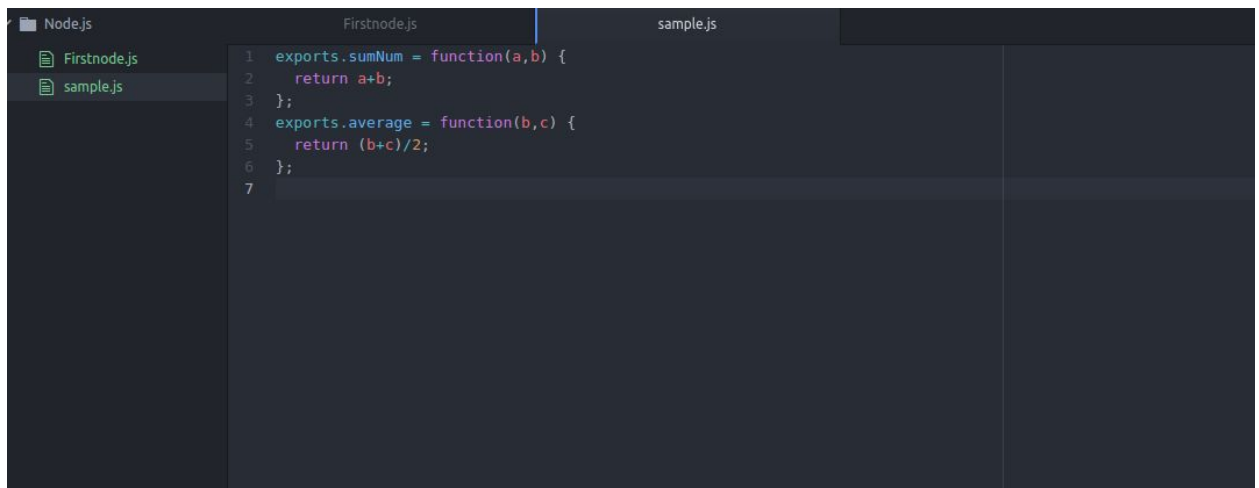
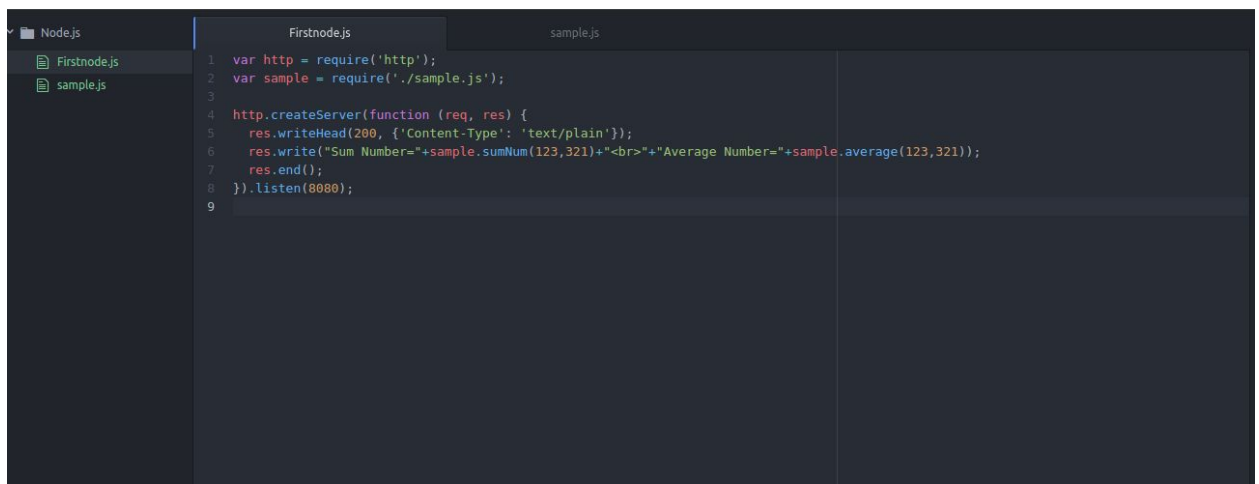
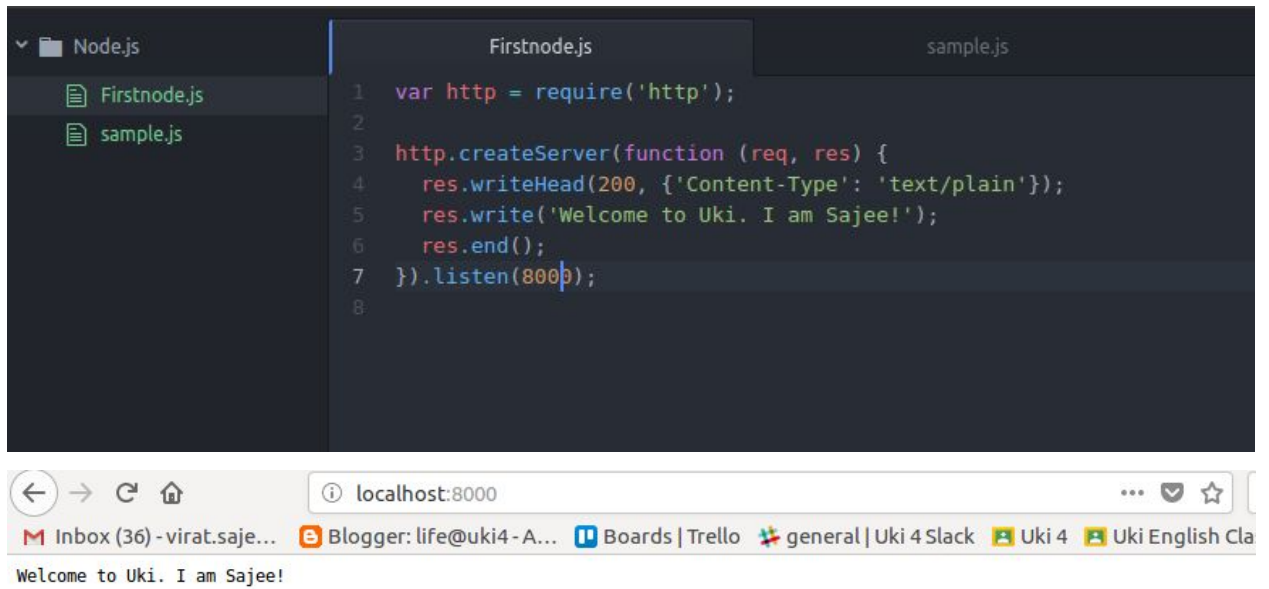


1. Create a custom module which returns the sum and average of any two numbers passed into it. Require the module and run the server by passing 123 and 321 so that the server prints out the sum and average.



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)

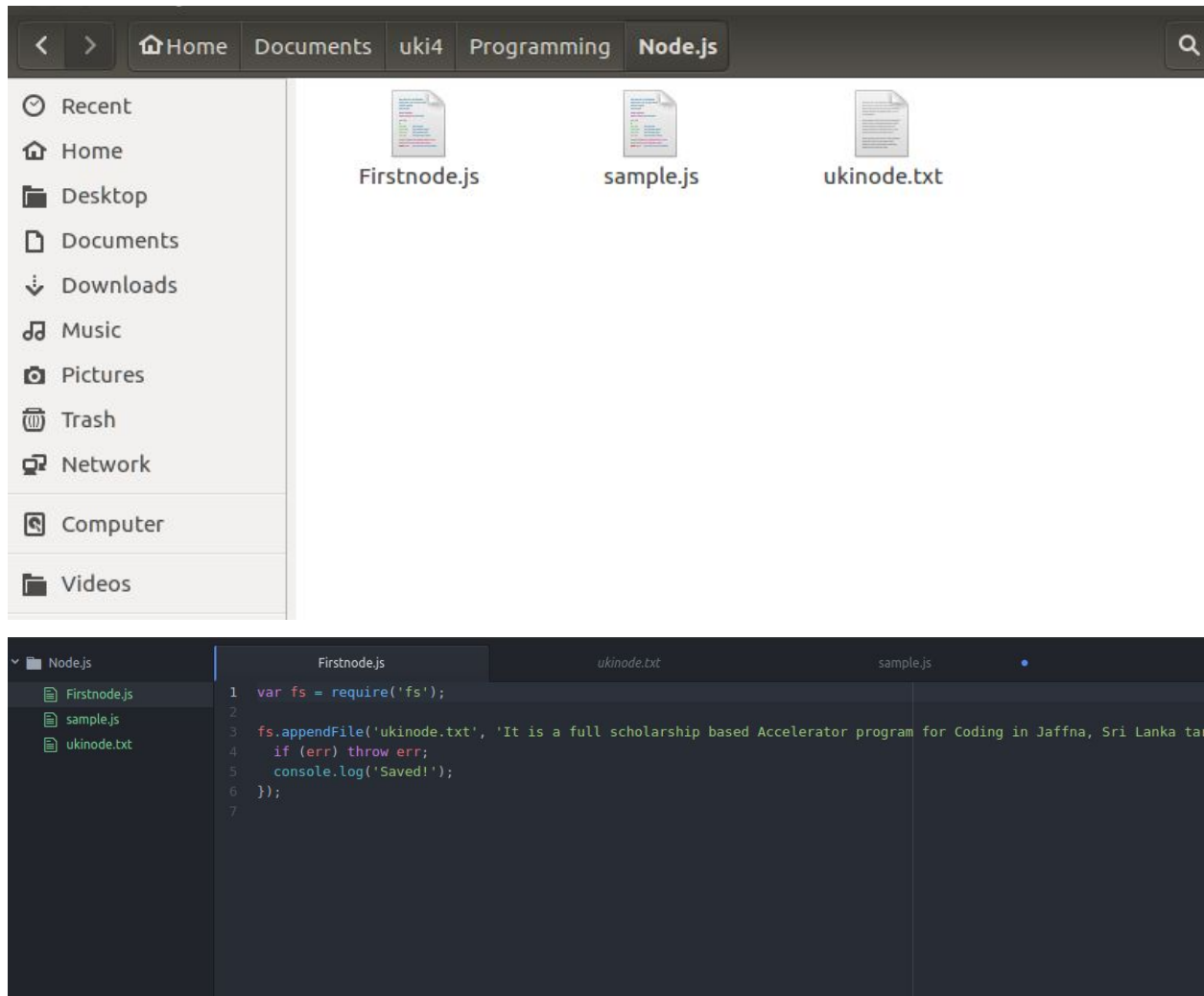


The image shows a code editor with a file explorer on the left displaying a folder named 'Node.js' containing two files: 'Firstnode.js' and 'sample.js'. The 'Firstnode.js' file is open in the editor, showing the following JavaScript code:

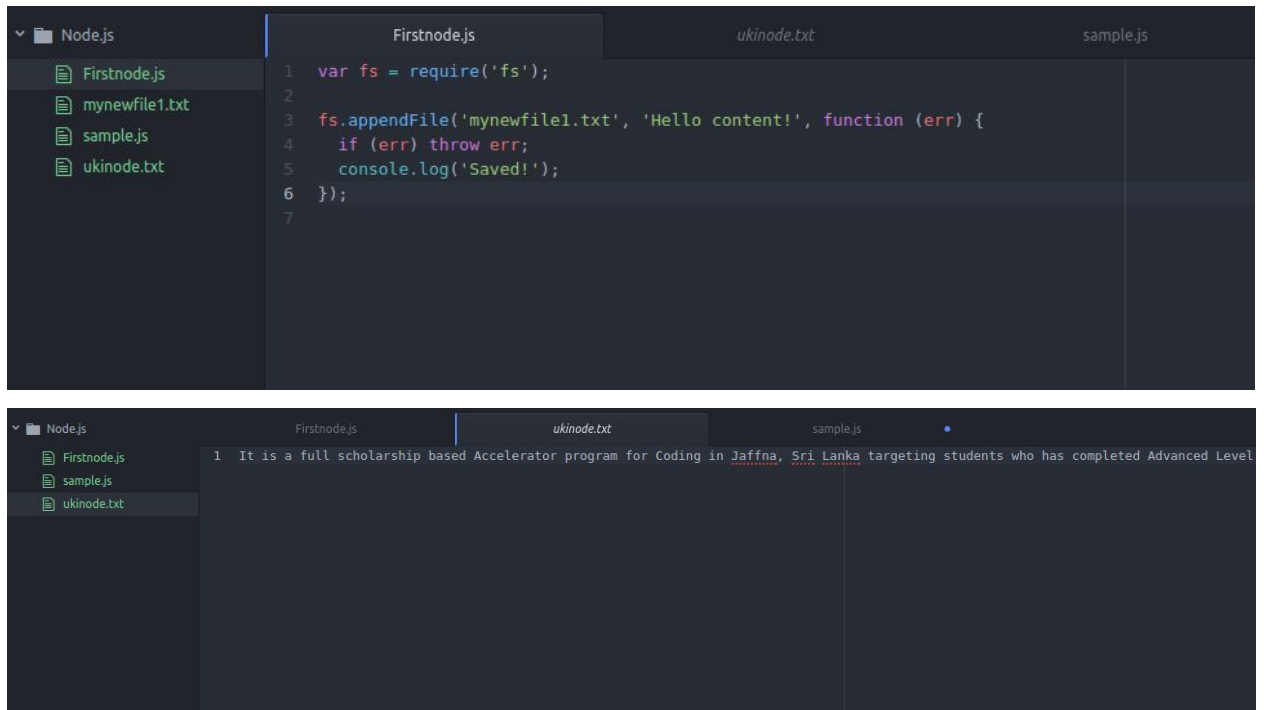
```
1 var http = require('http');
2
3 http.createServer(function (req, res) {
4   res.writeHead(200, {'Content-Type': 'text/plain'});
5   res.write('Welcome to Uki. I am Sajee!');
6   res.end();
7 }).listen(8000);
8
```

Below the code editor, a web browser window is open to the address 'localhost:8000'. The browser's address bar shows the URL, and the page content displays the message: 'Welcome to Uki. I am Sajee!'.

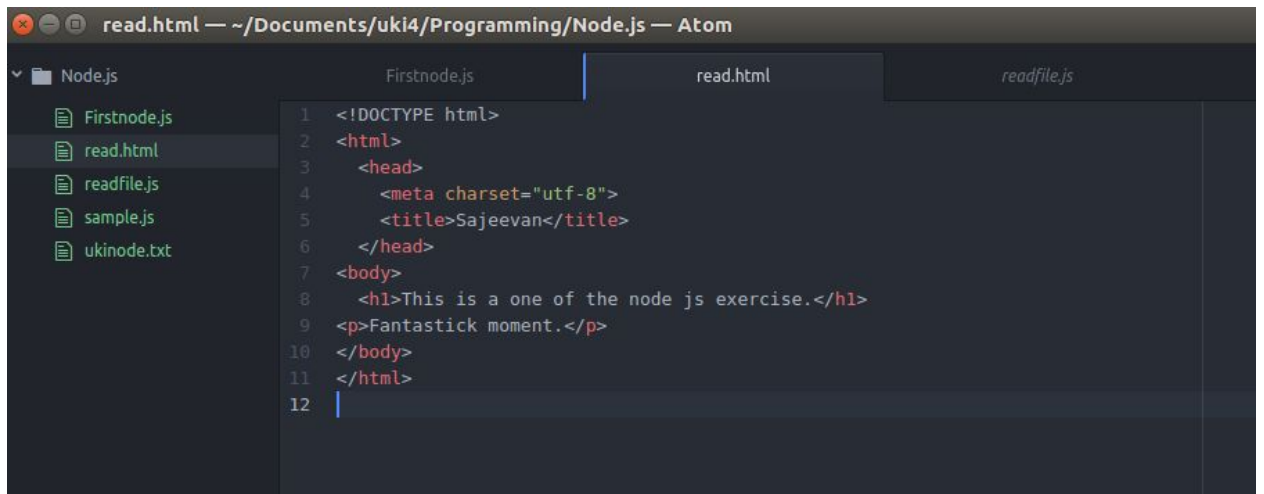
3. Using the file system module create a new file called ukinode.txt :




3.1 Write a paragraph about Uki into that file:



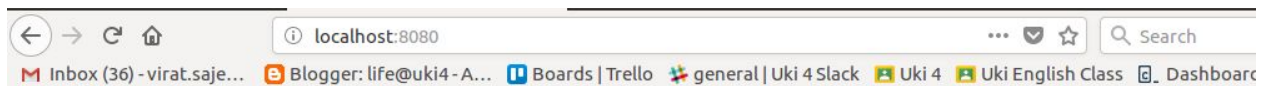
3.2 Serve that file to the client (Read File) over your server:

A screenshot of the Atom text editor. The title bar shows 'read.html — ~/Documents/uki4/Programming/Node.js — Atom'. The left sidebar shows a file explorer with 'Node.js' expanded, listing 'Firstnode.js', 'read.html', 'readfile.js', 'sample.js', and 'ukinode.txt'. The main editor area shows the content of 'read.html' with line numbers 1 through 12. The code is an HTML document with a title 'Sajeevan' and a body containing an h1 and a p tag.

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <title>Sajeevan</title>
6   </head>
7   <body>
8     <h1>This is a one of the node js exercise.</h1>
9     <p>Fantastick moment.</p>
10  </body>
11 </html>
12
```

A screenshot of the Atom text editor. The title bar shows 'readfile.js — ~/Documents/uki4/Programming/Node.js — Atom'. The left sidebar shows the same file explorer as the previous screenshot. The main editor area shows the content of 'readfile.js' with line numbers 1 through 10. The code is a JavaScript file that uses the http and fs modules to create a server that serves the 'read.html' file.

```
1 var http = require('http');
2 var fs = require('fs');
3 http.createServer(function (req, res) {
4   fs.readFile('read.html', function(err, data) {
5     res.writeHead(200, {'Content-Type': 'text/html'});
6     res.write(data);
7     res.end();
8   });
9 }).listen(8080);
10
```



This is a one of the node js exercise.

Fantastick moment.

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

```
ukistu02@ukipc02: ~/Documents/uki4/Programming/Node.js
[nodemon] to restart at any time, enter `rs`
[nodemon] watching: *.*
[nodemon] starting `node readfile.js`
[nodemon] restarting due to changes...
[nodemon] starting `node readfile.js`
[nodemon] restarting due to changes...
[nodemon] starting `node readfile.js`
[nodemon] restarting due to changes...
[nodemon] starting `node readfile.js`
^C

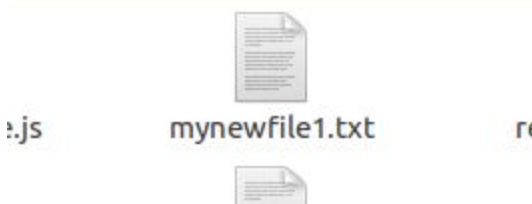
Update available: 1.18.9 (current: 1.11.0)
Run npm install -g nodemon to update.

ukistu02@ukipc02:~/Documents/uki4/Programming/Node.js$ nodemon append.js
[nodemon] 1.11.0
[nodemon] to restart at any time, enter `rs`
[nodemon] watching: *.*
[nodemon] starting `node append.js`
Saved!
[nodemon] clean exit - waiting for changes before restart
```

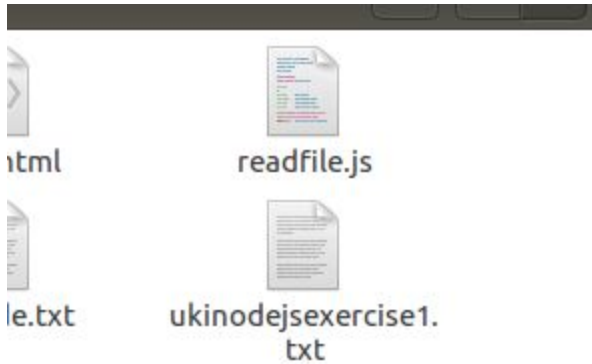
```
append.js — ~/Documents/uki4/Programming/Node.js — Atom

Node.js
  append.js
  Firstnode.js
  mynewfile1.txt
  read.html
  readfile.js
  sample.js
  ukinode.txt

1  var fs = require('fs');
2
3  fs.appendFile('mynewfile1.txt', 'Hello content!', function (err) {
4    if (err) throw err;
5    console.log('Saved!');
6  });
7
```



3.4 Rename the file as ukinodejsexercise1.txt:



```
FirstNode.js  read.html  append.js  rename.js

1  var fs = require('fs');
2
3  fs.rename('mynewfile1.txt', 'ukinodejsexercise1.txt', function (err) {
4    if (err) throw err;
5    console.log('File Renamed!');
6  });
7
```

```
nodemon] clean exit - waiting for changes before restart
Cukistu02@ukipc02:~/Documents/uki4/Programming/Node.js$ nodemon rename.js
nodemon] 1.11.0
nodemon] to restart at any time, enter `rs`
nodemon] watching: *.*
nodemon] starting `node rename.js`
File Renamed!
nodemon] clean exit - waiting for changes before restart
```

3.5 Delete the file you created:

```
ukistu02@uklpc02: ~/Documents/uki4/Programming/Node.js
ukistu02@uklpc02:~/Documents/uki4/Programming/Node.js$ nodemon unlink.js
[nodemon] 1.11.0
[nodemon] to restart at any time, enter `rs`
[nodemon] watching: *.*
[nodemon] starting `node unlink.js`
File deleted!
[nodemon] clean exit - waiting for changes before restart
```

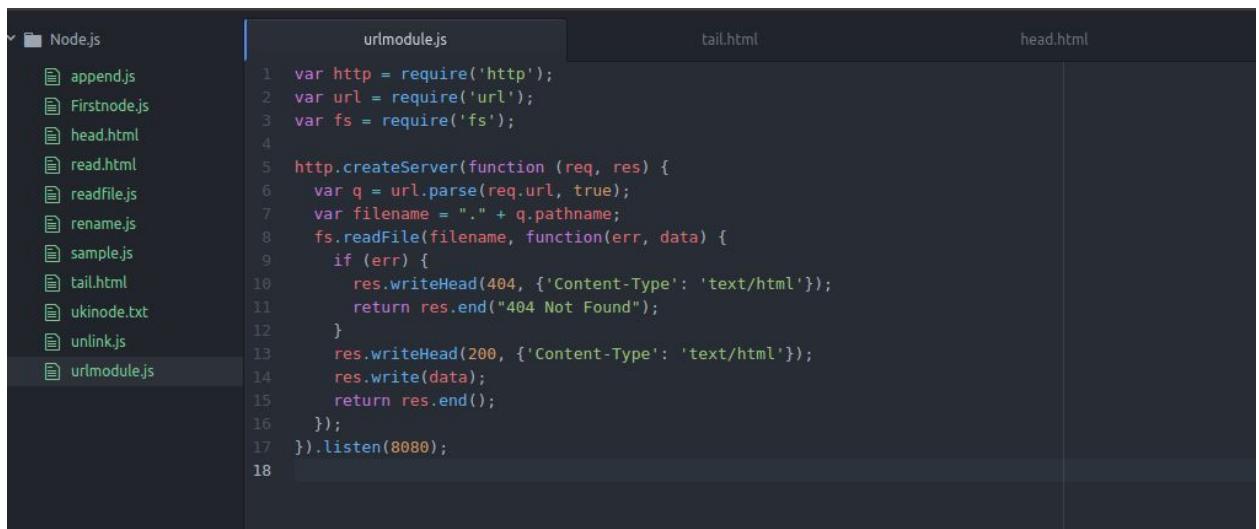
```
Firstnode.js  readfile.js  unlink.js
1  var fs = require('fs');
2
3  fs.unlink('ukinodejsexercise1.txt', function (err) {
4    if (err) throw err;
5    console.log('File deleted!');
6  });
7
```


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.

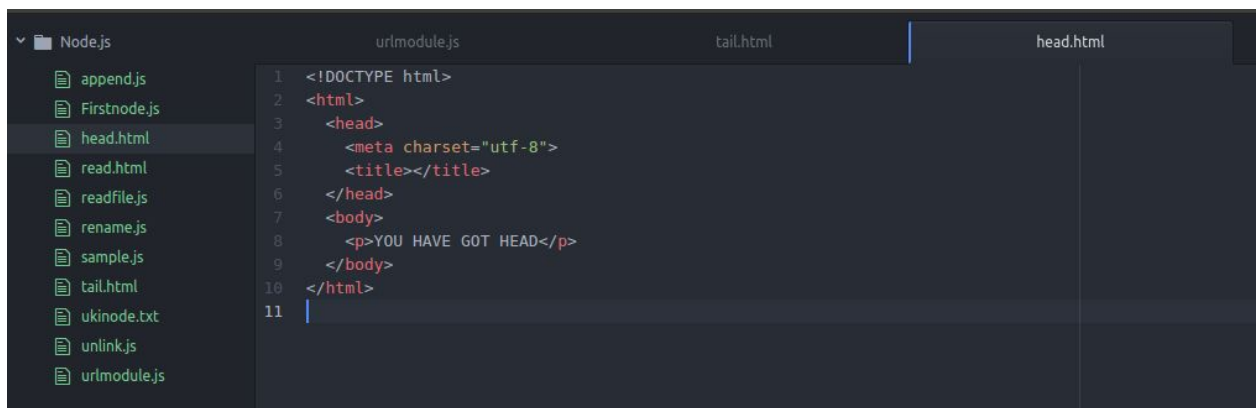
If you have followed the correct steps you should see two different results when opening these two addresses:

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

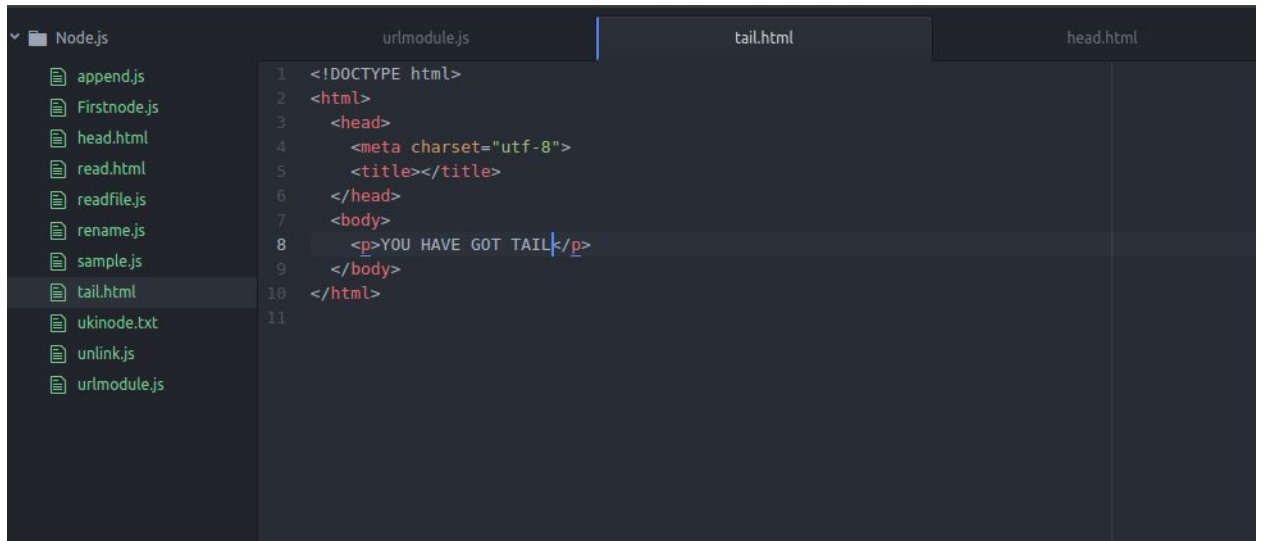
<http://localhost:8080/tail.html> -> You have got tail



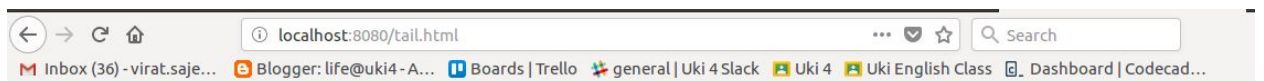
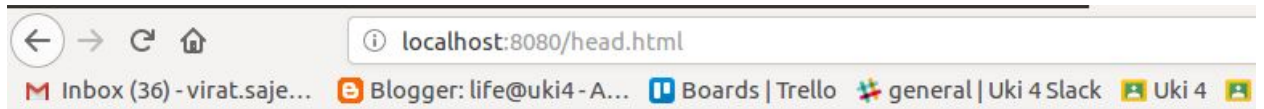
```
1 var http = require('http');
2 var url = require('url');
3 var fs = require('fs');
4
5 http.createServer(function (req, res) {
6   var q = url.parse(req.url, true);
7   var filename = "." + q.pathname;
8   fs.readFile(filename, function(err, data) {
9     if (err) {
10       res.writeHead(404, {'Content-Type': '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
```



```
1 <!DOCTYPE html>
2 <html>
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
```



```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta charset="utf-8">
5   <title></title>
6 </head>
7 <body>
8   <p>YOU HAVE GOT TAIL</p>
9 </body>
10 </html>
11
```



```
ukistu02@ukipc02: ~/Documents/uki4/Programming/Node.js
ukistu02@ukipc02:~/Documents/uki4/Programming/Node.js$ nodemon urlmodule.js
[nodemon] 1.11.0
[nodemon] to restart at any time, enter `rs`
[nodemon] watching: *.*
[nodemon] starting `node urlmodule.js`
[nodemon] restarting due to changes...
[nodemon] restarting due to changes...
[nodemon] starting `node urlmodule.js`
[nodemon] restarting due to changes...
[nodemon] starting `node urlmodule.js`
```

5. Install the package “upper-case” using NPM and create a Node.js file that will convert the output " into upper-case letters.?

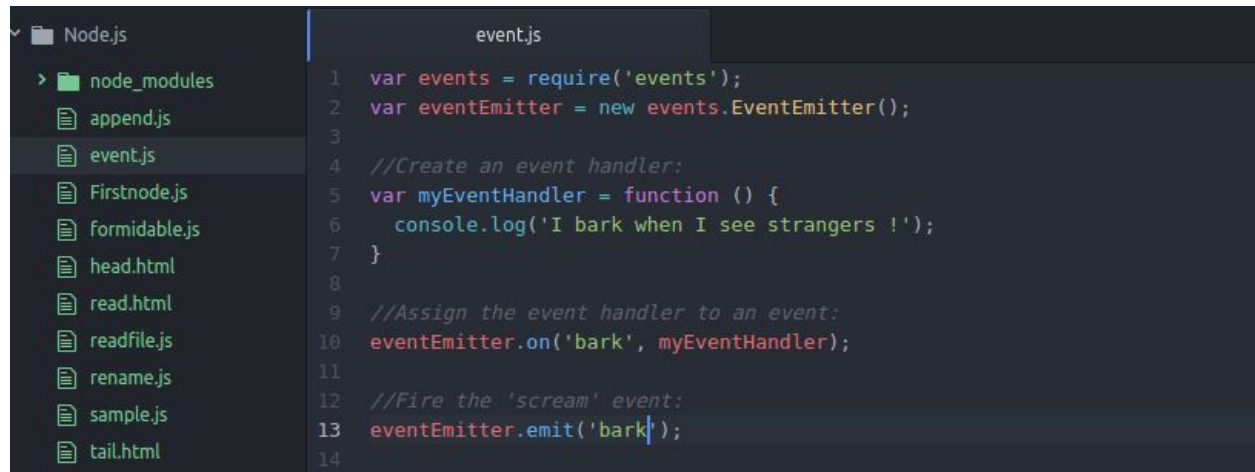
```
Node.js
├── node_modules
│   ├── append.js
│   ├── Firstnode.js
│   ├── head.html
│   ├── read.html
│   ├── readfile.js
│   └── rename.js
└── uppercase.js

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   res.write(uc("Uki is the best place to learn programming !"));
6   res.end();
7 }).listen(8080);
8

ukistu02@ukipc02:~/Documents/uki4/Programming/Node.js$ nodemon uppercase.js
[nodemon] 1.11.0
[nodemon] to restart at any time, enter `rs`
[nodemon] watching: *.*
[nodemon] starting `node uppercase.js`
```

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

```
[nodemon] starting `node formidable.js`  
^Cukistu02@ukipc02:~/Documents/uki4/Programming/Node.js$ nodemon event.js  
[nodemon] 1.11.0  
[nodemon] to restart at any time, enter `rs`  
[nodemon] watching: *.*  
[nodemon] starting `node event.js`  
I bark when I see strangers !  
[nodemon] clean exit - waiting for changes before restart
```



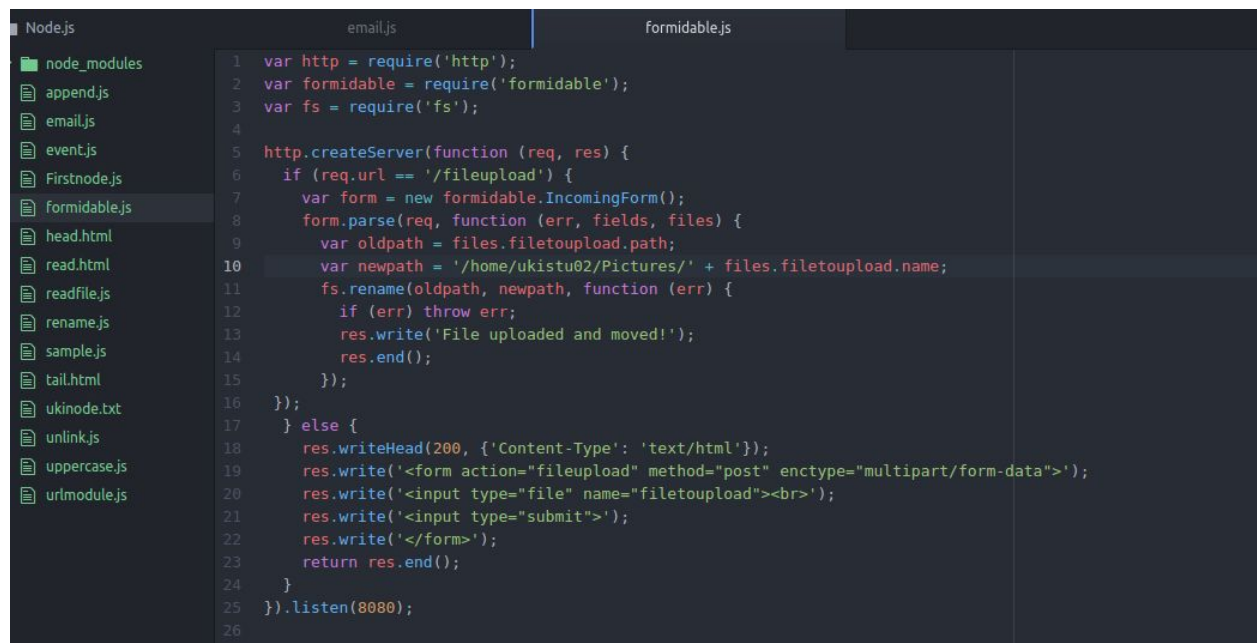
```
1 var events = require('events');  
2 var EventEmitter = new events.EventEmitter();  
3  
4 //Create an event handler:  
5 var myEventHandler = function () {  
6   console.log('I bark when I see strangers !');  
7 }  
8  
9 //Assign the event handler to an event:  
10 EventEmitter.on('bark', myEventHandler);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

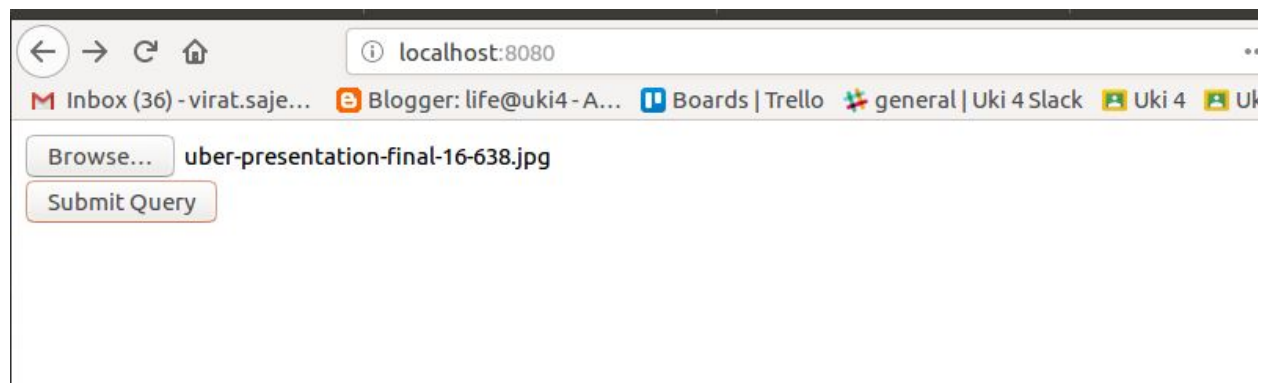
7.1 Save that uploaded file into your Documents directory.?

```
ukistu02@ukipc02:~/Documents/uki4/Programming/Node.js$ nodemon formidable.js
[nodemon] 1.11.0
[nodemon] to restart at any time, enter `rs`
[nodemon] watching: *.*
[nodemon] starting `node formidable.js`
```

File uploaded and moved!



```
1 var http = require('http');
2 var formidable = require('formidable');
3 var fs = require('fs');
4
5 http.createServer(function (req, res) {
6   if (req.url == '/fileupload') {
7     var form = new formidable.IncomingForm();
8     form.parse(req, function (err, fields, files) {
9       var oldpath = files.fileupload.path;
10      var newpath = '/home/ukistu02/Pictures/' + files.fileupload.name;
11      fs.rename(oldpath, newpath, function (err) {
12        if (err) throw err;
13        res.write('File uploaded and moved!');
14        res.end();
15      });
16    });
17   } else {
18     res.writeHead(200, {'Content-Type': 'text/html'});
19     res.write('<form action="/fileupload" method="post" enctype="multipart/form-data">');
20     res.write('<input type="file" name="fileupload"><br>');
21     res.write('<input type="submit">');
22     res.write('</form>');
23     return res.end();
24   }
25 }).listen(8080);
26
```



localhost:8080

Inbox (36) - virat.saje... Blogger: life@uki4-A... Boards | Trello general | Uki 4 Slack Uki 4 Uki 4

Browse... uber-presentation-final-16-638.jpg

Submit Query

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 !” ?

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

```
ukistu02@ukipc02: ~/Documents/uki4/Programming/Node.js
  at IncomingMessage.<anonymous> (/home/ukistu02/Documents/uki4/Programming/Node.js/node_modules/formidable/lib/incoming_form.js:134:14)
  at emitNone (events.js:86:13)
  at IncomingMessage.emit (events.js:185:7)
  at endReadableNT (_stream_readable.js:974:12)
  at _combinedTickCallback (internal/process/next_tick.js:80:11)
[nodemon] app crashed - waiting for file changes before starting...
^Cukistu02@ukipc02:~/Documents/uki4/Programming/Node.js$ nodemon formidable.js
[nodemon] 1.11.0
[nodemon] to restart at any time, enter `rs`
[nodemon] watching: *.*
[nodemon] starting `node formidable.js`
^Cukistu02@ukipc02:~/Documents/uki4/Programming/Node.js$ nodemon email.js
[nodemon] 1.11.0
[nodemon] to restart at any time, enter `rs`
[nodemon] watching: *.*
[nodemon] starting `node email.js`
{ Error: Invalid login: 535-5.7.8 Username and Password not accepted. Learn more at
535 5.7.8 https://support.google.com/mail/?p=BadCredentials y5sm8059087ioj.9 -
gsmtip
  at SMTPConnection._formatError (/home/ukistu02/Documents/uki4/Programming/Node.js/node_modules/nodemailer/lib/smtp-connection/index.js:774:19)
  at SMTPConnection._actionAUTHComplete (/home/ukistu02/Documents/uki4/Program
```

```
Node.js
  node_modules
  append.js
  email.js
  event.js
  Firstnode.js
  formidable.js
  head.html
  read.html
  readfile.js
  rename.js
  sample.js
  tail.html
  ukinode.txt
  unlink.js
  uppercase.js
  urlmodule.js

email.js
1  var nodemailer = require('nodemailer');
2
3  var transporter = nodemailer.createTransport({
4    service: 'gmail',
5    auth: {
6      user: 'virat.sajee1998@gmail.com',
7      pass: 'solla'
8    }
9  });
10
11 var mailOptions = {
12   from: 'virat.sajee1998@gmail.com',
13   to: 'info@uki.life',
14   subject: 'Testing my nodemailer module',
15   text: 'This is easy !',
16   html: '<h1>Welcome</h1><p>That was easy!</p>'
17 };
18
19 transporter.sendMail(mailOptions, function(error, info){
20   if (error) {
21     console.log(error);
22   } else {
23     console.log('Email sent: ' + info.response);
24   }
25 });
26
```