is 'sum.js', the time is 17:17, and the encoding is UTF-8.

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.



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

email.js — ~/Desktop/node file — Atom

Project

- node file
 - email.js
 - head.html
 - node.js
 - sum.js
 - tail.html
 - thuh.html
 - ukinode.txt
 - Untitled Document
 - Untitled Document 2

```
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
```

email.js 6:46

LF UTF-8 Babel master 83407 files

ukistu08@ukipc08: ~/Desktop/node file

```
ukistu08@ukipc08:~/Desktop/node file$ node email.js
I bark when I see strangers !
ukistu08@ukipc08:~/Desktop/node file$
```

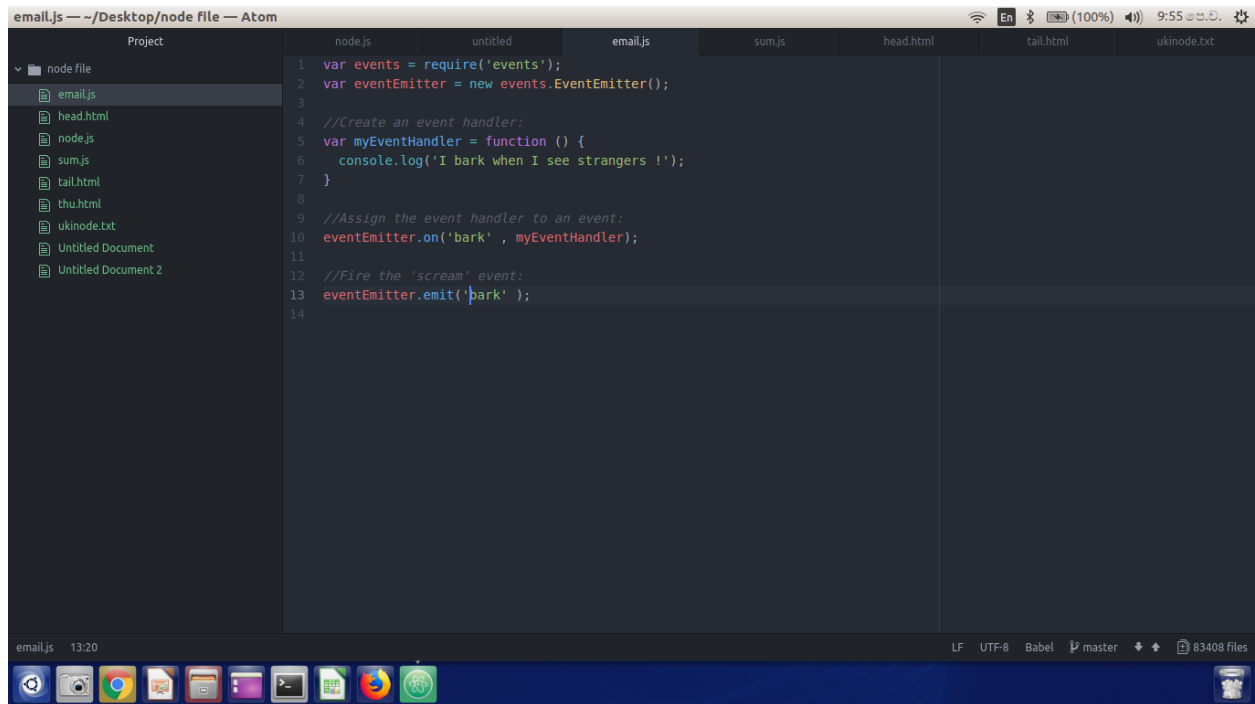
Project

- node file
 - email.js
 - head.html
 - node.js
 - sum.js
 - tail.html
 - thuh.html
 - ukinode.txt
 - Untitled Document
 - Untitled Document 2

```
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
```

email.js 13:20

LF UTF-8 Babel master 83409 files



```
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
```

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.

email.js — ~/Desktop/node file — Atom

Project: node file

- email.js
- node.js
- thu.html
- Untitled Document
- Untitled Document 2

```
1 var nodemailer = require('nodemailer');
2
3 var transporter = nodemailer.createTransport({
4   service: 'gmail',
5   auth: {
6     user: 'pullinkouki6@gmail.com',
7     pass: '@ukipulli6'
8   }
9 });
10
11 var mailOptions = {
12   from: 'pullinkouki6@gmail.com',
13   to: 'info@uki.life',
14   subject: 'Testing my nodemailer module',
15   text: 'This is 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
```

email.js 12:32 LF UTF-8 Babel master 82950 files 1 update

ukistu08@ukipc08: ~/Desktop/node file

```
ukistu08@ukipc08:~/Desktop/node file$ node email.js
Email sent: 250 2.0.0 OK 1580205214 g7sm19122794pfq.33 - gsmt
ukistu08@ukipc08:~/Desktop/node file$
```

Project: node file

- email.js
- node.js
- thu.html
- Untitled Document
- Untitled Document 2

```
1 var nodemailer = require('nodemailer');
2
3 var transporter = nodemailer.createTransport({
4   service: 'gmail',
5   auth: {
6     user: 'pullinkouki6@gmail.com',
7     pass: '@ukipulli6'
8   }
9 });
10
11 var mailOptions = {
12   from: 'pullinkouki6@gmail.com',
13   to: 'info@uki.life',
14   subject: 'Testing my nodemailer module',
15   text: 'This is 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
```

email.js 12:32 LF UTF-8 Babel master 82948 files 1 update