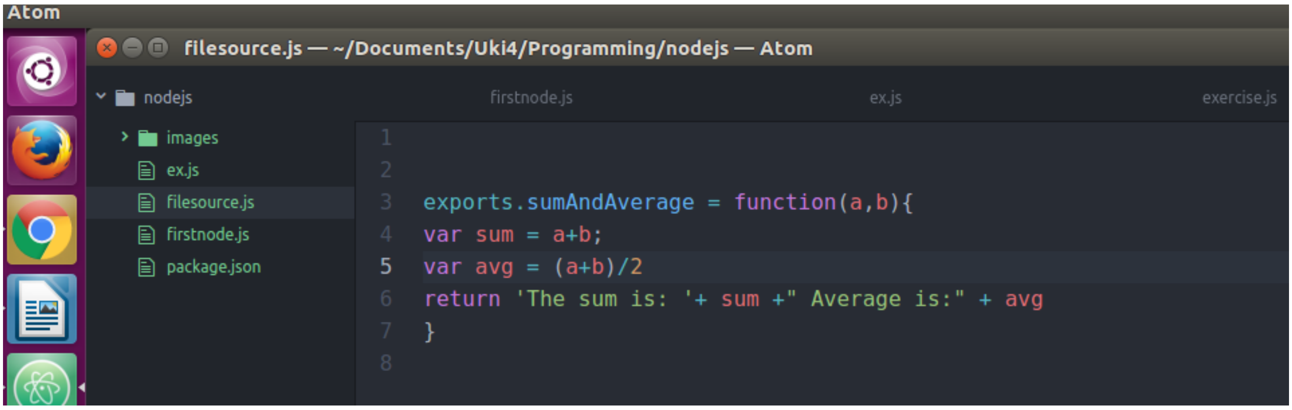
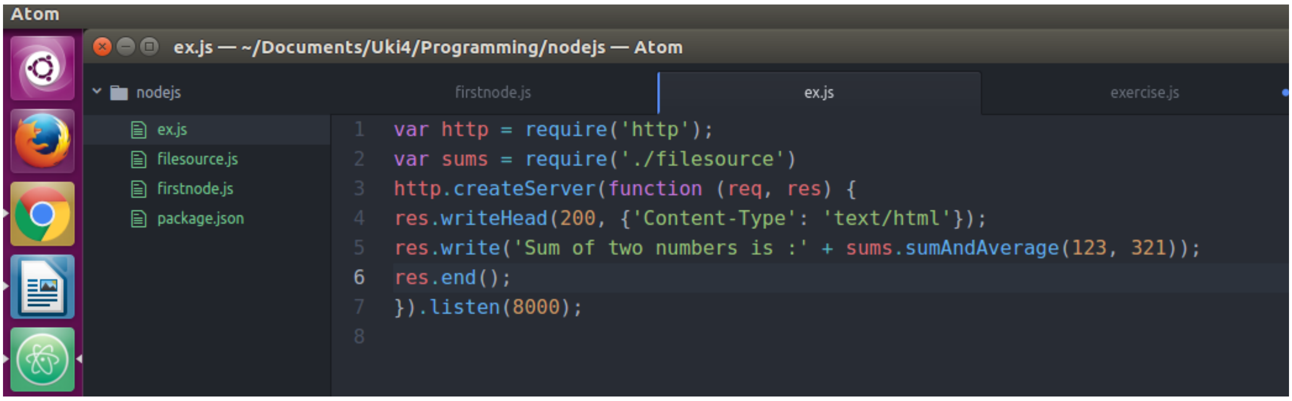
1. Create a custom module which returns the sum and average of any two numbers passed into it. Require the module and run the server by passing 123 and 321 so that the server prints out the sum and average.

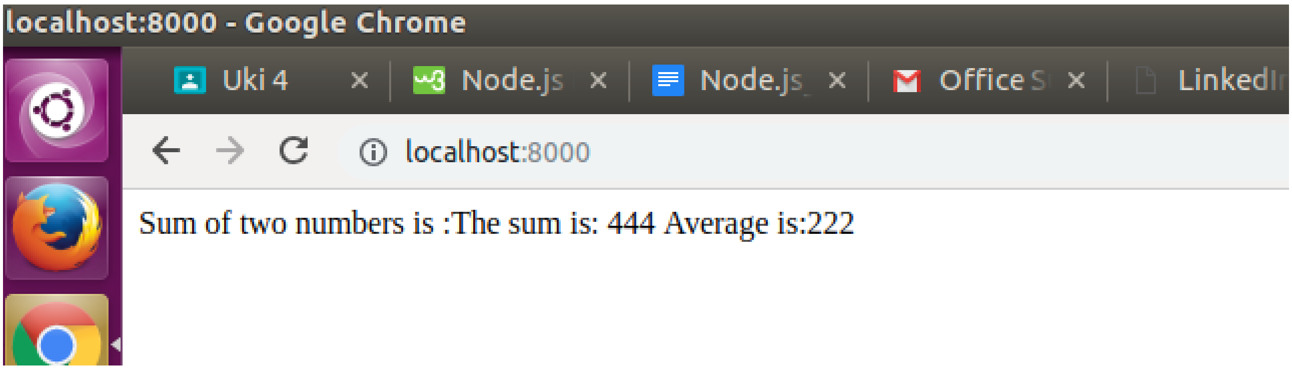
```
Code: exports.sumAndAverage = function(a,b){
var sum = a+b;
var avg = a+b/2
return 'The sum is: '+ sum +" Average is:" + avg
}
var http = require('http');
var sums = require('./filesource')
http.createServer(function (req, res) {
res.writeHead(200, {'Content-Type': 'text/html'});
res.write('Sum of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



```

exports.sumAndAverage = function(a,b){
var sum = a+b;
var avg = (a+b)/2
return 'The sum is: '+ sum +" Average is:" + avg
}

```



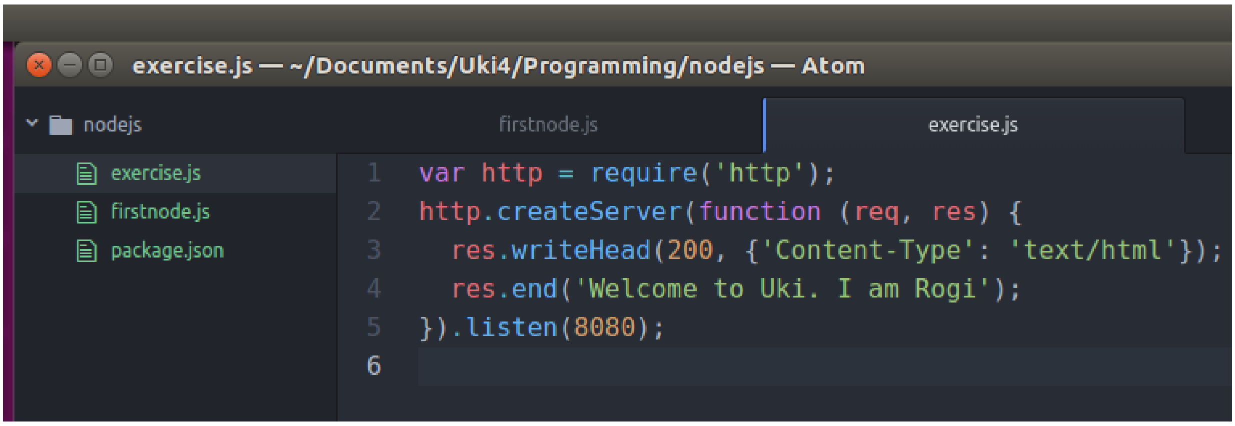
```
var http = require('http');
var sums = require('./filesource')
http.createServer(function (req, res) {
res.writeHead(200, {'Content-Type': 'text/html'});
res.write('Sum of two numbers is :' + sums.sumAndAverage(123, 321));
res.end();
}).listen(8000);
```



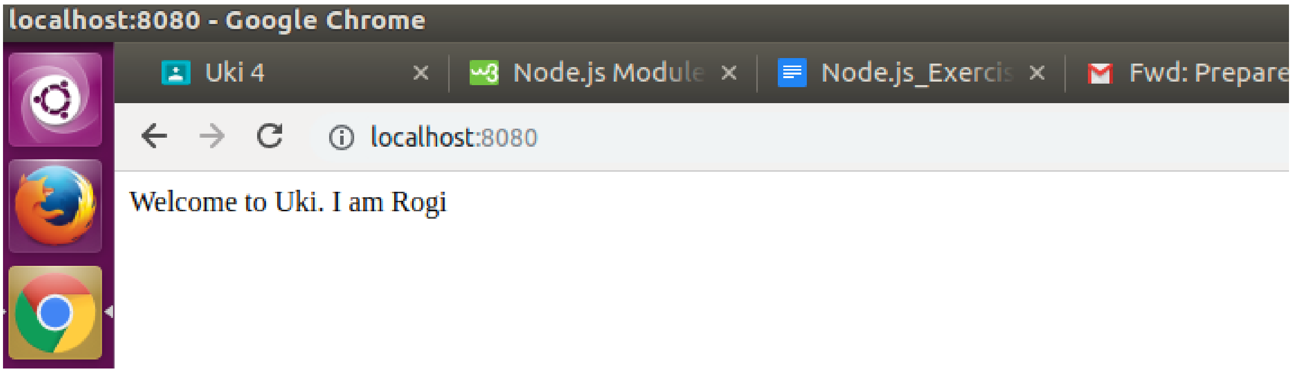
```
Sum of two numbers is :The sum is: 444 Average is:222
```

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);
```



```
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);
```

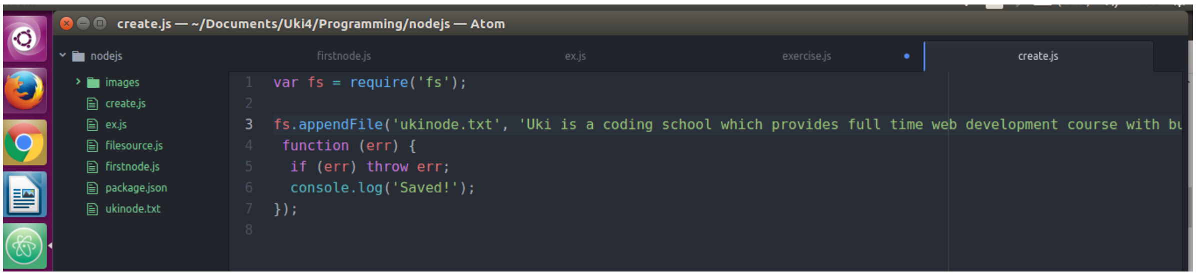


Welcome to Uki. I am Rogi

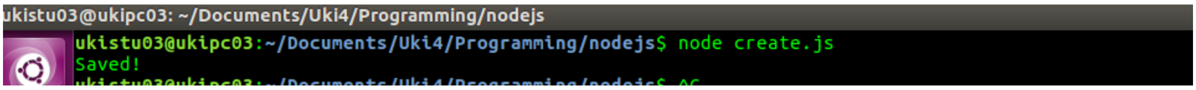
3.Using the file system module create a new file called ukinode.txt

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!');
});
```



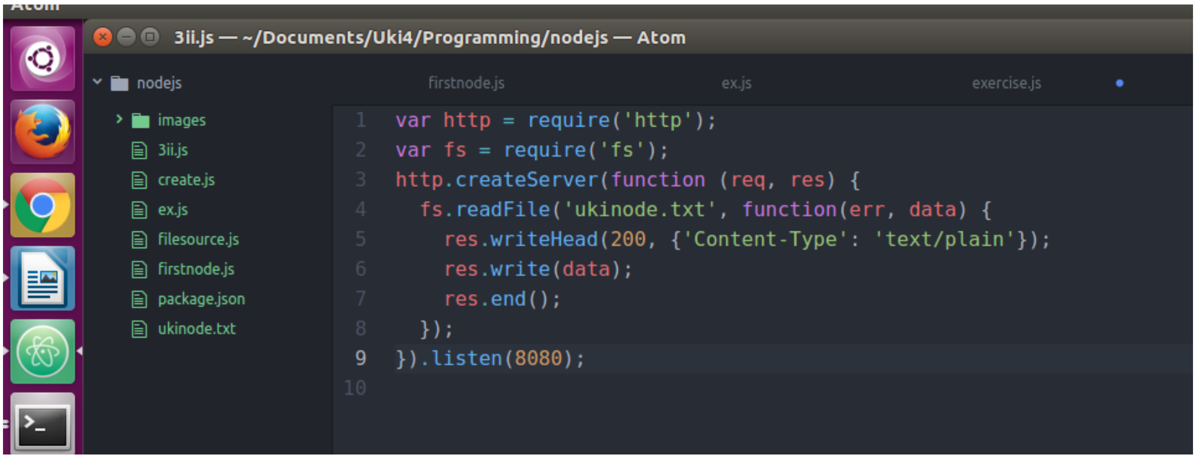
```
var fs = require('fs');

fs.appendFile('ukinode.txt', 'Uki is a coding school which provides full time web development course with bu
 function (err) {
  if (err) throw err;
  console.log('Saved!');
});
```



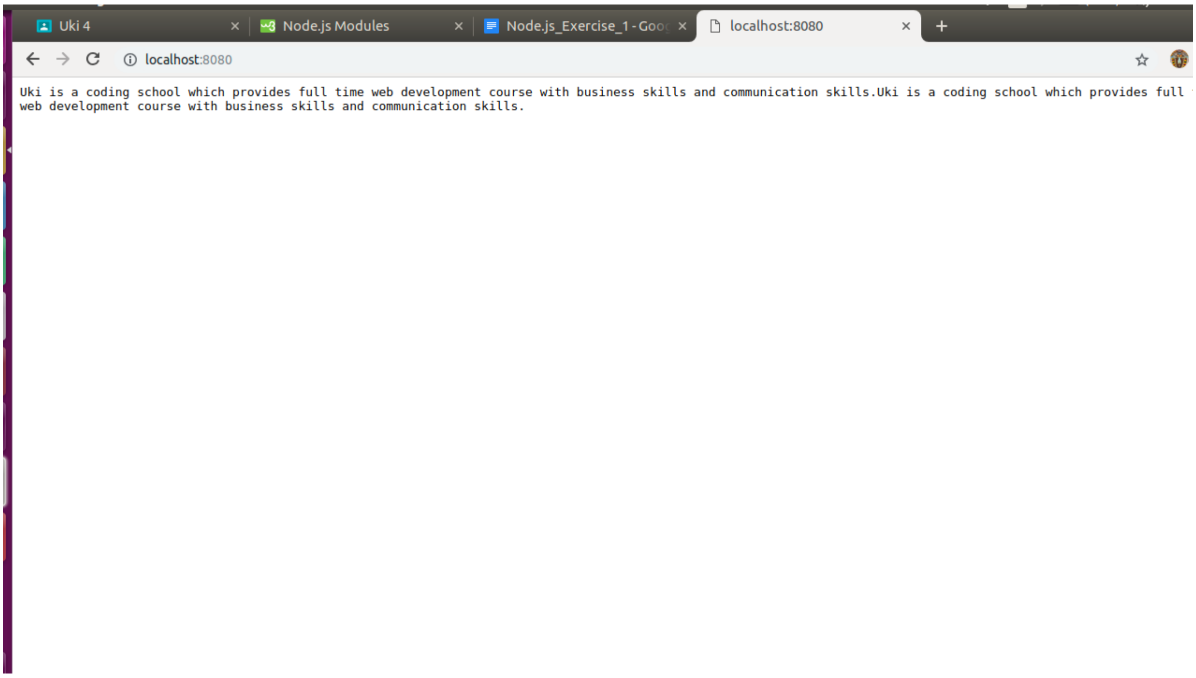
```
ukistu03@ukipc03:~/Documents/Uki4/Programming/nodejs$ node create.js
Saved!
ukistu03@ukipc03:~/Documents/Uki4/Programming/nodejs$ ^C
```

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);
```



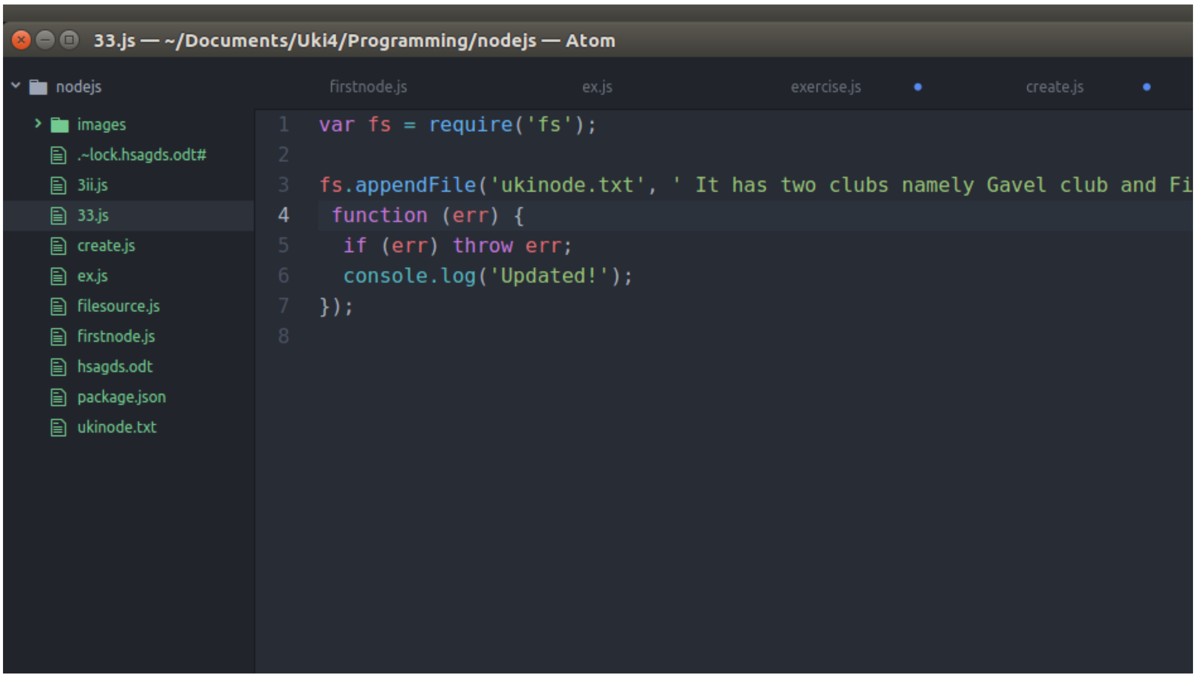
```
var http = require('http');
var fs = require('fs');
http.createServer(function (req, res) {
  fs.readFile('ukinode.txt', function(err, data) {
    res.writeHead(200, {'Content-Type': 'text/plain'});
    res.write(data);
    res.end();
  });
}).listen(8080);
```



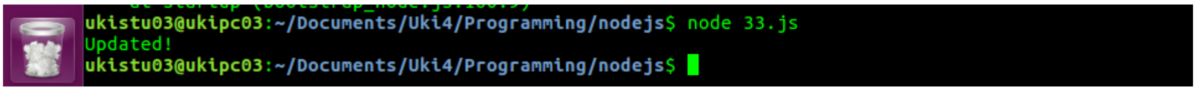
```
Uki is a coding school which provides full time web development course with business skills and communication skills.Uki is a coding school which provides full
web development course with business skills and communication skills.
```

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!');
});
```



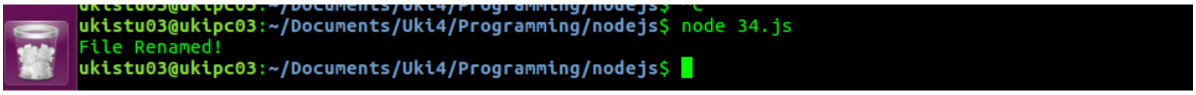
```
var fs = require('fs');

fs.appendFile('ukinode.txt', ' It has two clubs namely Gavel club and Fi
 function (err) {
  if (err) throw err;
  console.log('Updated!');
});
```



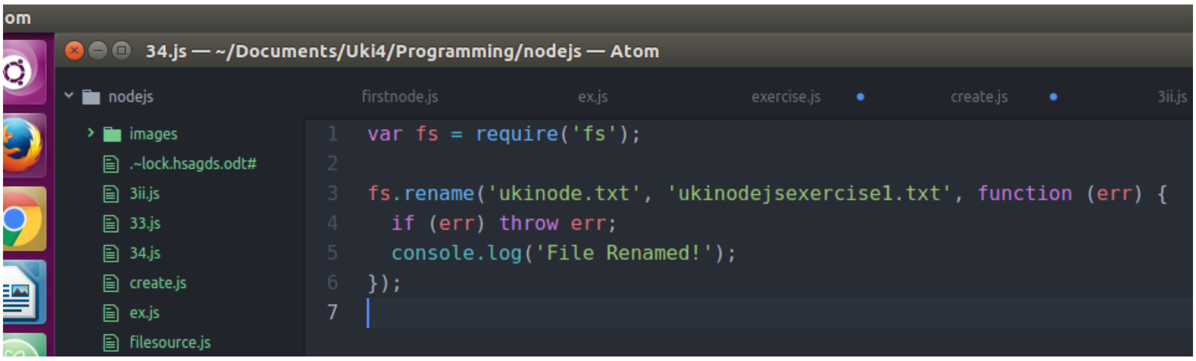
```
    at startup (bootstrap_node.js:160:9)
ukistu03@ukipc03:~/Documents/Uki4/Programming/nodejs$ node 33.js
Updated!
ukistu03@ukipc03:~/Documents/Uki4/Programming/nodejs$ 
```

3.4 Rename the file as ukinodejsexercise1.txt

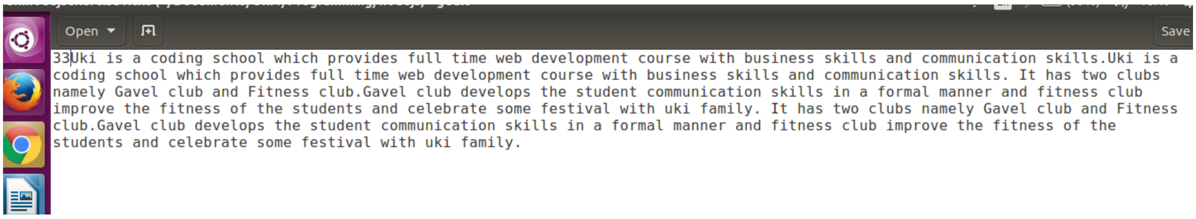
```
code: var fs = require('fs');

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



```
ukistu03@ukipc03:~/Documents/Uki4/Programming/nodejs$ ^C
ukistu03@ukipc03:~/Documents/Uki4/Programming/nodejs$ node 34.js
File Renamed!
ukistu03@ukipc03:~/Documents/Uki4/Programming/nodejs$ 
```



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

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

```

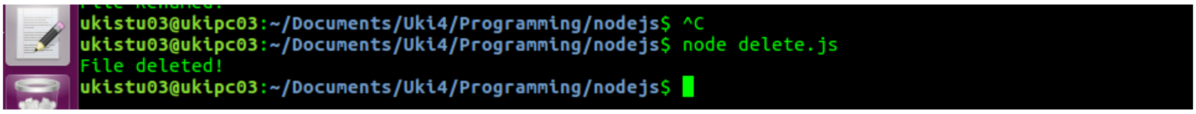


```
33Uki is a coding school which provides full time web development course with business skills and communication skills.Uki is a
coding school which provides full time web development course with business skills and communication skills. It has two clubs
namely Gavel club and Fitness club.Gavel club 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. 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.
```

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!');
});
```

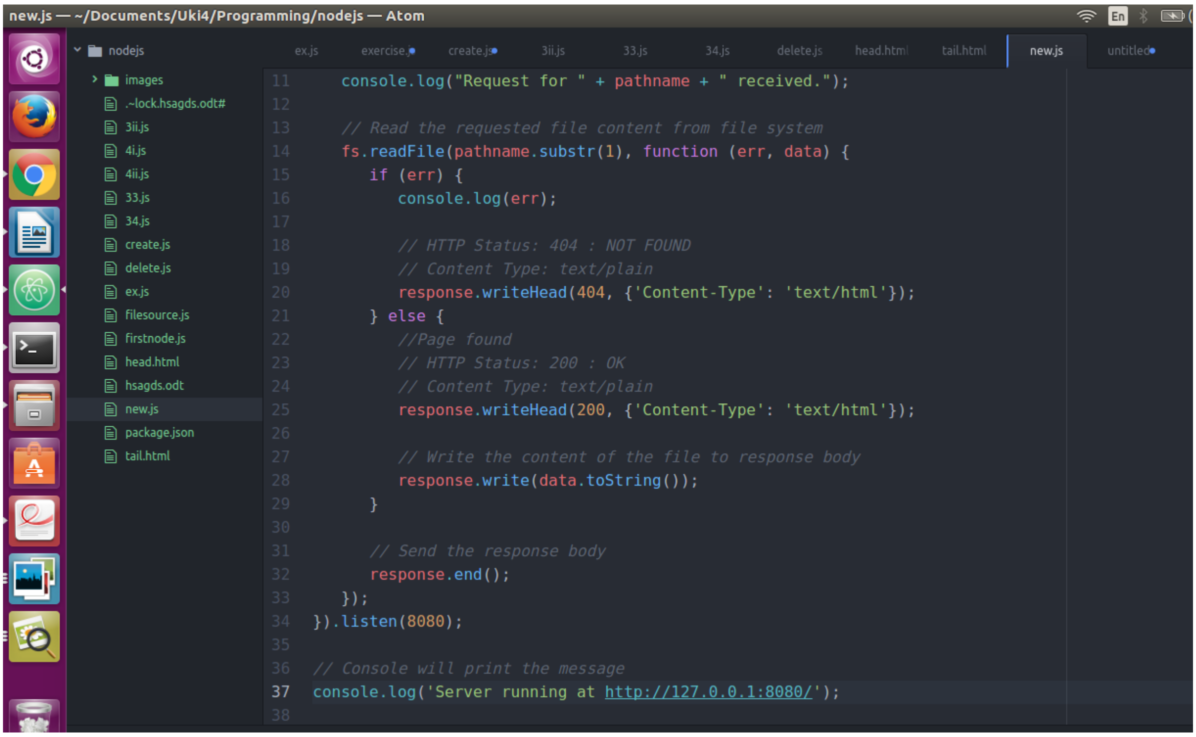


```
ukistu03@ukipc03:~/Documents/Uki4/Programming/nodejs$ ^C
ukistu03@ukipc03:~/Documents/Uki4/Programming/nodejs$ node delete.js
File deleted!
ukistu03@ukipc03:~/Documents/Uki4/Programming/nodejs$
```

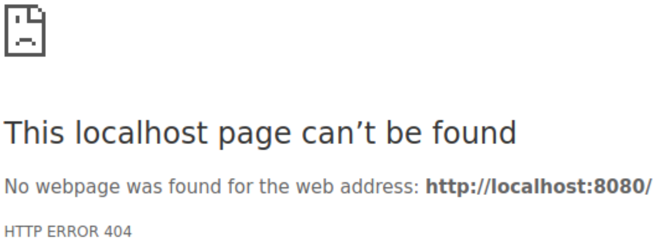
4. Create two html files called head.html which is a web page which says ‘you have got head ‘ and tail.html which is a web page which says ‘you have got tail’ and save them in the same folder as your node.js files. Create a Node.js file that opens the requested file and returns the content to the client. 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/');
```



```
    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/');

```



This localhost page can’t be found

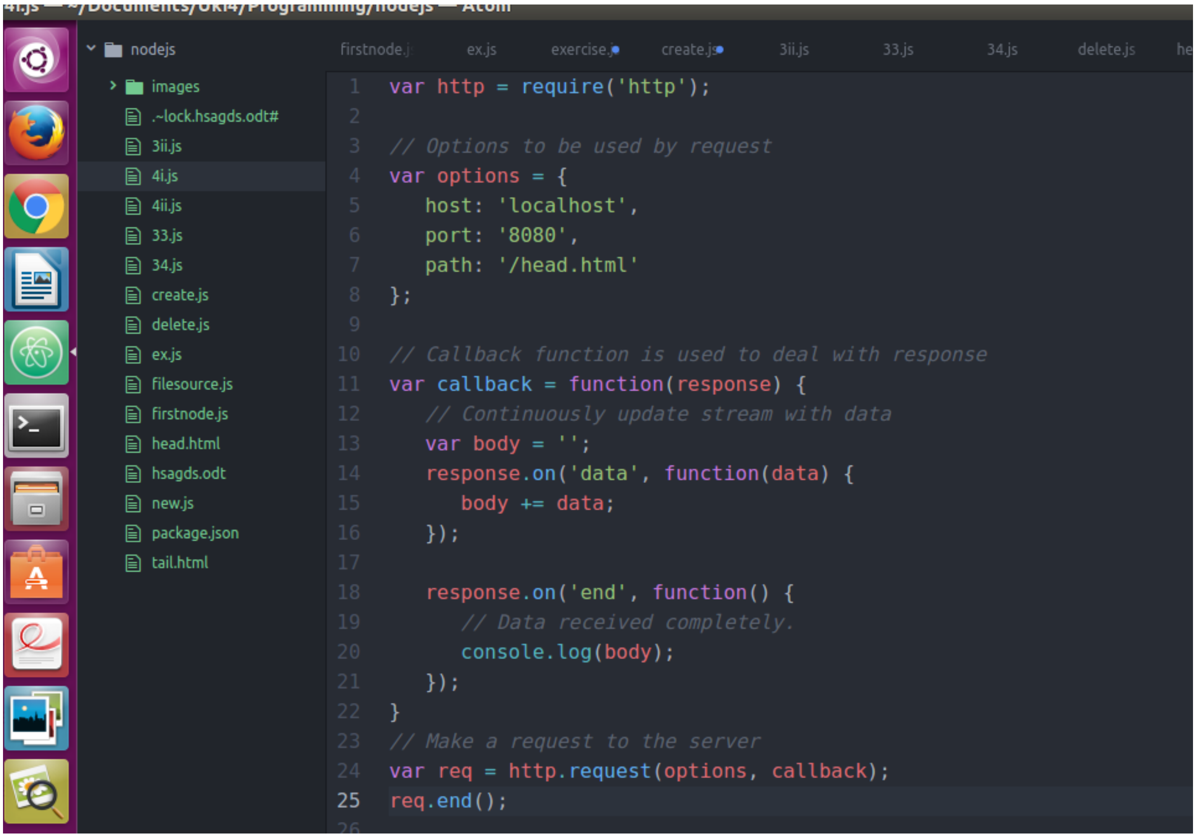
No webpage was found for the web address: **http://localhost:8080/**

HTTP ERROR 404

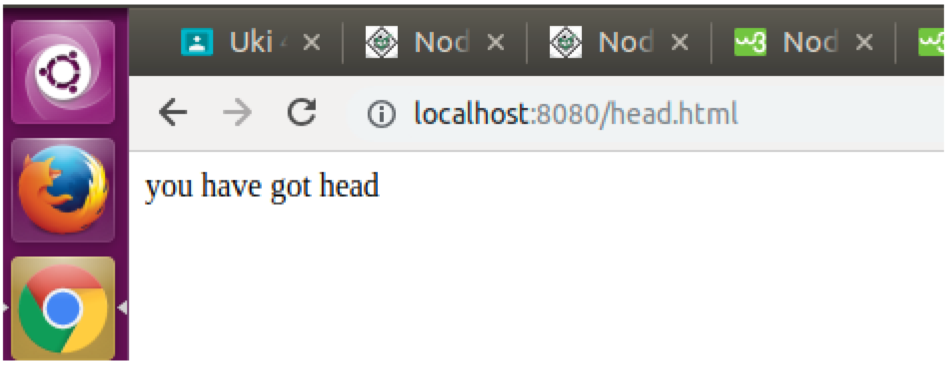
If you have followed the correct steps you should see two different results when opening these two addresses:

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();
```



```
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();
```



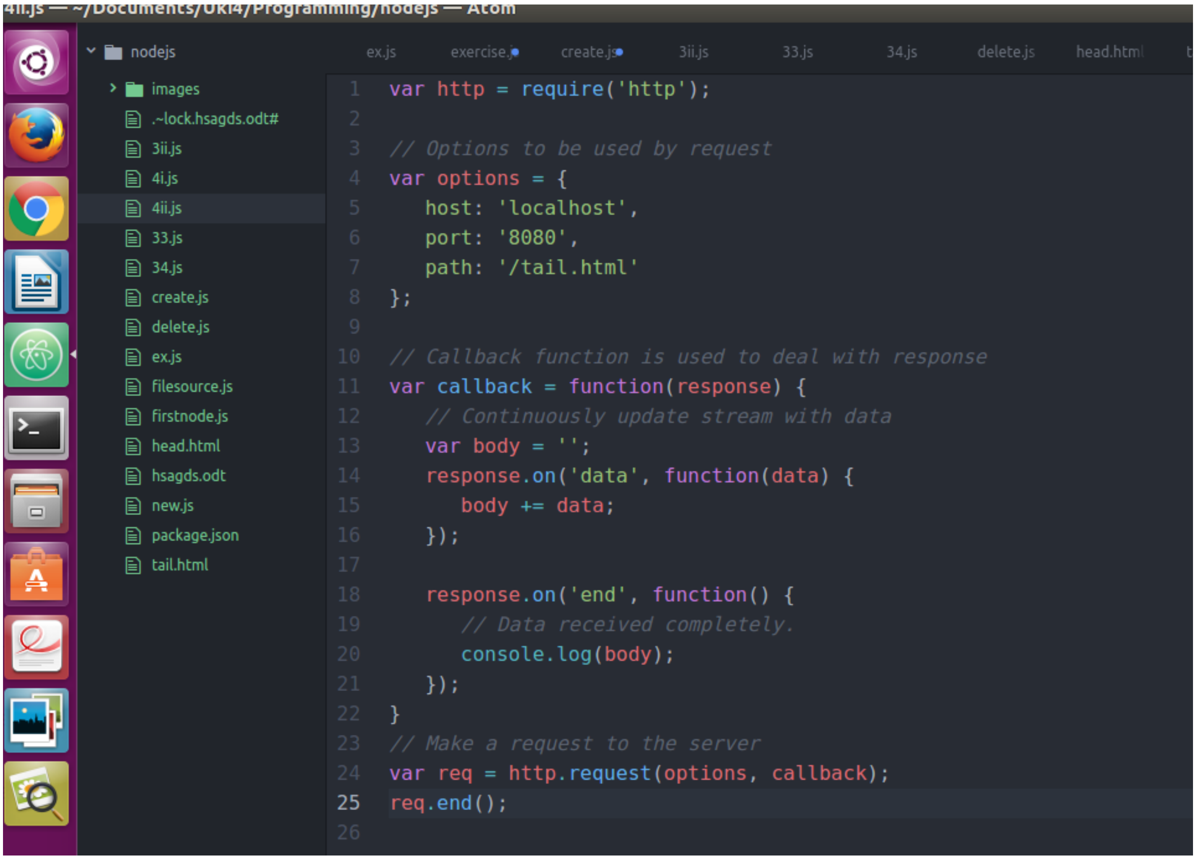
you have got head

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

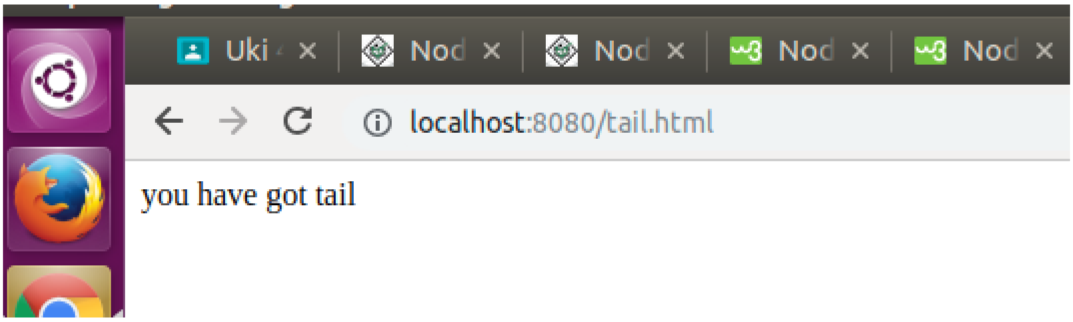
code:var http = require('http');

```
// 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();
```



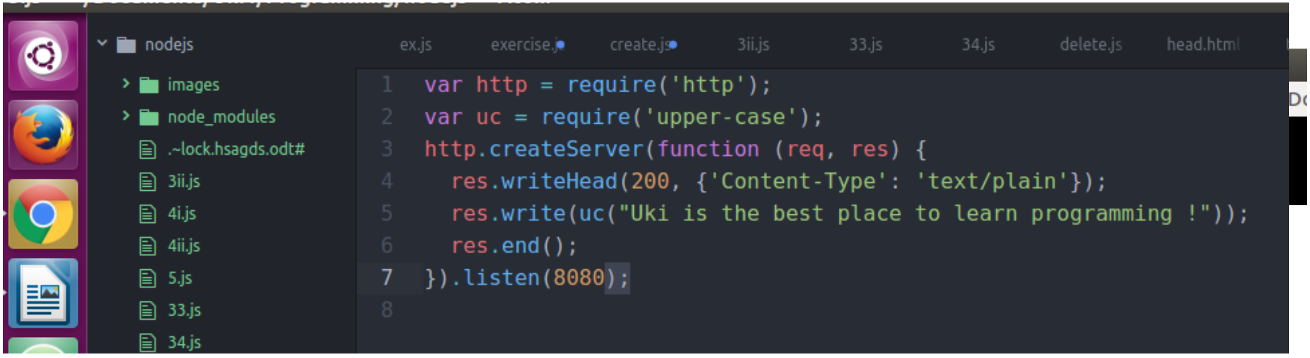
```
var http = require('http');

// 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();
```



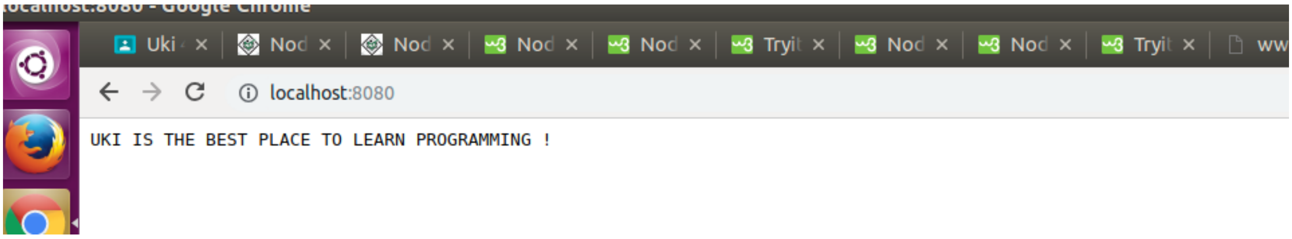
you have got tail

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);
```



```
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);
```

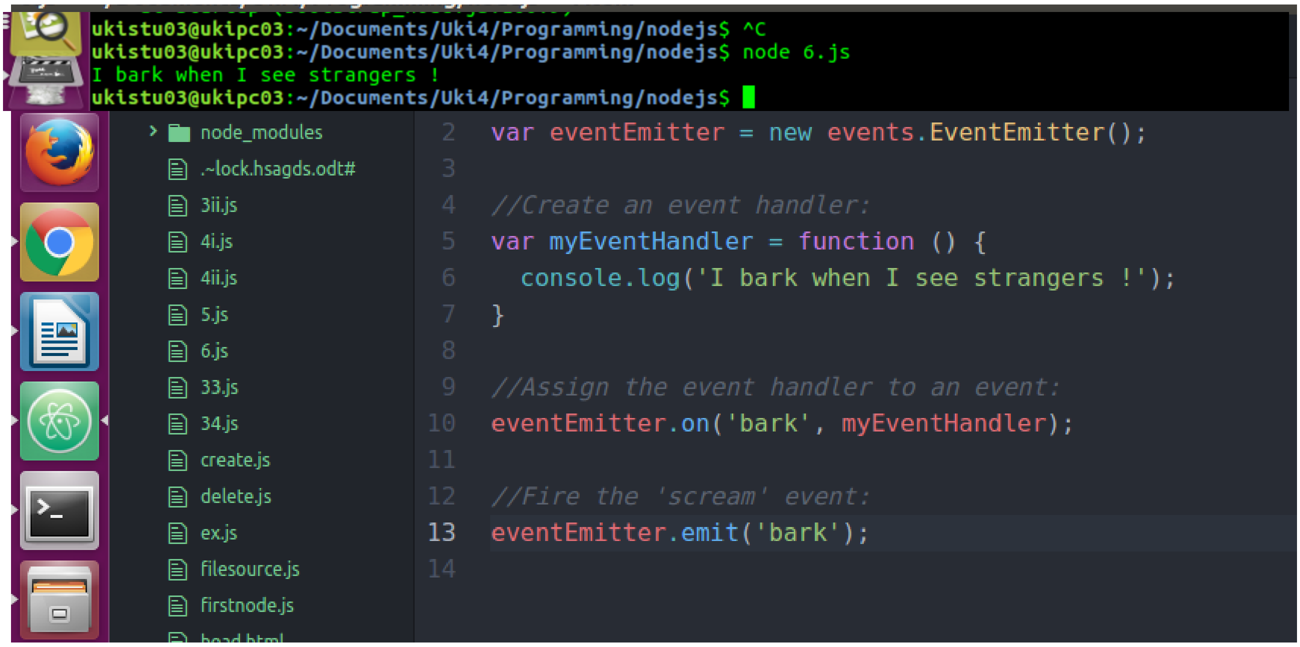


```
UKI IS THE BEST PLACE TO LEARN PROGRAMMING !
```

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');
```

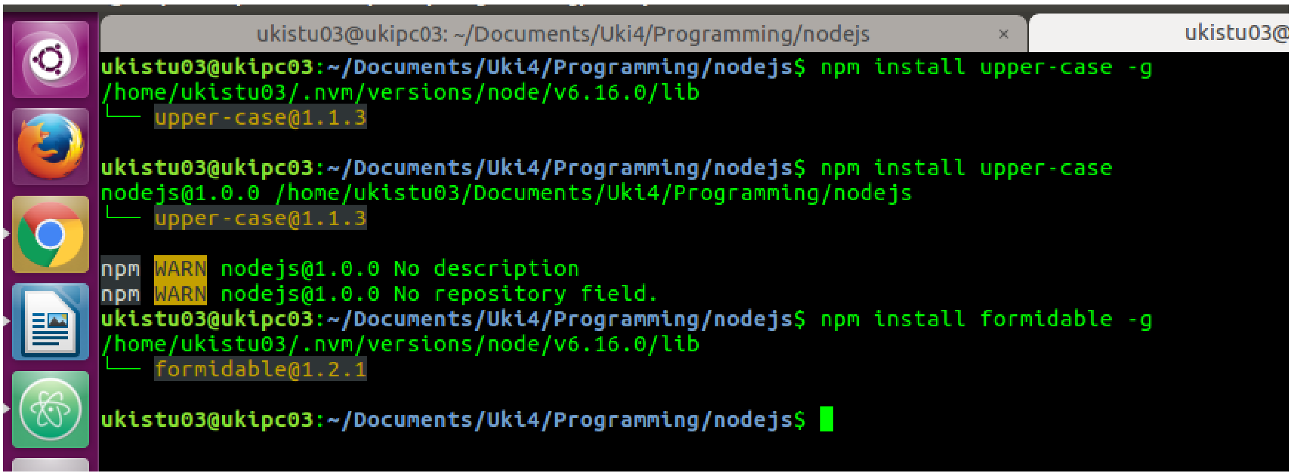


```
ukistu03@ukipc03:~/Documents/Uki4/Programming/nodejs$ ^C
ukistu03@ukipc03:~/Documents/Uki4/Programming/nodejs$ node 6.js
I bark when I see strangers !
ukistu03@ukipc03:~/Documents/Uki4/Programming/nodejs$
```

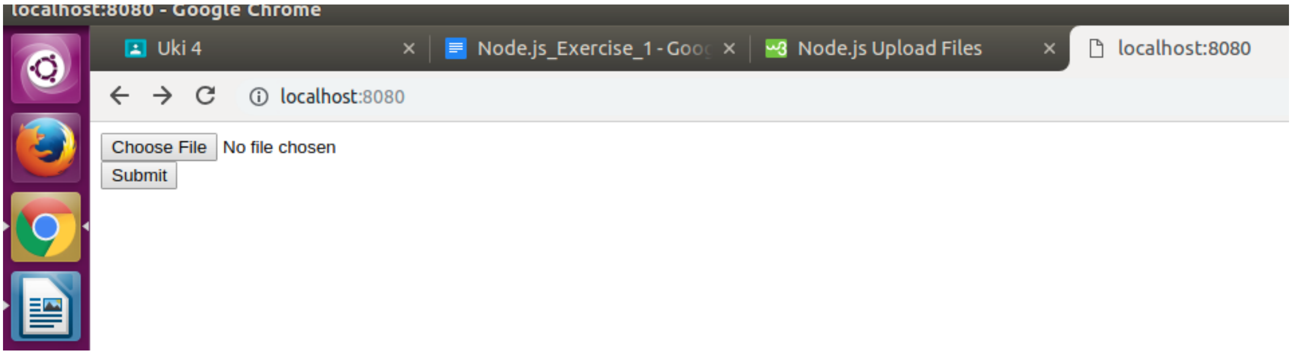
```
var eventEmitter = new events.EventEmitter();

//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');

```

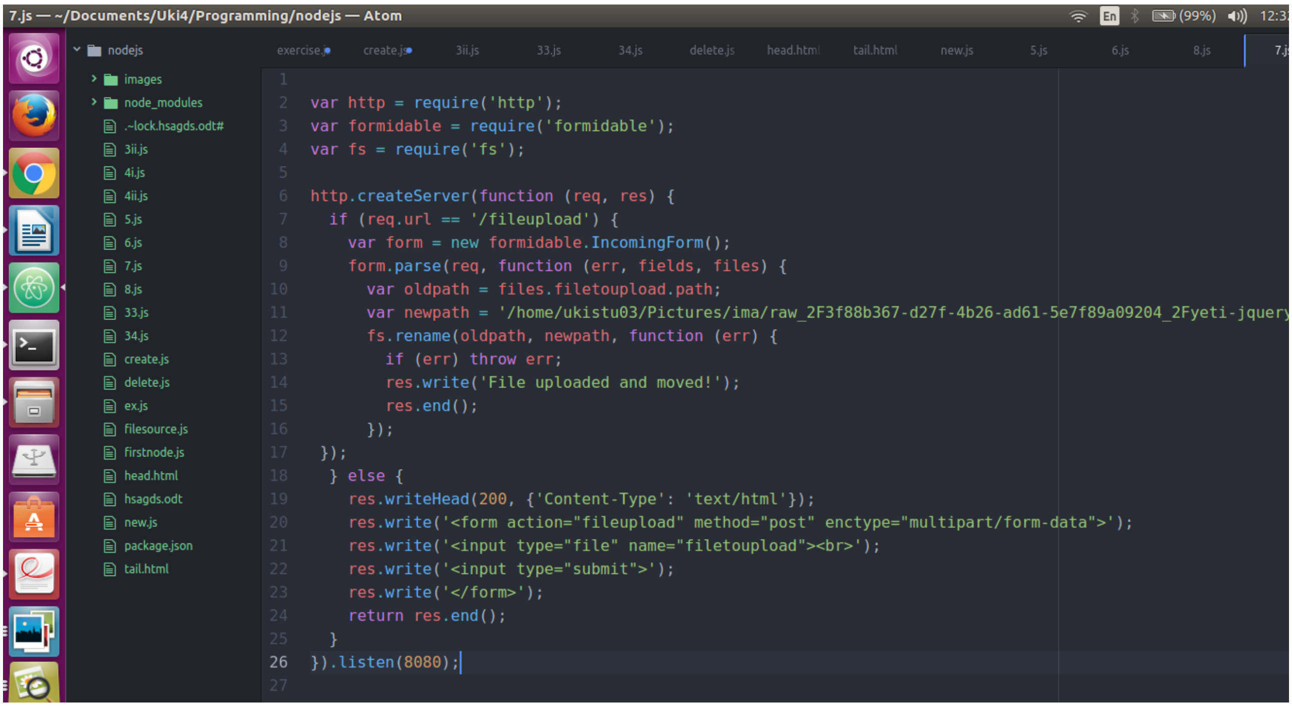
7. Install “formidable” module using npm and make a web page in Node.js that lets the user upload files to your computer.

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



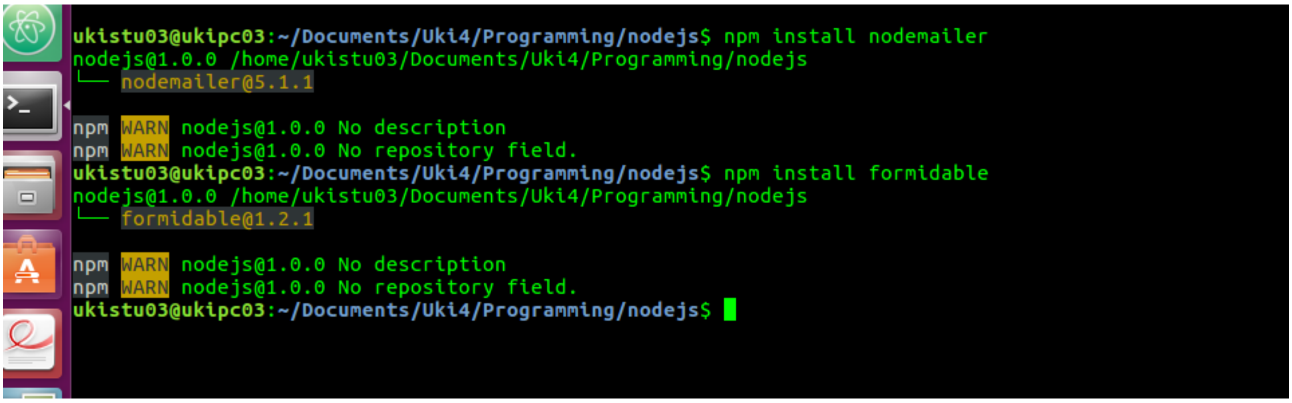
```
ukistu03@ukipc03:~/Documents/Uki4/Programming/nodejs$ npm install upper-case -g
/home/ukistu03/.nvm/versions/node/v6.16.0/lib
└── upper-case@1.1.3

ukistu03@ukipc03:~/Documents/Uki4/Programming/nodejs$ npm install upper-case
nodejs@1.0.0 /home/ukistu03/Documents/Uki4/Programming/nodejs
└── upper-case@1.1.3

npm WARN nodejs@1.0.0 No description
npm WARN nodejs@1.0.0 No repository field.
ukistu03@ukipc03:~/Documents/Uki4/Programming/nodejs$ npm install formidable -g
/home/ukistu03/.nvm/versions/node/v6.16.0/lib
└── formidable@1.2.1

ukistu03@ukipc03:~/Documents/Uki4/Programming/nodejs$ 
```





```

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.filetoupload.path;
      var newpath = '/home/ukistu03/Pictures/ima/raw_2F3f88b367-d27f-4b26-ad61-5e7f89a09204_2Fyeti-jquery
      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);

```

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 !”

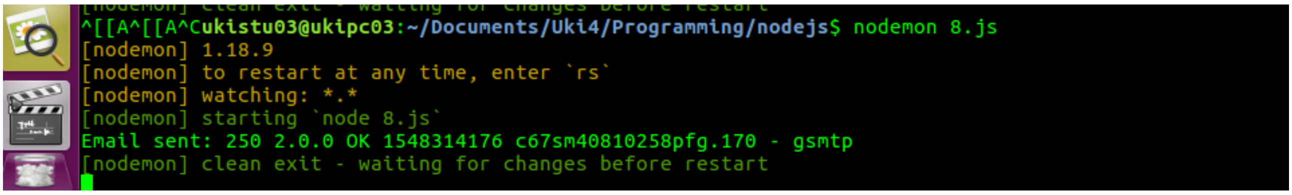


```
ukistu03@ukipc03:~/Documents/Uki4/Programming/nodejs$ npm install nodemailer
nodejs@1.0.0 /home/ukistu03/Documents/Uki4/Programming/nodejs
└── nodemailer@5.1.1

npm WARN nodejs@1.0.0 No description
npm WARN nodejs@1.0.0 No repository field.
ukistu03@ukipc03:~/Documents/Uki4/Programming/nodejs$ npm install formidable
nodejs@1.0.0 /home/ukistu03/Documents/Uki4/Programming/nodejs
└── formidable@1.2.1

npm WARN nodejs@1.0.0 No description
npm WARN nodejs@1.0.0 No repository field.
ukistu03@ukipc03:~/Documents/Uki4/Programming/nodejs$ 
```

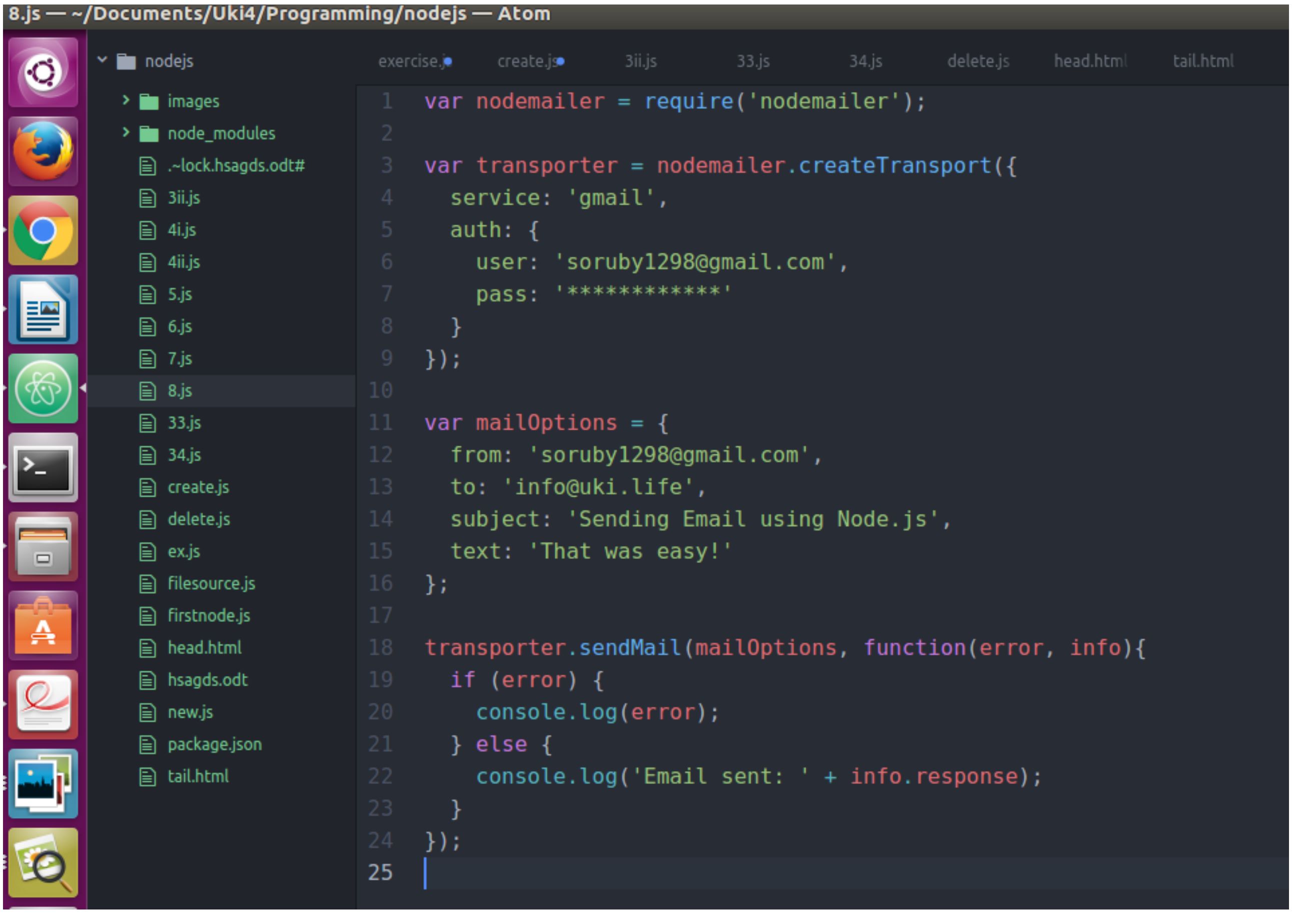
code: var nodemailer = require('nodemailer');

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);
  }
});
```



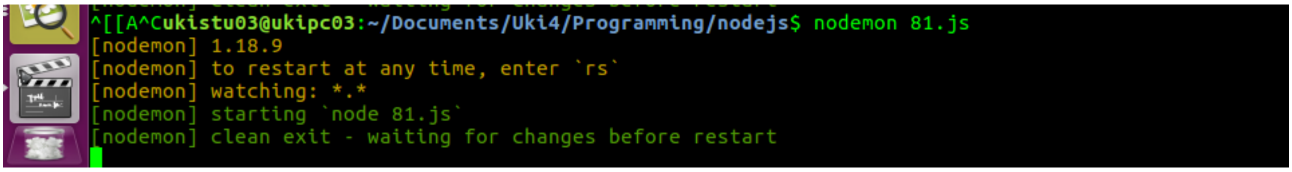
```
[nodemon] clean exit - waiting for changes before restart
^[[A^[[A^Cukistu03@ukipc03:~/Documents/Uki4/Programming/nodejs$ nodemon 8.js
[nodemon] 1.18.9
[nodemon] to restart at any time, enter `rs`
[nodemon] watching: *.*
[nodemon] starting `node 8.js`
Email sent: 250 2.0.0 OK 1548314176 c67sm40810258pfg.170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



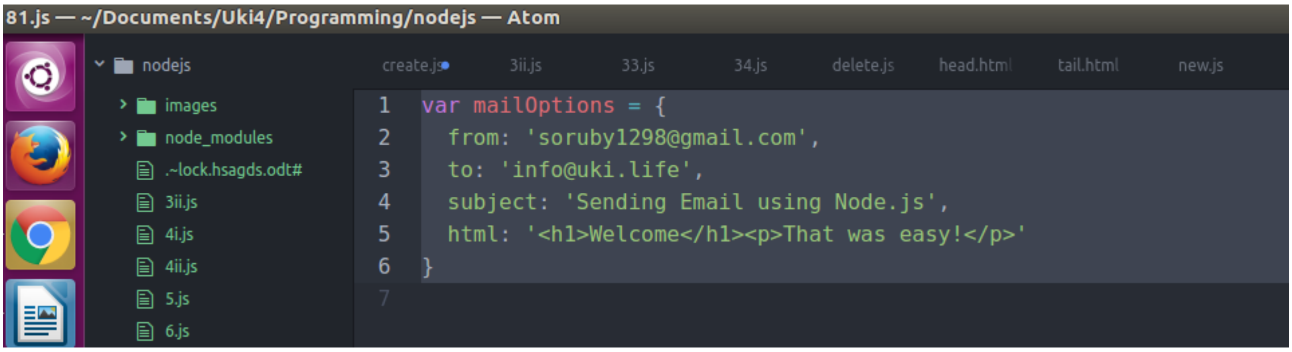
```
var nodemailer = require('nodemailer');

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);
  }
});

```

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^Cukistu03@ukipc03:~/Documents/Uki4/Programming/nodejs$ nodemon 81.js
[nodemon] 1.18.9
[nodemon] to restart at any time, enter `rs`
[nodemon] watching: *.*
[nodemon] starting `node 81.js`
[nodemon] clean exit - waiting for changes before restart
```



```
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>'
}

```