Net Ninja

https://nodejs.org/en —>download latest Features

Go to Cmd to check if **node js** is installed or not?

```
>node -v
v20.3.0

C:\Users\gudiw>node
Welcome to Node.js v20.3.0.
Type ".help" for more information.
> 5+5
10
>
> var name = 'raghava'
undefined
> name
'raghava'
>
(To exit, press Ctrl+C again or Ctrl+D or type .exit)
>
```

This is done in cmd

```
C:\Users\gudiw>

C:\Users\gudiw>cd documents

C:\Users\gudiw\Documents>cd tuts

C:\Users\gudiw\Documents\tuts>mkdir node-crash-course

C:\Users\gudiw\Documents\tuts>cd node-crash-course

C:\Users\gudiw\Documents\tuts\node-crash-course>code .

C:\Users\gudiw\Documents\tuts\node-crash-course>
```

This will open **VS code**

test.js

```
const name = 'ramu';

console.log(name);
```

O/p: ramu

Or Terminal:

```
PS C:\Users\gudiw\Documents\tuts\node-crash-course> node test
ramu
```

Lesson 2

Backtick symbol found at b/w esc & tab keys
Shift + key = tilde
Key = backtick

test.js

```
const greet = (name) => {
    console.log(`hello, ${name}`);    // Using backticks for template string
}

greet('raghava');
greet('ramu');
```

O/P:

```
PS C:\Users\gudiw\Documents\tuts\node-crash-course> node test
hello, raghava
hello, ramu
```

Global Object

global.js

```
console.log(global);
```

O/P:

```
PS C:\Users\gudiw\Documents\tuts\node-crash-course> node global
<ref *1> Object [global] {
  global: [Circular *1],
  queueMicrotask: [Function: queueMicrotask],
  clearImmediate: [Function: clearImmediate],
  setImmediate: [Function: setImmediate] {
    [Symbol(nodejs.util.promisify.custom)]: [Getter]
  },
  structuredClone: [Getter/Setter],
  clearInterval: [Function: clearInterval],
  clearTimeout: [Function: clearTimeout],
  setInterval: [Function: setInterval],
  setTimeout: [Function: setTimeout] {
    [Symbol(nodejs.util.promisify.custom)]: [Getter]
  },
  atob: [Getter/Setter],
  btoa: [Getter/Setter],
  performance: [Getter/Setter],
  fetch: [AsyncFunction: fetch],
  crypto: [Getter]
}
```

global.js

→wait for 3sec

```
global.setTimeout(() => {
   console.log('in the timeout');
}, 3000);
```

O/P:

```
PS C:\Users\gudiw\Documents\tuts\node-crash-course> node global
in the timeout
```

```
setTimeout(() => {
   console.log('in the timeout');
}, 3000);
```

→the above two codes work same

O/P:

```
PS C:\Users\gudiw\Documents\tuts\node-crash-course> node global
in the timeout
```

The setTimeout() function is called on the global object (global in Node.js or window in a web browser).

1. `() => { ... }`
   - This part of the code defines an arrow function (also known as a lambda function) that serves as the callback function. The arrow function has no parameters.

2. `3000`
   - The number `3000` represents the delay in milliseconds. It specifies how long the script should wait before executing the callback function. In this case, the delay is 3000 milliseconds, which is equivalent to 3 seconds.

```
setTimeout(() => {
   console.log('in the timeout');
}, 3000);

const int = setInterval(() => {
    console.log('in the interval');
}, 1000);
```

O/P:

```
PS C:\Users\gudiw\Documents\tuts\node-crash-course> node global
in the interval
in the interval
in the timeout
in the interval
in the interval
```

in the interval
in the interval
in the interval
in the interval

→in the above program “the interval function is not stopped”

We have to use **clearInterval()**

```
setTimeout(() => {
   console.log('in the timeout');
   clearInterval(int);
}, 3000);

const int = setInterval(() => {
    console.log('in the interval');
}, 1000);
```

O/P:

```
PS C:\Users\gudiw\Documents\tuts\node-crash-course> node global
in the interval
in the interval
in the timeout
```

```
console.log(__dirname);       //To know the current directory of the node
console.log(__filename);
```

O/P:

```
PS C:\Users\gudiw\Documents\tuts\node-crash-course> node global
C:\Users\gudiw\Documents\tuts\node-crash-course
C:\Users\gudiw\Documents\tuts\node-crash-course\global.js
```

```
console.log(document.querySelector);
```

→This is available on window object

Modules & Require

modules.js

```
const xyz = require('./people');

console.log(xyz);

console.log(xyz.people, xyz.ages);
```

people.js

```
const people = ['raghava', 'ryu', 'chun-li', 'shreya'];
const ages = [20, 25, 30, 35];

console.log(people);

//module.exports = 'hello';

//module.exports = people;  //this will export single thing

module.exports = {
   people, ages
};
```

O/P:

```
PS C:\Users\gudiw\Documents\tuts\node-crash-course> node modules
[ 'raghava', 'ryu', 'chun-li', 'shreya' ]

{
  people: [ 'raghava', 'ryu', 'chun-li', 'shreya' ],
  ages: [ 20, 25, 30, 35 ]
```

}

['raghava', 'ryu', 'chun-li', 'shreya'] [20, 25, 30, 35]

→**module.js** written different way also

```
const {people, ages} = require('./people');

//console.log(xyz);

console.log(people, ages);
```

→Both are gives same Output

→extended/ adding some script in **module.js**

```
const {people, ages} = require('./people');

console.log(people, ages);

const os = require('os');

console.log(os);

console.log(os.platform(), os.homedir());
```

O/P:

```
PS C:\Users\gudiw\Documents\tuts\node-crash-course> node modules
[ 'raghava', 'ryu', 'chun-li', 'shreya' ]
[ 'raghava', 'ryu', 'chun-li', 'shreya' ] [ 20, 25, 30, 35 ]
{
  arch: [Function: arch] {
    [Symbol(Symbol.toPrimitive)]: [Function (anonymous)]
  },
  availableParallelism: [Function: getAvailableParallelism] {
```

```
  [Symbol(Symbol.toPrimitive)]: [Function (anonymous)]
},
cpus: [Function: cpus],
endianness: [Function: endianness] {
  [Symbol(Symbol.toPrimitive)]: [Function (anonymous)]
},
freemem: [Function: getFreeMem] {
  [Symbol(Symbol.toPrimitive)]: [Function (anonymous)]
},
getPriority: [Function: getPriority],
homedir: [Function: __node_internal_checkError] {
  [Symbol(Symbol.toPrimitive)]: [Function (anonymous)]
},
hostname: [Function: __node_internal_checkError] {
  [Symbol(Symbol.toPrimitive)]: [Function (anonymous)]
},
loadavg: [Function: loadavg],
networkInterfaces: [Function: networkInterfaces],
platform: [Function: platform] {
  [Symbol(Symbol.toPrimitive)]: [Function (anonymous)]
},
release: [Function: getOSRelease] {
  [Symbol(Symbol.toPrimitive)]: [Function (anonymous)]
},
setPriority: [Function: setPriority],
tmpdir: [Function: tmpdir] {
  [Symbol(Symbol.toPrimitive)]: [Function (anonymous)]
},
totalmem: [Function: getTotalMem] {
  [Symbol(Symbol.toPrimitive)]: [Function (anonymous)]
},
type: [Function: getOSType] {
  [Symbol(Symbol.toPrimitive)]: [Function (anonymous)]
},
userInfo: [Function: userInfo],
uptime: [Function: __node_internal_checkError] {
  [Symbol(Symbol.toPrimitive)]: [Function (anonymous)]
},
version: [Function: getOSVersion] {
  [Symbol(Symbol.toPrimitive)]: [Function (anonymous)]
```

```
},
machine: [Function: getMachine] {
  [Symbol(Symbol.toPrimitive)]: [Function (anonymous)]
},
constants: [Object: null prototype] {
  UV_UDP_REUSEADDR: 4,
  dlopen: [Object: null prototype] {},
  errno: [Object: null prototype] {
    E2BIG: 7,
    EACCES: 13,
    EADDRINUSE: 100,
    EADDRNOTAVAIL: 101,
    EAFNOSUPPORT: 102,
    EAGAIN: 11,
    EALREADY: 103,
    EBADF: 9,
    EBADMSG: 104,
    EBUSY: 16,
    ECANCELED: 105,
    ECHILD: 10,
    ECONNABORTED: 106,
    ECONNREFUSED: 107,
    ECONNRESET: 108,
    EDEADLK: 36,
    EDESTADDRREQ: 109,
    EDOM: 33,
    EEXIST: 17,
    EFAULT: 14,
    EFBIG: 27,
    EHOSTUNREACH: 110,
    EIDRM: 111,
    EILSEQ: 42,
    EINPROGRESS: 112,
    EINTR: 4,
    EINVAL: 22,
    EIO: 5,
    EISCONN: 113,
    EISDIR: 21,
    ELOOP: 114,
    EMFILE: 24,
    EMLINK: 31,
    EMSGSIZE: 115,
    ENAMETOOLONG: 38,
    ENETDOWN: 116,
    ENETRESET: 117,
    ENETUNREACH: 118,
    ENFILE: 23,
    ENOBUFS: 119,
    ENODATA: 120,
    ENODEV: 19,
    ENOENT: 2,
    ENOEXEC: 8,
    ENOLCK: 39,
    ENOLINK: 121,
    ENOMEM: 12,
    ENOMSG: 122,
    ENOPROTOOPT: 123,
    ENOSPC: 28,
    ENOSR: 124,
    ENOSTR: 125,
    ENOSYS: 40,
    ENOTCONN: 126,
    ENOTDIR: 20,
    ENOTEMPTY: 41,
    ENOTSOCK: 128,
    ENOTSUP: 129,
    ENOTTY: 25,
    ENXIO: 6,
    EOPNOTSUPP: 130,
    EOVERFLOW: 132,
    EPERM: 1,
    EPIPE: 32,
    EPROTO: 134,
    EPROTONOSUPPORT: 135,
    EPROTOTYPE: 136,
    ERANGE: 34,
    EROFS: 30,
    ESPIPE: 29,
    ESRCH: 3,
    ETIME: 137,
    ETIMEDOUT: 138,
    ETXTBSY: 139,
    EWOULDBLOCK: 140,
    EXDEV: 18,
    WSAEINTR: 10004,
    WSAEBADF: 10009,
    WSAEACCES: 10013,
    WSAEFAULT: 10014,
    WSAEINVAL: 10022,
    WSAEMFILE: 10024,
    WSAEWOULDBLOCK: 10035,
    WSAEINPROGRESS: 10036,
    WSAEALREADY: 10037,
    WSAENOTSOCK: 10038,
    WSAEDESTADDRREQ: 10039,
    WSAEMSGSIZE: 10040,
    WSAEPROTOTYPE: 10041,
    WSAENOPROTOOPT: 10042,
    WSAEPROTONOSUPPORT: 10043,
    WSAESOCKTNOSUPPORT: 10044,
    WSAEOPNOTSUPP: 10045,
    WSAEPFNOSUPPORT: 10046,
    WSAEAFNOSUPPORT: 10047,
    WSAEADDRINUSE: 10048,
    WSAEADDRNOTAVAIL: 10049,
    WSAENETDOWN: 10050,
    WSAENETUNREACH: 10051,
    WSAENETRESET: 10052,
    WSAECONNABORTED: 10053,
    WSAECONNRESET: 10054,
    WSAENOBUFS: 10055,
    WSAEISCONN: 10056,
    WSAENOTCONN: 10057,
    WSAESHUTDOWN: 10058,
    WSAETOOMANYREFS: 10059,
    WSAETIMEDOUT: 10060,
    WSAECONNREFUSED: 10061,
    WSAELOOP: 10062,
    WSAENAMETOOLONG: 10063,
    WSAEHOSTDOWN: 10064,
    WSAEHOSTUNREACH: 10065,
    WSAENOTEMPTY: 10066,
    WSAEPROCLIM: 10067,
    WSAEUSERS: 10068,
    WSAEDQUOT: 10069,
    WSAESTALE: 10070,
    WSAEREMOTE: 10071,
    WSASYSNOTREADY: 10091,
    WSAVERNOTSUPPORTED: 10092,
    WSANOTINITIALISED: 10093,
    WSAEDISCON: 10101,
    WSAENOMORE: 10102,
    WSAECANCELLED: 10103,
    WSAEINVALIDPROCTABLE: 10104,
    WSAEINVALIDPROVIDER: 10105,
    WSAEPROVIDERFAILEDINIT: 10106,
    WSASYSCALLFAILURE: 10107,
    WSASERVICE_NOT_FOUND: 10108,
    WSATYPE_NOT_FOUND: 10109,
    WSA_E_NO_MORE: 10110,
    WSA_E_CANCELLED: 10111,
    WSAEREFUSED: 10112
  },
  signals: [Object: null prototype] {
    SIGHUP: 1,
    SIGINT: 2,
    SIGQUIT: 3,
    SIGILL: 4,
    SIGABRT: 22,
    SIGFPE: 8,
    SIGKILL: 9,
    SIGSEGV: 11,
    SIGTERM: 15,
    SIGBREAK: 21,
    SIGWINCH: 28
  },
  priority: [Object: null prototype] {
    PRIORITY_LOW: 19,
    PRIORITY_BELOW_NORMAL: 10,
    PRIORITY_NORMAL: 0,
      PRIORITY_ABOVE_NORMAL: -7,
      PRIORITY_HIGH: -14,
      PRIORITY_HIGHEST: -20
    }
  },
  EOL: '\r\n',
  devNull: '\\\\.\\nul'
}
win32 C:\Users\gudiw
```

→The above **bold** one is the output for **console.log(os);**

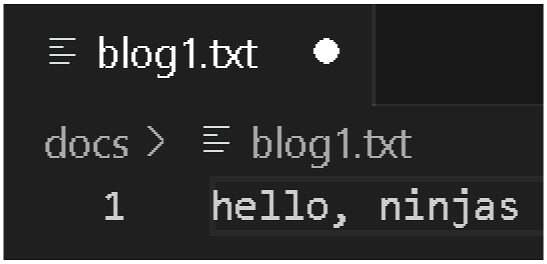
The File System

files.js

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

// reading files
fs.readFile('./docs/blog1.txt', (err, data) =>{
if(err){
    console.log(err);
}
console.log(data);                  //buffer data
console.log(data.toString());       // string data
});

console.log('last line')
```

→create a folder **docs** inside the file **blog1.txt**

docs → blog1.txt



```
hello, ninjas
```

O/P:

PS C:\Users\gudiw\Documents\tuts\node-crash-course> **node files**
last line
<Buffer 68 65 6c 6c 6f 2c 20 6e 69 6e 6a 61 73>
hello, ninjas

→last line is printing first because **readFile** it will take some to read the data.

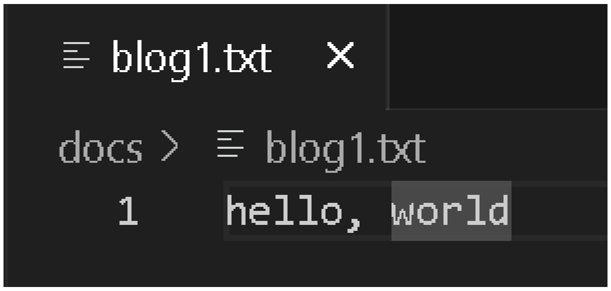
Writing the file

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

// writing files
fs.writeFile('./docs/blog1.txt', 'hello, world', () =>{
    console.log('file was written');
});

fs.writeFile('./docs/blog2.txt', 'hello, again blog2.txt is created 
automatically', () =>{
    console.log('file was written');
});
```

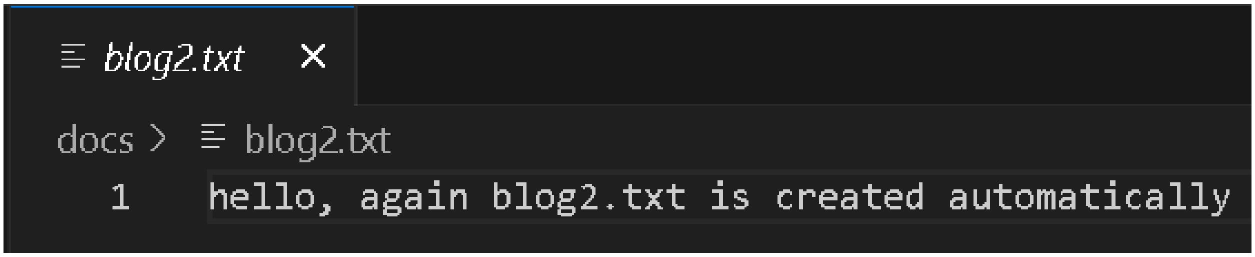
O/P:

PS C:\Users\gudiw\Documents\tuts\node-crash-course> node files

file was written
file was written



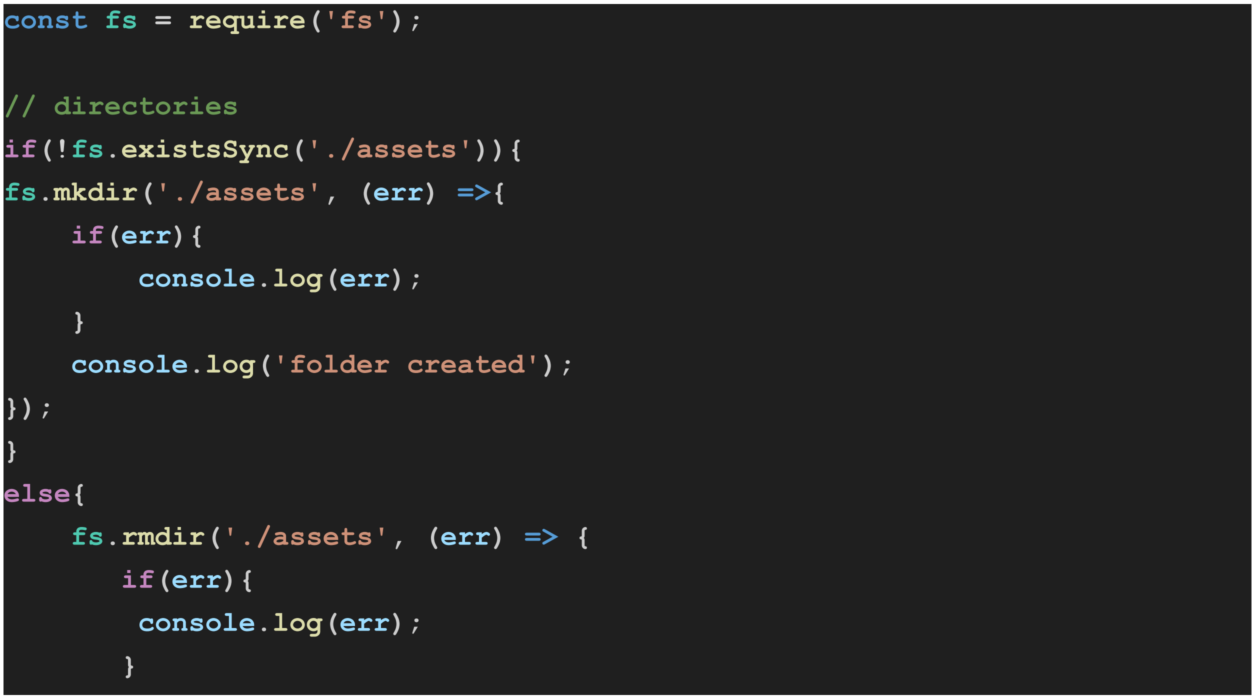
```
hello, world
```



```
hello, again blog2.txt is created automatically
```

→Initially the **blog2.txt file** is not available by using **writeFile** it is created**.**

Directories



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

// directories
if(!fs.existsSync('./assets')){
fs.mkdir('./assets', (err) =>{
    if(err){
        console.log(err);
    }
    console.log('folder created');
});
}
else{
    fs.rmdir('./assets', (err) => {
       if(err){
        console.log(err);
       }
```

```
        console.log('folder deleted');
    });
}
```

O/P:

```
PS C:\Users\gudiw\Documents\tuts\node-crash-course> node files
folder created
PS C:\Users\gudiw\Documents\tuts\node-crash-course> node files
folder deleted
PS C:\Users\gudiw\Documents\tuts\node-crash-course> node files
folder created
PS C:\Users\gudiw\Documents\tuts\node-crash-course> node files
folder deleted
PS C:\Users\gudiw\Documents\tuts\node-crash-course> node files
folder created
```

→If the directory does not exist it will be created, otherwise it will be deleted.

Deleting Files

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

// deleting files
if(fs.existsSync('./docs/deleteme.txt')){
    fs.unlink('./docs/deleteme.txt', (err) => {
    if(err){
        console.log(err);
    }
    console.log('file deleted');
    })
}
```

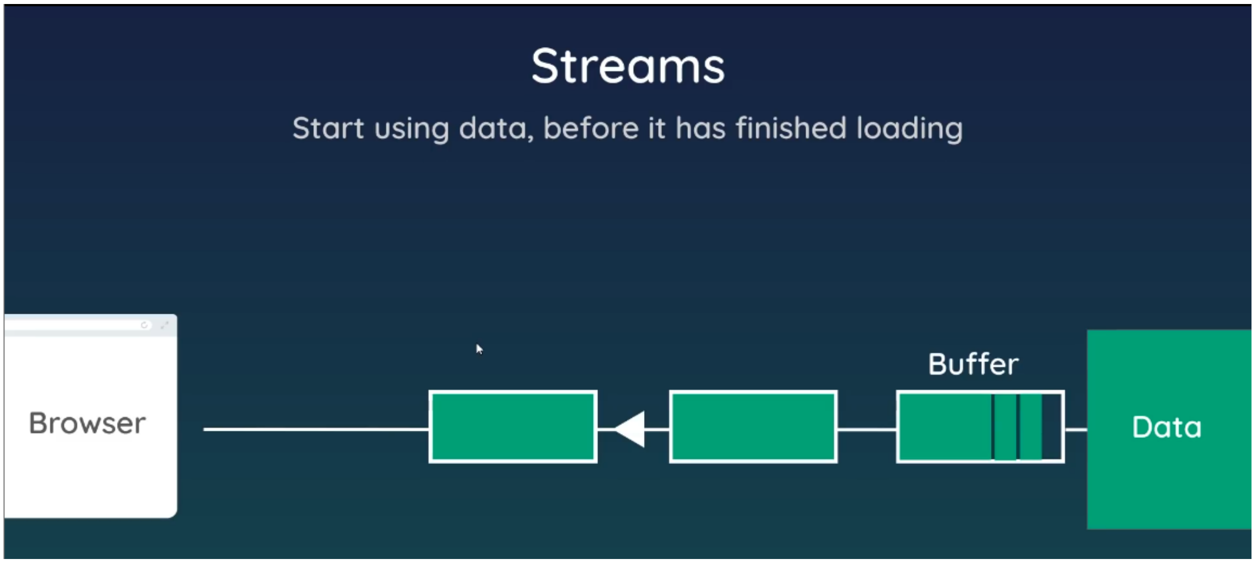
O/P:

```
PS C:\Users\gudiw\Documents\tuts\node-crash-course> node files
```

```
file deleted
PS C:\Users\gudiw\Documents\tuts\node-crash-course>
PS C:\Users\gudiw\Documents\tuts\node-crash-course> node files
```

→If the file does not exist it won't do anything. Otherwise it will delete the file.

Streams & Buffers



Streams

Start using data, before it has finished loading

→When large data is available it will take more time to load the data.

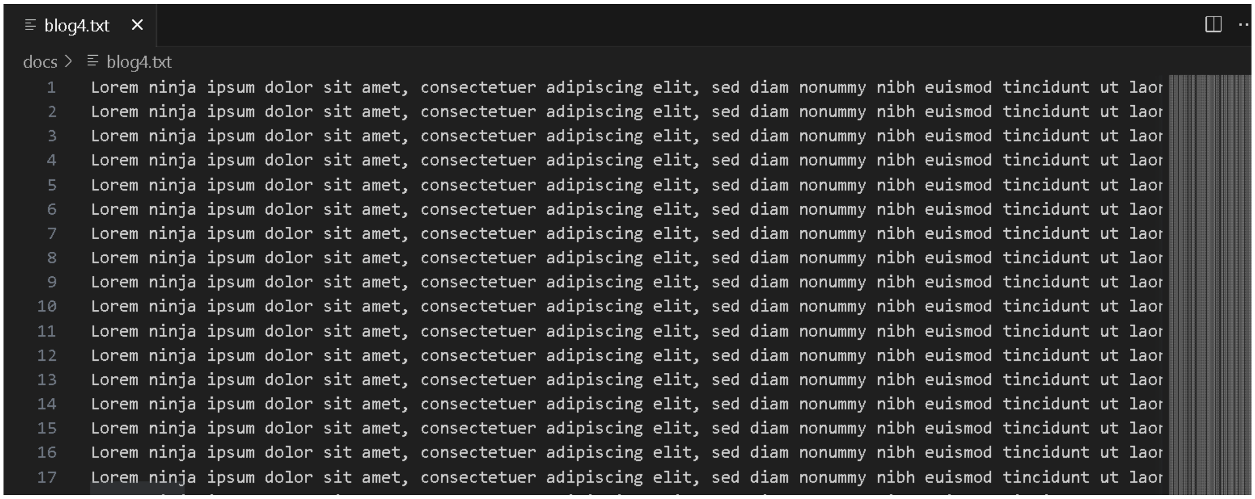
```
Lorem ninja ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laor
Lorem ninja ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laor
Lorem ninja ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laor
Lorem ninja ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laor
Lorem ninja ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laor
Lorem ninja ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laor
Lorem ninja ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laor
Lorem ninja ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laor
Lorem ninja ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laor
Lorem ninja ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laor
Lorem ninja ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laor
Lorem ninja ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laor
Lorem ninja ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laor
Lorem ninja ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laor
Lorem ninja ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laor
Lorem ninja ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laor
Lorem ninja ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laor
Lorem ninja ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laor
```

stream.js

```javascript
const fs = require('fs');

const readStream = fs.createReadStream('./docs/blog3.txt', {encoding:
'utf-8'});
const writeStream = fs.createWriteStream('./docs/blog4.txt');

// readStream.on('data', (chunk) =>{
//      console.log('----- NEW CHUNK -----');
//      console.log(chunk)
//      //console.log(chunk.toString());

//      writeStream.write('\n NEW CHUNK \n');
//      writeStream.write(chunk);
// });

//piping --> anotherway of writing to th file
readStream.pipe(writeStream);
```

O/P:

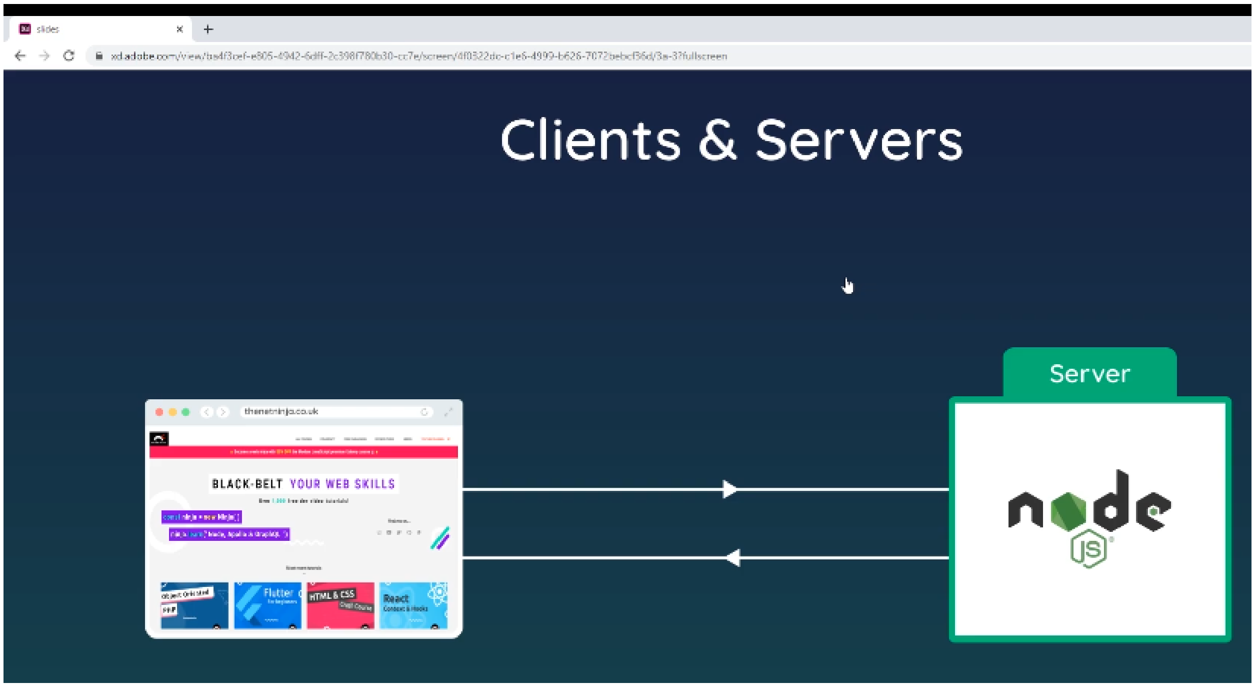
blog4.txt is created



```
Lorem ninja ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laor
Lorem ninja ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laor
Lorem ninja ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laor
Lorem ninja ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laor
Lorem ninja ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laor
Lorem ninja ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laor
Lorem ninja ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laor
Lorem ninja ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laor
Lorem ninja ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laor
Lorem ninja ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laor
Lorem ninja ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laor
Lorem ninja ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laor
Lorem ninja ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laor
Lorem ninja ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laor
Lorem ninja ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laor
Lorem ninja ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laor
Lorem ninja ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laor
```

Lesson 3

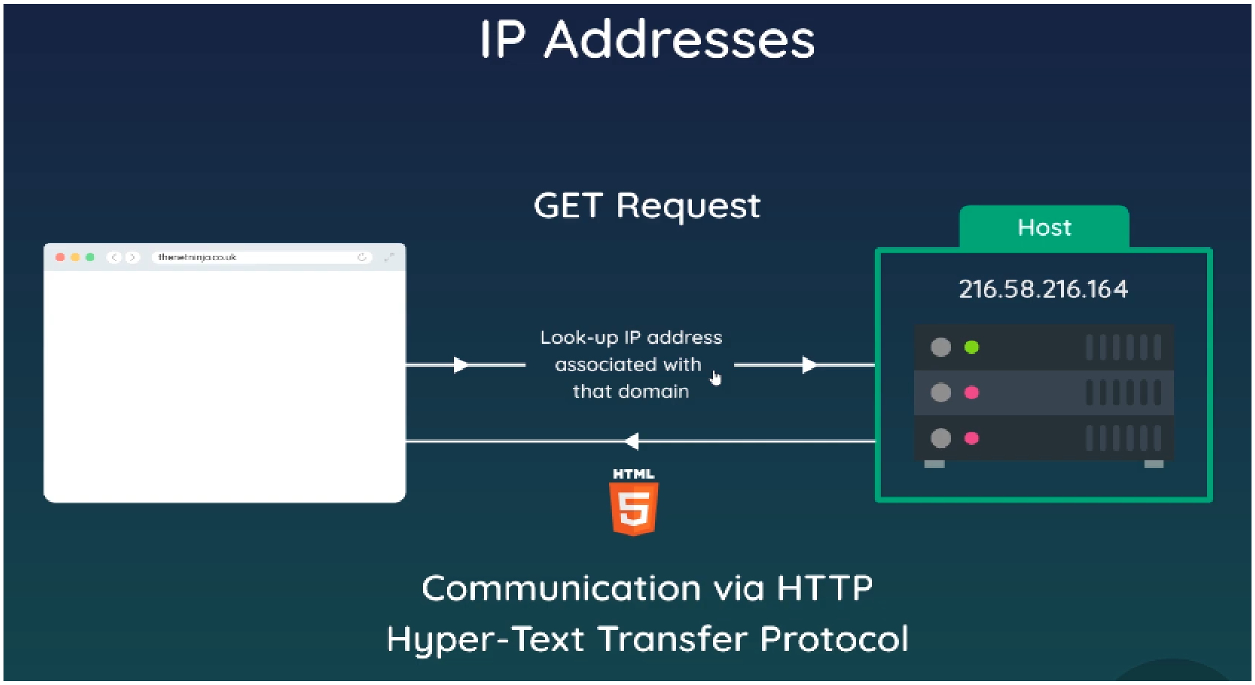
Clients & Servers



Clients & Servers

slides
xd.adobe.com/view/bs4f3cef-e805-4942-6dff-2c398f780b30-cc7e/screen/4f0322dc-c1e6-4999-b626-7072bebcf36d/3a-3?fullscreen
thenetninja.co.uk
BLACK-BELT YOUR WEB SKILLS
Flutter
HTML & CSS
React
Server
node JS

IP Addresses



IP Addresses

GET Request

Communication via HTTP
Hyper-Text Transfer Protocol

Creating a Server
server.js

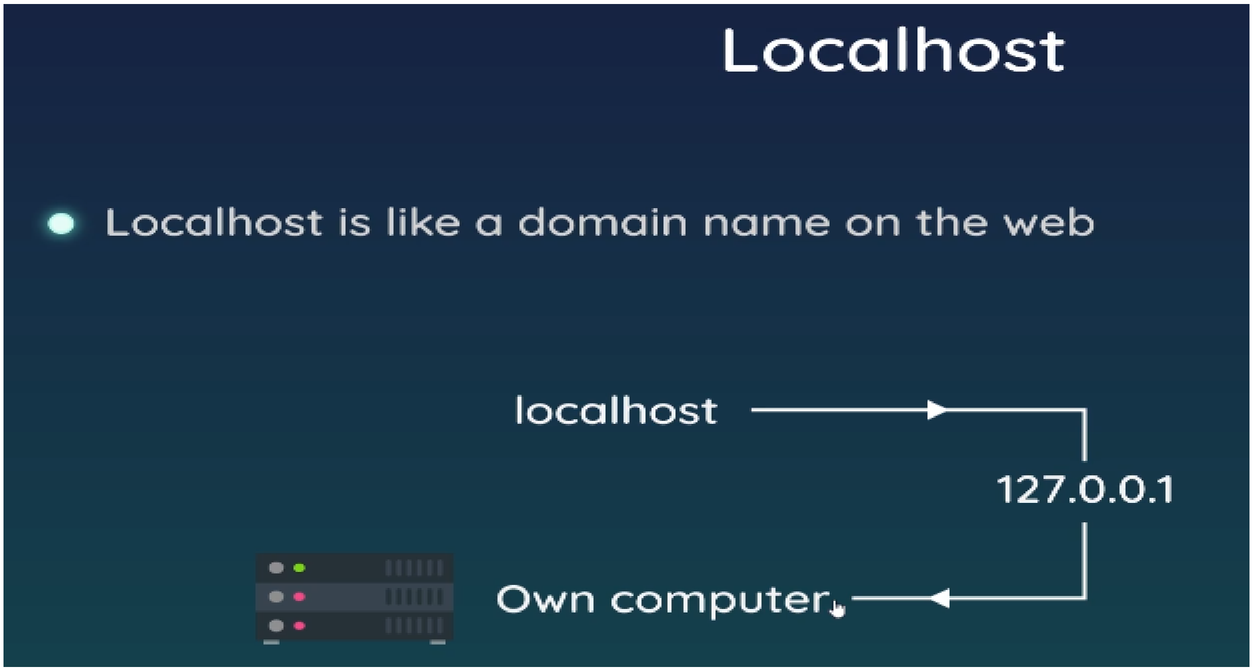
```
const http = require('http');

const server = http.createServer((req, res) =>{
    console.log('request made');
});

server.listen(3000, 'localhost', () =>{
    console.log('listening for requests on port 3000');
});
```

O/P:

```
PS C:\Users\gudiw\Documents\tuts\node-crash-course> node server
listening for requests on port 3000
request made
```

→Here request made is executed when we type **localhost:3000** in browser

LocalHost & Port Numbers



Localhost

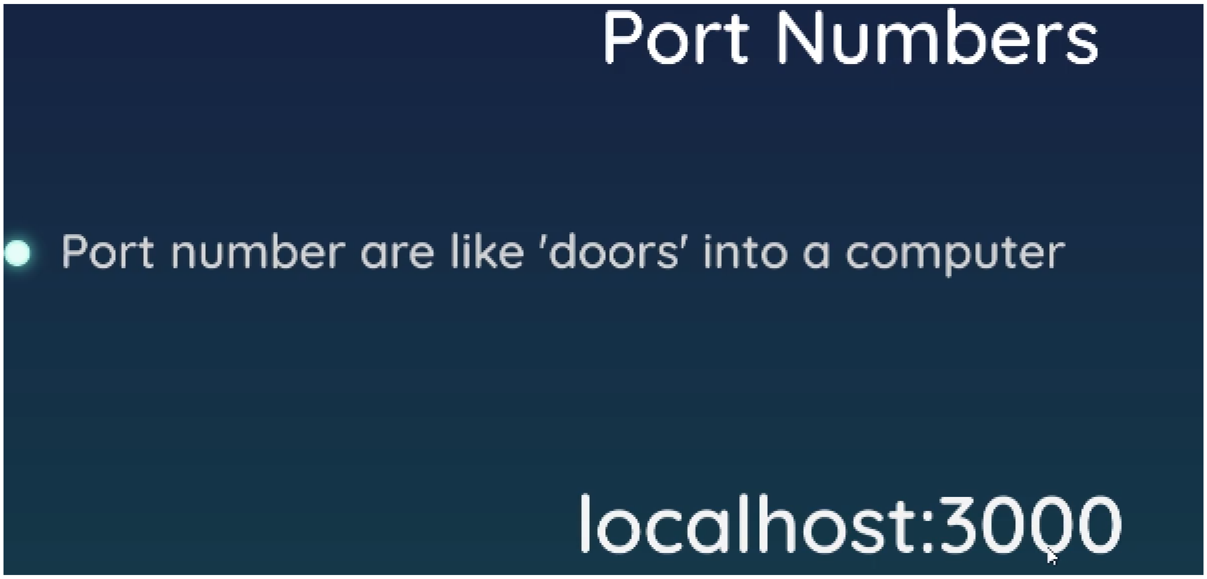
- Localhost is like a domain name on the web

localhost

127.0.0.1

Own computer

→Here Own computer as “host”



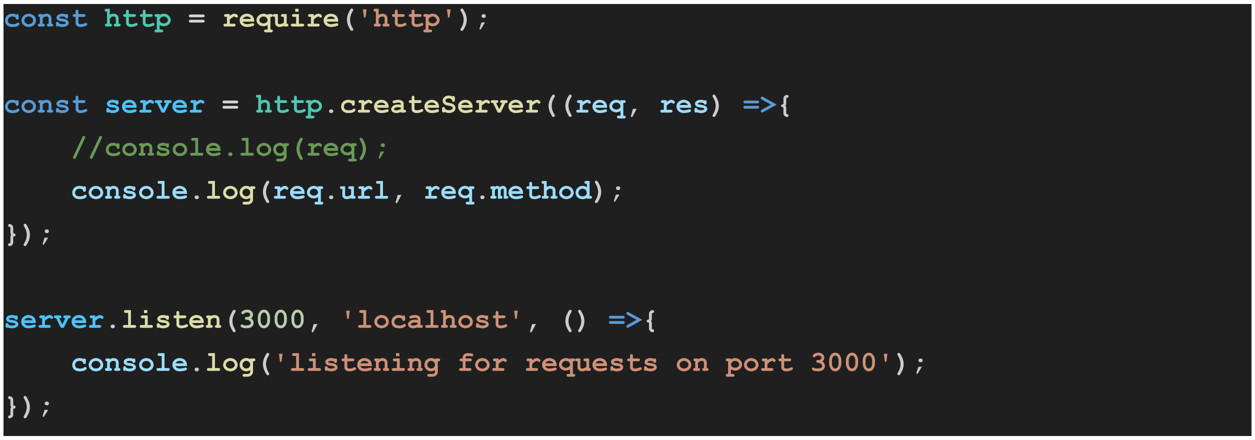
Port Numbers

- Port number are like 'doors' into a computer

localhost:3000

Lesson 4

server.js

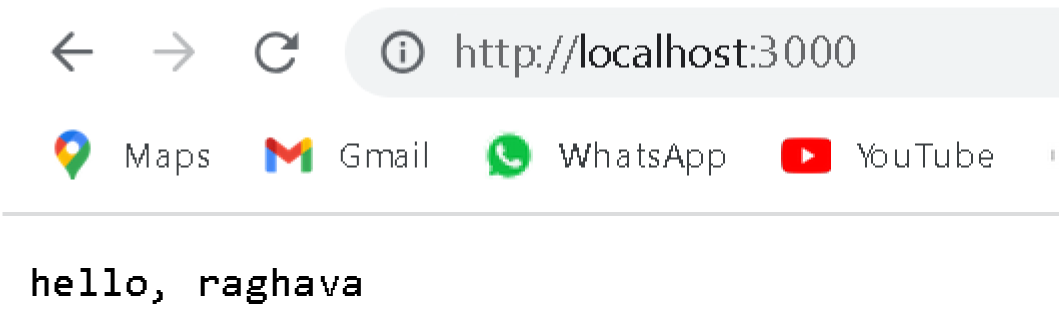


```
const http = require('http');

const server = http.createServer((req, res) =>{
    //console.log(req);
    console.log(req.url, req.method);
});

server.listen(3000, 'localhost', () =>{
    console.log('listening for requests on port 3000');
});
```

O/P:

```
PS C:\Users\gudiw\Documents\tuts\node-crash-course> node server
listening for requests on port 3000
/ GET
/about GET
```

The Response Object



http://localhost:3000

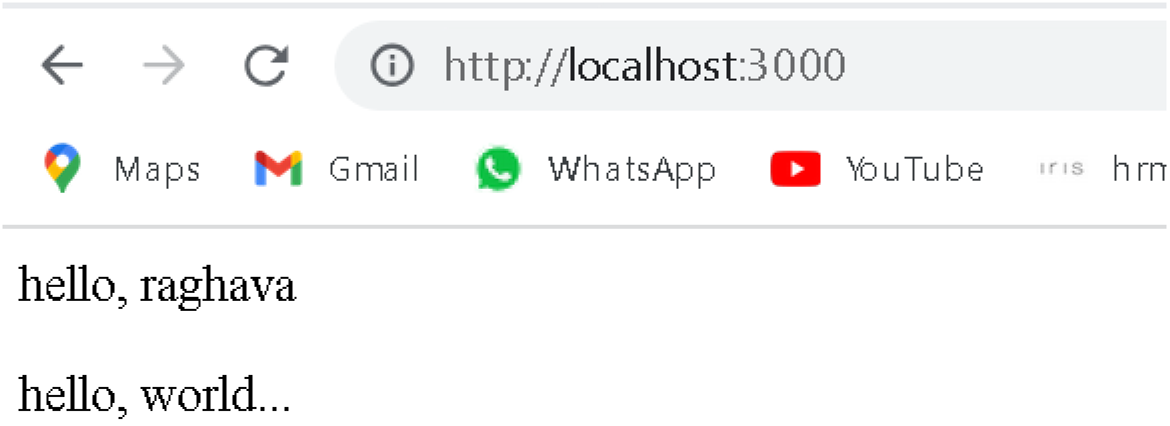
```
hello, raghava
```

server.js

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

const server = http.createServer((req, res) =>{
    //console.log(req);
    console.log(req.url, req.method);

    //set header content type
    // res.setHeader('Content-Type', 'text/plain');

    // res.write('hello, raghava');

    res.setHeader('Content-Type', 'text/html');

    res.write('<head><link rel="styleseet" href="#"></head>')
    res.write('<p>hello, raghava</p>');
    res.write('<p>hello, world...</p>');
    res.end();

});

server.listen(3000, 'localhost', () =>{
    console.log('listening for requests on port 3000');
});
```

O/P:

```
PS C:\Users\gudiw\Documents\tuts\node-crash-course> node server
listening for requests on port 3000
/ GET
/favicon.ico GET
```



hello, raghava

hello, world...

Returning Html Pages

server.js

```
const http = require('http');
const fs = require('fs')

const server = http.createServer((req, res) =>{

    console.log(req.url, req.method);

    // set header content type
    res.setHeader('Content-Type', 'text/html');

    // send an html file
    fs.readFile('./views/index.html', (err, data) =>{
      if(err){
        console.log(err);
        res.end();
      } else{
       //res.write(data);
        res.end(data);
      }
    })
});
```

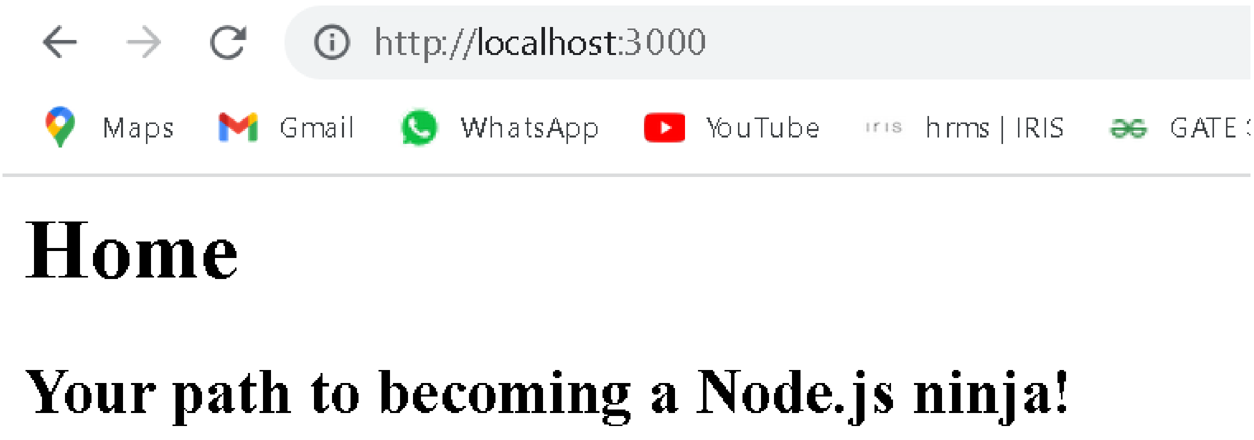
```
server.listen(3000, 'localhost', () =>{
    console.log('listening for requests on port 3000');
});
```

views→idex.html

index.html

```
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Node.js Crash Course</title>
</head>
<body>
  <h1>Home</h1>
  <h2>Your path to becoming a Node.js ninja!</h2>
</body>
</html>
```

O/P:



Home

Your path to becoming a Node.js ninja!

PS C:\Users\gudiw\Documents\tuts\node-crash-course> **node server**

listening for requests on port 3000
/ GET
/favicon.ico GET

Basic Routing

server.js

```
const http = require('http');
const fs = require('fs')

const server = http.createServer((req, res) =>{

    console.log(req.url, req.method);

    // set header content type
    res.setHeader('Content-Type', 'text/html');

    let path = './views/';

    switch(req.url){
        case '/':
            path += 'index.html';
            break;
        case '/about':
            path += 'about.html';
            break;
        default:
            path += '404.html';
            break;
    }

    // send an html file
    fs.readFile(path, (err, data) =>{
      if(err){
        console.log(err);
        res.end();
      } else{
        //res.write(data);
        res.end(data);
      }
    })
});

server.listen(3000, 'localhost', () =>{
    console.log('listening for requests on port 3000');
});
```

Index.html →already present in above

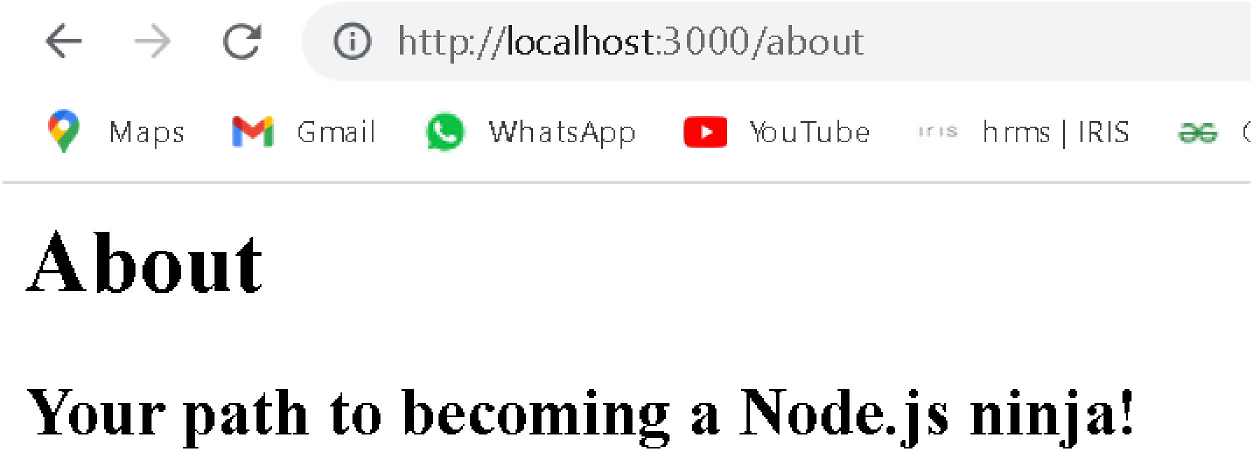
about.html

```
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Node.js Crash Course</title>
</head>
<body>
  <h1>About</h1>
  <h2>Your path to becoming a Node.js ninja!</h2>
</body>
</html>
```

404.html

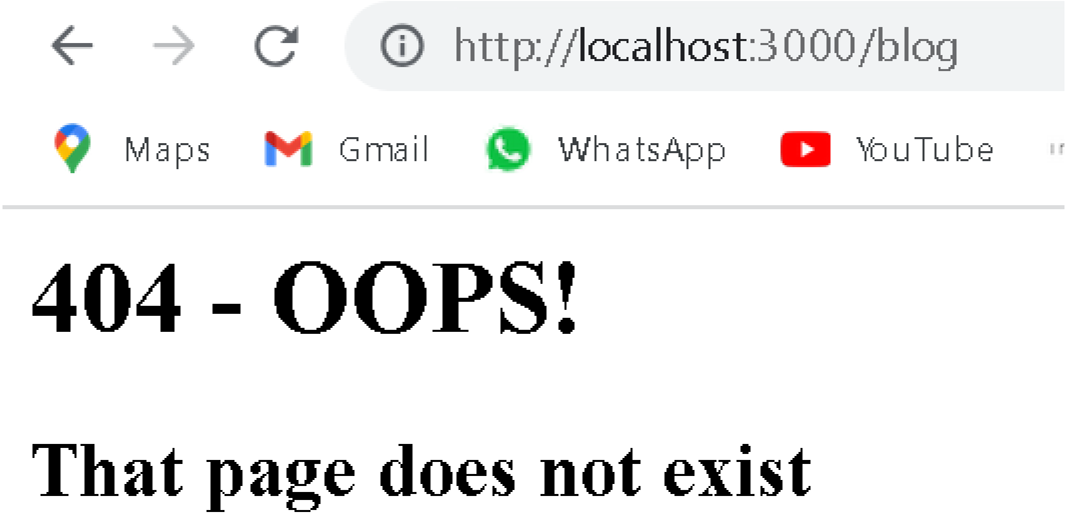
```
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Node.js Crash Course</title>
</head>
<body>
  <h1>404 - OOPS!</h1>
  <h2>That page does not exist</h2>
</body>
</html>
```

O/P:



About

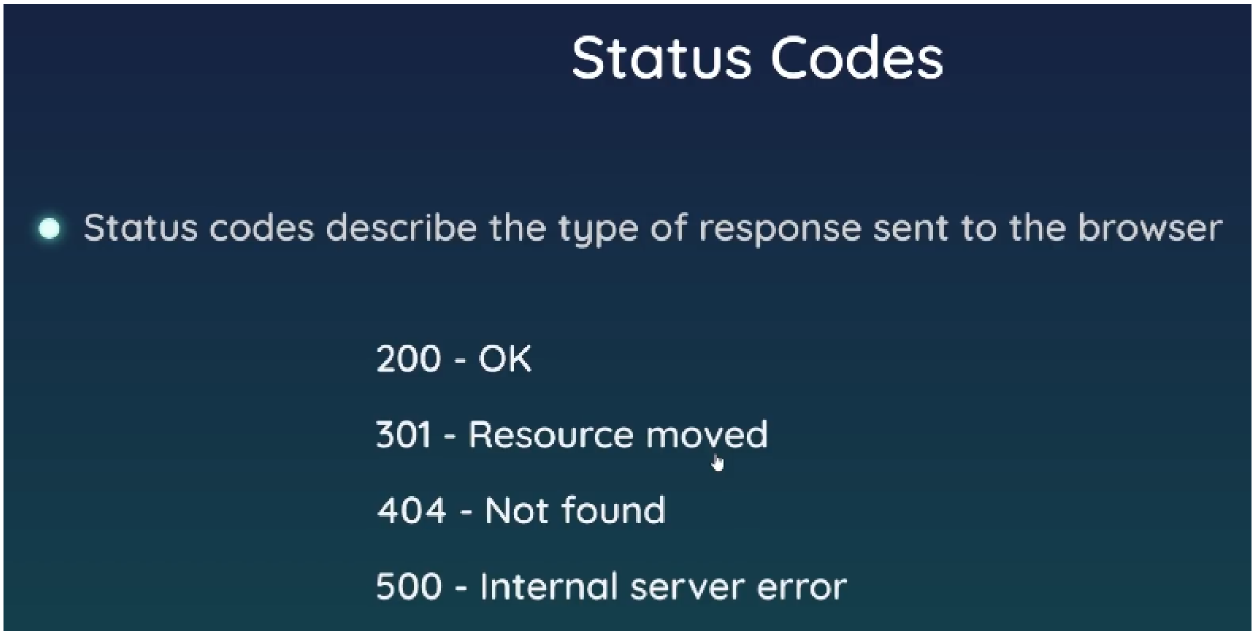
Your path to becoming a Node.js ninja!



404 - OOPS!

That page does not exist

Status Code



Status Codes

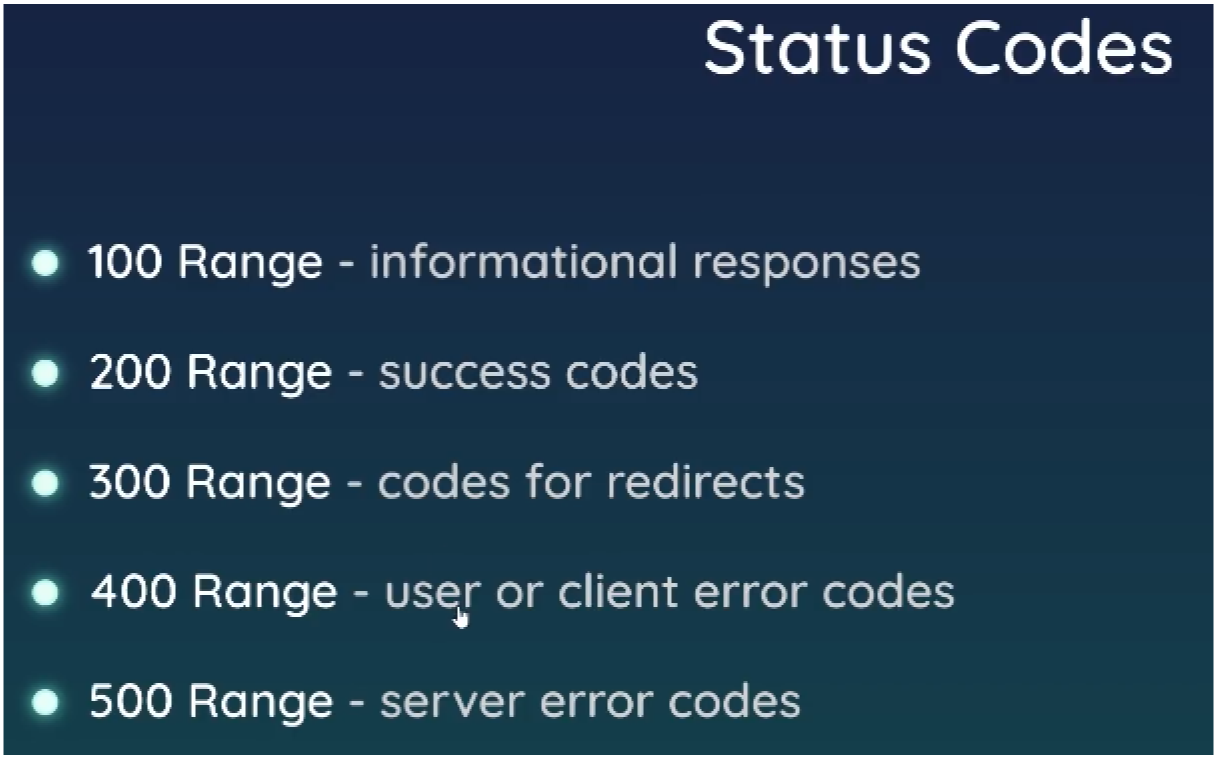
- Status codes describe the type of response sent to the browser

200 - OK

301 - Resource moved

404 - Not found

500 - Internal server error



Status Codes

- 100 Range - informational responses
- 200 Range - success codes
- 300 Range - codes for redirects
- 400 Range - user or client error codes
- 500 Range - server error codes

server.js

```
const http = require('http');
const fs = require('fs')

const server = http.createServer((req, res) =>{

  console.log(req.url, req.method);

  // set header content type
  res.setHeader('Content-Type', 'text/html');

  let path = './views/';

  switch(req.url){
    case '/':
      path += 'index.html';
      res.statusCode = 200;
      break;
    case '/about':
      path += 'about.html';
      res.statusCode = 200;
      break;
    default:
      path += '404.html';
      res.statusCode = 404 ;
      break;
  }

  // send an html file
  fs.readFile(path, (err, data) =>{
   if(err){
    console.log(err);
    res.end();
   } else{
    //res.write(data);
    res.end(data);
   }
  })
});
```

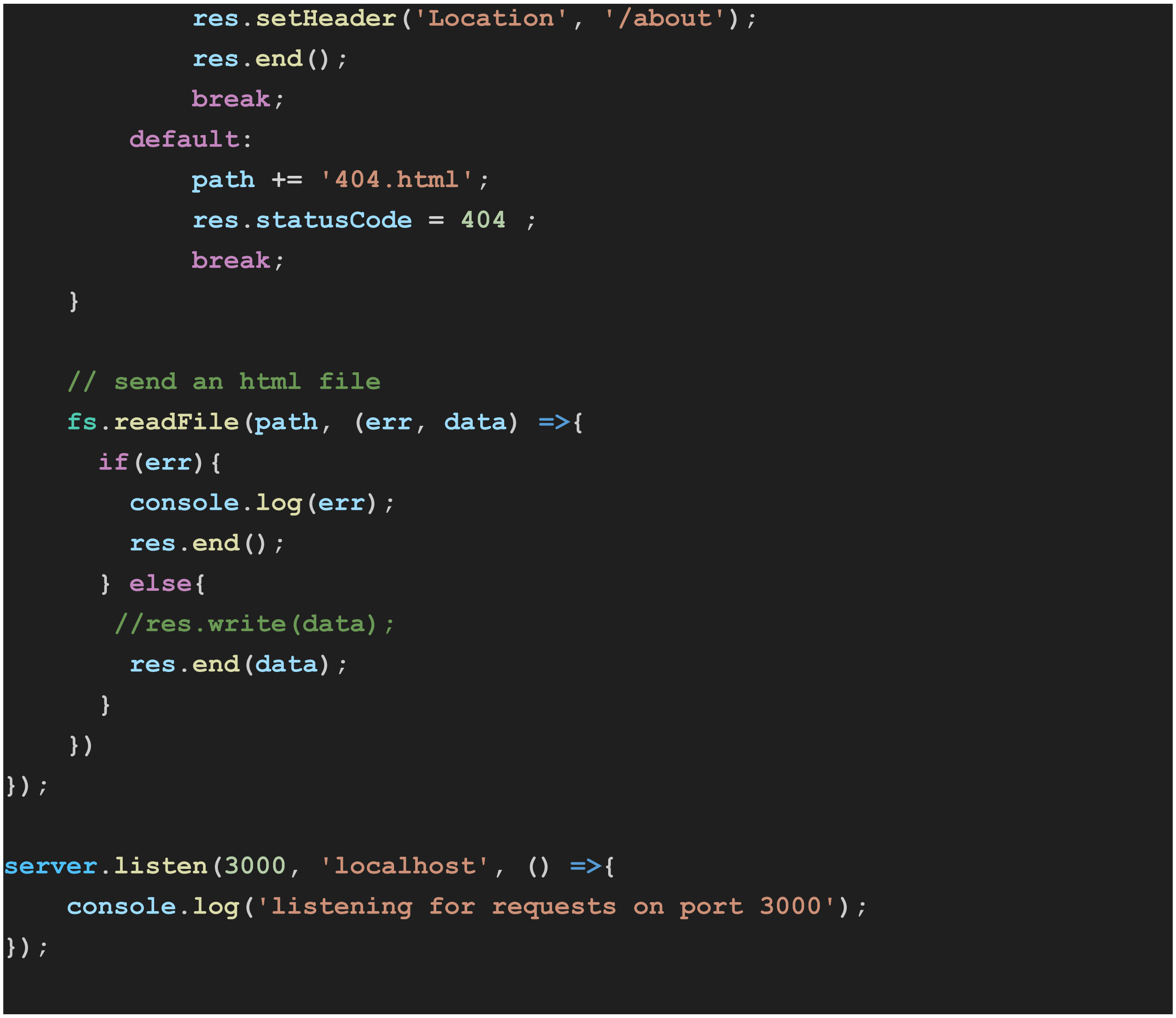
```
server.listen(3000, 'localhost', () =>{
    console.log('listening for requests on port 3000');
});
```

Redirects

In the scenario →**/about-me** is redirect to **/about**

server.js

```
const http = require('http');
const fs = require('fs')

const server = http.createServer((req, res) =>{

    console.log(req.url, req.method);

    // set header content type
    res.setHeader('Content-Type', 'text/html');

    let path = './views/';

    switch(req.url){
        case '/':
            path += 'index.html';
            res.statusCode = 200;
            break;
        case '/about':
            path += 'about.html';
            res.statusCode = 200;
            break;
        case '/about-me':
            res.statusCode = 301;   //resources moved
```



```
            res.setHeader('Location', '/about');
            res.end();
            break;
        default:
            path += '404.html';
            res.statusCode = 404 ;
            break;
    }

    // send an html file
    fs.readFile(path, (err, data) =>{
      if(err){
        console.log(err);
        res.end();
      } else{
       //res.write(data);
        res.end(data);
      }
    })
});

server.listen(3000, 'localhost', () =>{
    console.log('listening for requests on port 3000');
});
```

Lesson 5

NPM(Node Package Manager)

→**https://www.npmjs.com/**

Installing Packages Globally

```
→PS C:\Users\gudiw\Documents\tuts\node-crash-course> npm install -g nodemon
```

If any change is made in script everytime we need to make restart and run again for that purpose we install **nodemon**

```
→PS C:\Users\gudiw\Documents\tuts\node-crash-course> nodemon server
nodemon : File C:\Users\gudiw\AppData\Roaming\npm\nodemon.ps1 cannot be loaded because running scripts is disabled on this
system. For more information, see about_Execution_Policies at
https:/go.microsoft.com/fwlink/?LinkID=135170.
At line:1 char:1
+ nodemon server
+ ~~~~~~~
    + CategoryInfo          : SecurityError: (:) [], PSSecurityException
    + FullyQualifiedErrorId : UnauthorizedAccess
```

While execution “nodemon server” it is getting then bypass the error

```
→PS C:\Users\gudiw\Documents\tuts\node-crash-course> powershell -ExecutionPolicy Bypass -Command "nodemon server"
```

Packages.json File

→The packages required for the project

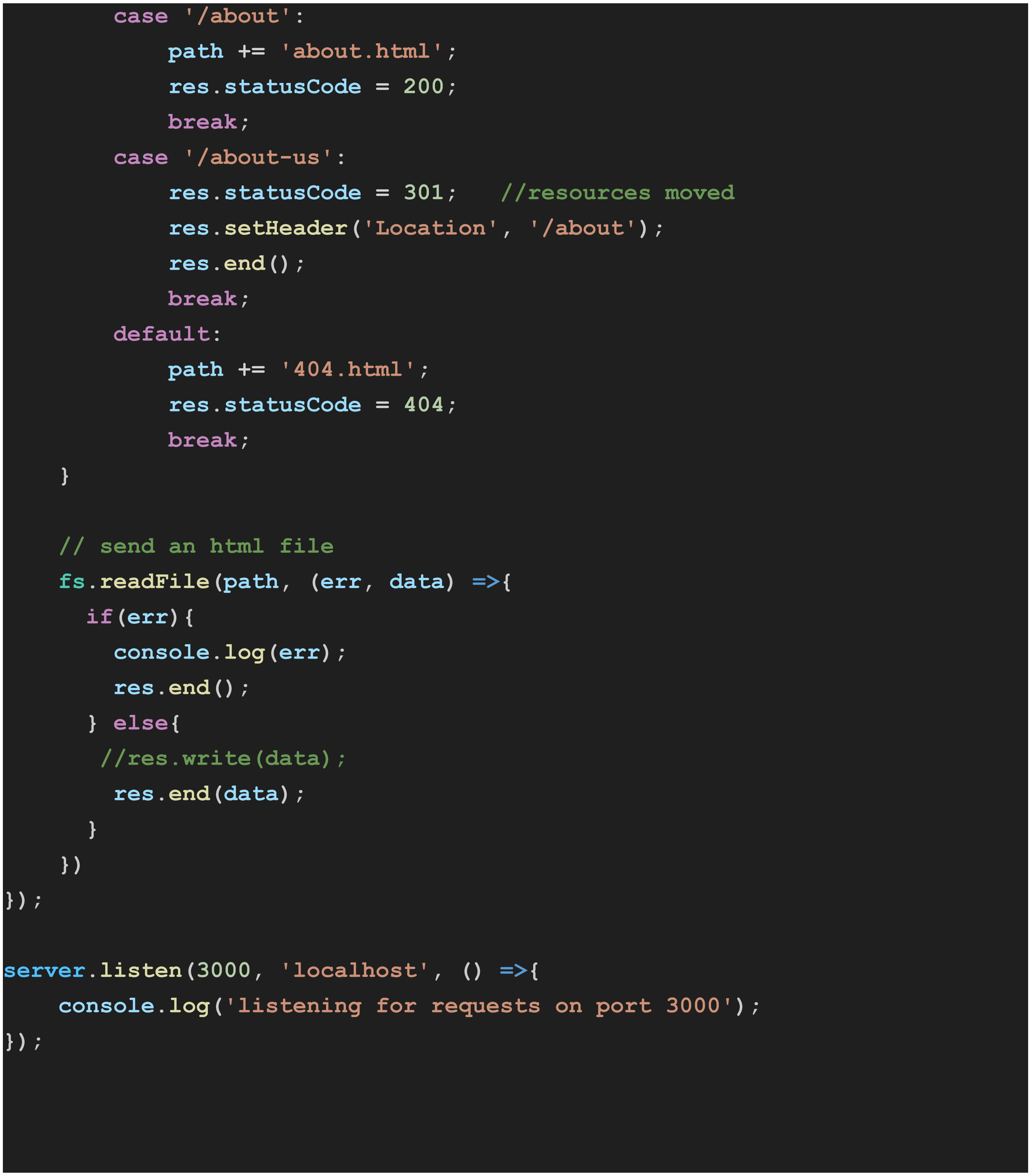
→PS C:\Users\gudiw\Documents\tuts\node-crash-course> **npm init**

PS C:\Users\gudiw\Documents\tuts\node-crash-course> **npm install lodash**

→If we install local packages it is present in **node_modules**

server.js

```
const http = require('http');
const fs = require('fs')
const _ = require('lodash');

const server = http.createServer((req, res) =>{

    //lodash
    const num = _.random(0, 20);

    console.log(num);

    const greet = _.once(() => {
        console.log('hello');
    });

    greet();
    greet();

    // set header content type
    res.setHeader('Content-Type', 'text/html');

    let path = './views/';

    switch(req.url){
        case '/':
            path += 'index.html';
            res.statusCode = 200;
            break;
```



```
        case '/about':
            path += 'about.html';
            res.statusCode = 200;
            break;
        case '/about-us':
            res.statusCode = 301;   //resources moved
            res.setHeader('Location', '/about');
            res.end();
            break;
        default:
            path += '404.html';
            res.statusCode = 404;
            break;
    }

    // send an html file
    fs.readFile(path, (err, data) =>{
      if(err){
        console.log(err);
        res.end();
      } else{
       //res.write(data);
        res.end(data);
      }
    })
});

server.listen(3000, 'localhost', () =>{
    console.log('listening for requests on port 3000');
});
```

O/P:

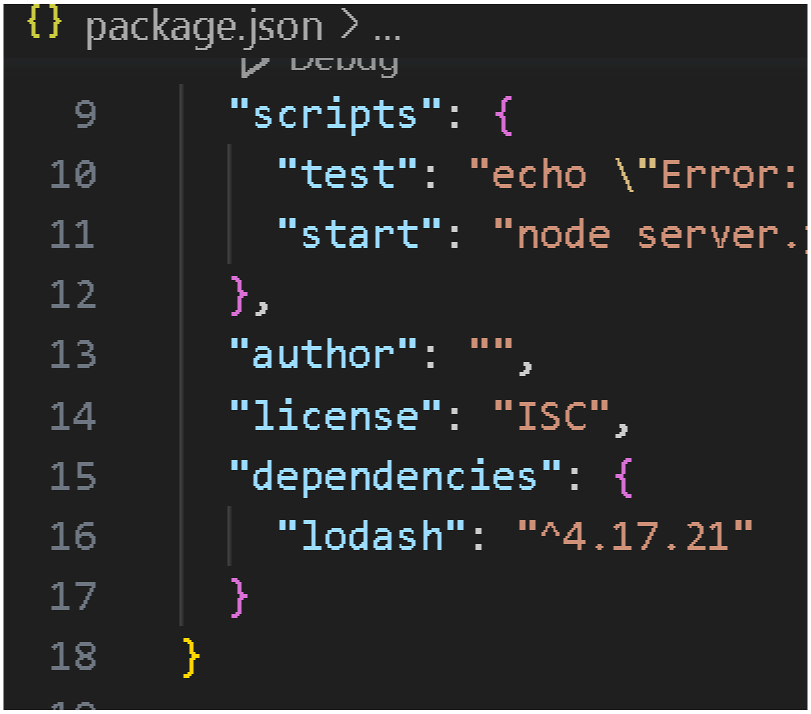
PS C:\Users\gudiw\Documents\tuts\node-crash-course> **powershell -ExecutionPolicy Bypass -Command "nodemon server"**
[nodemon] 2.0.22

```
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node server.js`
listening for requests on port 3000
[nodemon] restarting due to changes...
[nodemon] starting `node server.js`
listening for requests on port 3000
16
20
3
4
[nodemon] restarting due to changes...
[nodemon] starting `node server.js`
listening for requests on port 3000
17
hello
10
hello
```

→hello is only once it is restricted

Dependencies

→if the packages downloaded will be deleted then we are unable to run the code then **npm install** →This will give all the required packages.



```
  "scripts": {
    "test": "echo \"Error:
    "start": "node server.
  },
  "author": "",
  "license": "ISC",
  "dependencies": {
    "lodash": "^4.17.21"
  }
}
```

Lesson 6

Express Apps

Express is a popular web application framework for Node.js that simplifies the process of building web applications and APIs. It provides a set of robust features and tools that help developers manage various aspects of web development more easily.

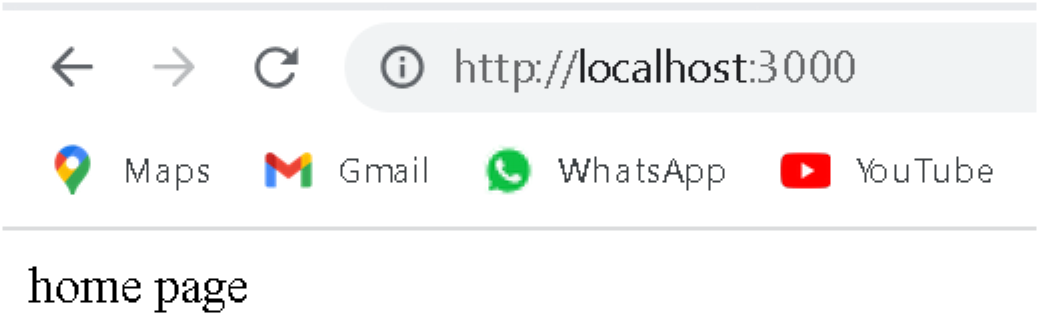
PS C:\Users\gudiw\Documents\tuts\node-crash-course> **npm install express**

Creating an Express APP

express.js

```
const express = require('express');

// express app
const app = express();

// listen for requests
app.listen(3000);

app.get('/', (req, res)=>{

    res.send('<p>home page</p>');
});
```

O/P:



home page

Routing & HTML

app.js

```
const express = require('express');

// express app
const app = express();

// listen for requests
app.listen(3000);

app.get('/', (req, res)=>{
    //res.send('<p>home page</p>');
    res.sendFile('./views/index.html', {root: __dirname});
});

app.get('/about', (req, res)=>{
    // res.send('<p>about page</p>');
    res.sendFile('./views/about.html', {root: __dirname});
});
```

index.html

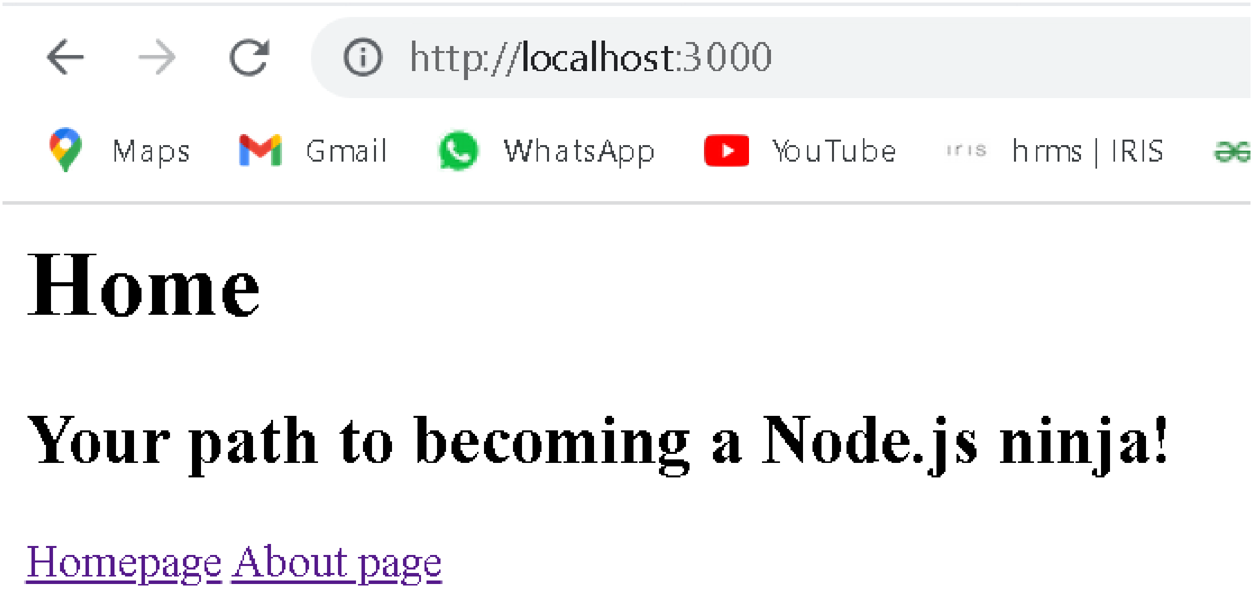
```
<html lang="en">
```

```
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Node.js Crash Course</title>
</head>
<body>
  <h1>Home</h1>
  <h2>Your path to becoming a Node.js ninja!</h2>
  <nav>
    <a href="/">Homepage</a>
    <a href="/about">About page</a>
  </nav>
</body>
</html>
```

about.html

```
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Node.js Crash Course</title>
</head>
<body>
  <h1>About</h1>
  <h2>Your path to becoming a Node.js ninja!</h2>
  <nav>
    <a href="/">Homepage</a>
    <a href="/about">About page</a>
  </nav>
</body>
</html>
```

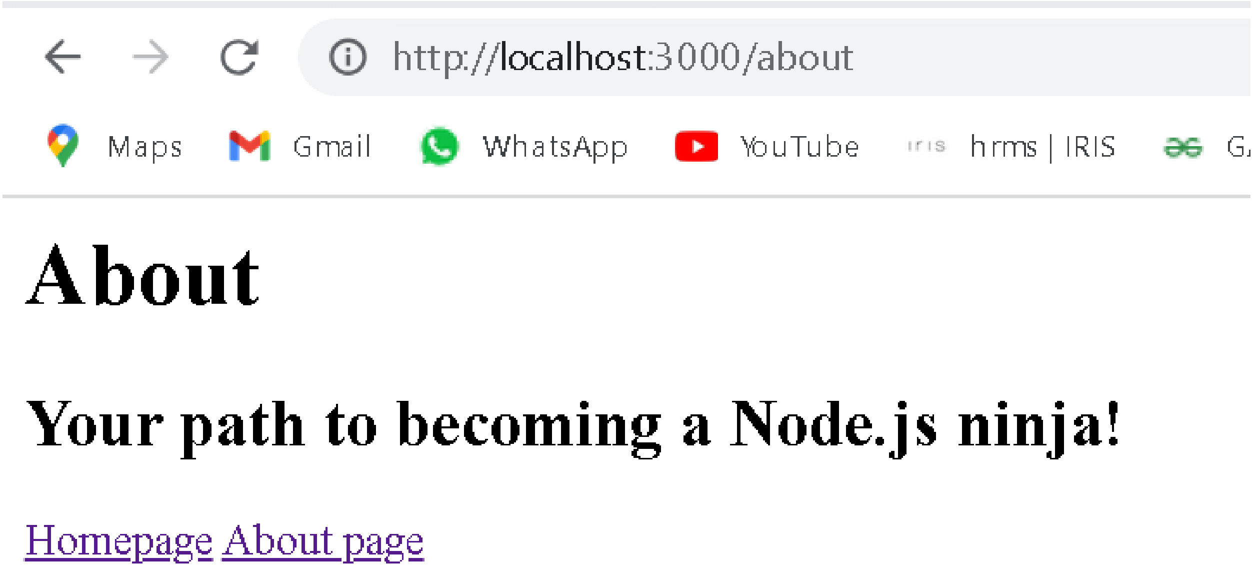
O/P:



Home

Your path to becoming a Node.js ninja!

Homepage About page



About

Your path to becoming a Node.js ninja!

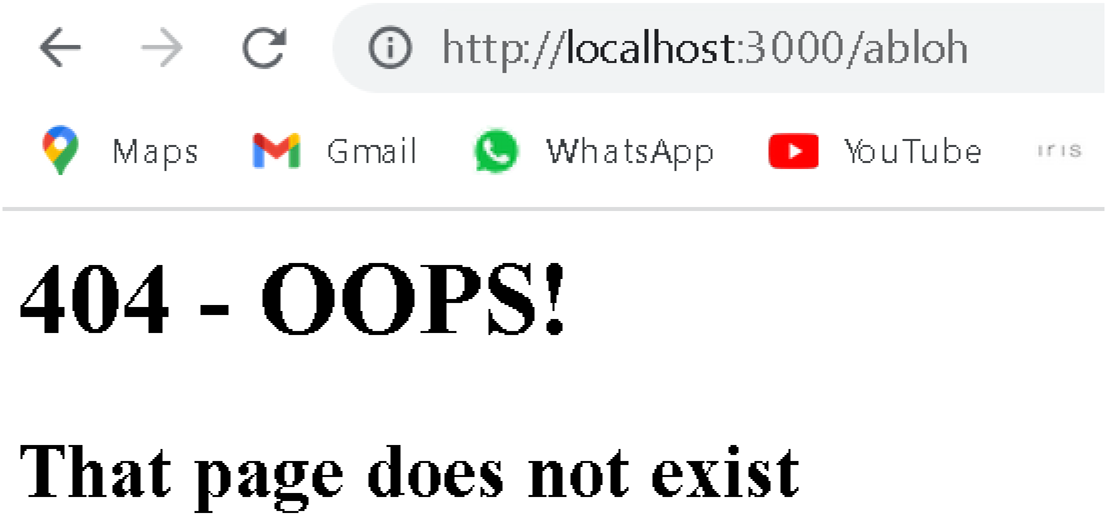
Homepage About page

Redirects & 404’s

```
const express = require('express');

// express app
const app = express();

// listen for requests
app.listen(3000);
app.get('/', (req, res)=>{
    //res.send('<p>home page</p>');
    res.sendFile('./views/index.html', {root: __dirname});
});

app.get('/about', (req, res)=>{
    // res.send('<p>about page</p>');
    res.sendFile('./views/about.html', {root: __dirname});
});

// redirects
//Here automatically sets status code
app.get('/about-us', (req, res)=>{
    res.redirect('/about');
});

// 404 page
//use is used to create middleware function
//use function fires for every single request
//position of use is also important it should be at bottom only
app.use((req, res) =>{
    res.status(404).sendFile('./views/404.html', {root: __dirname})
});
```

O/P:

→if you use **"/about-us"** it will redirects to **"/about.html"** page.



404 - OOPS!

That page does not exist

Lesson 7

View Engines

→By using view Engine we can generate dynamic html pages

EJS(Embedded JavaScript)

→Install EJS

PS C:\Users\gudiw\Documents\tuts\node-crash-course> **npm install ejs**

app.js

```
const express = require('express');

// express app
const app = express();

//register view engine
app.set('view engine', 'ejs');

// listen for requests
app.listen(3000);

app.get('/', (req, res)=>{
    res.render('index');
});

app.get('/about', (req, res)=>{
    res.render('about');
});

app.get('/blogs/create', (req, res)=>{
    res.render('create');
});
```

```
// 404 page
//use is used to create middleware function
//use function fires for every single request
//position of use is also important it should be at bottom only
app.use((req, res) =>{
    res.status(404).render('404');
});
```

index.ejs

```
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Blog Ninja</title>
</head>
<body>

    <nav>
        <div class = "site-title">
            <a href="/"><h1>Blog Ninja</h1></a>
            <p>A Net Ninja Site</p>
        </div>
        <ul>
            <li><a href="/">Blogs</a></li>
            <li><a href="/about">About</a></li>
            <li><a href="/blogs/create">New Blog</a></li>
        </ul>
    </nav>

    <div class="blogs content">
        <h2>All Blogs</h2>

    </div>

</body>
</html>
```

about.ejs

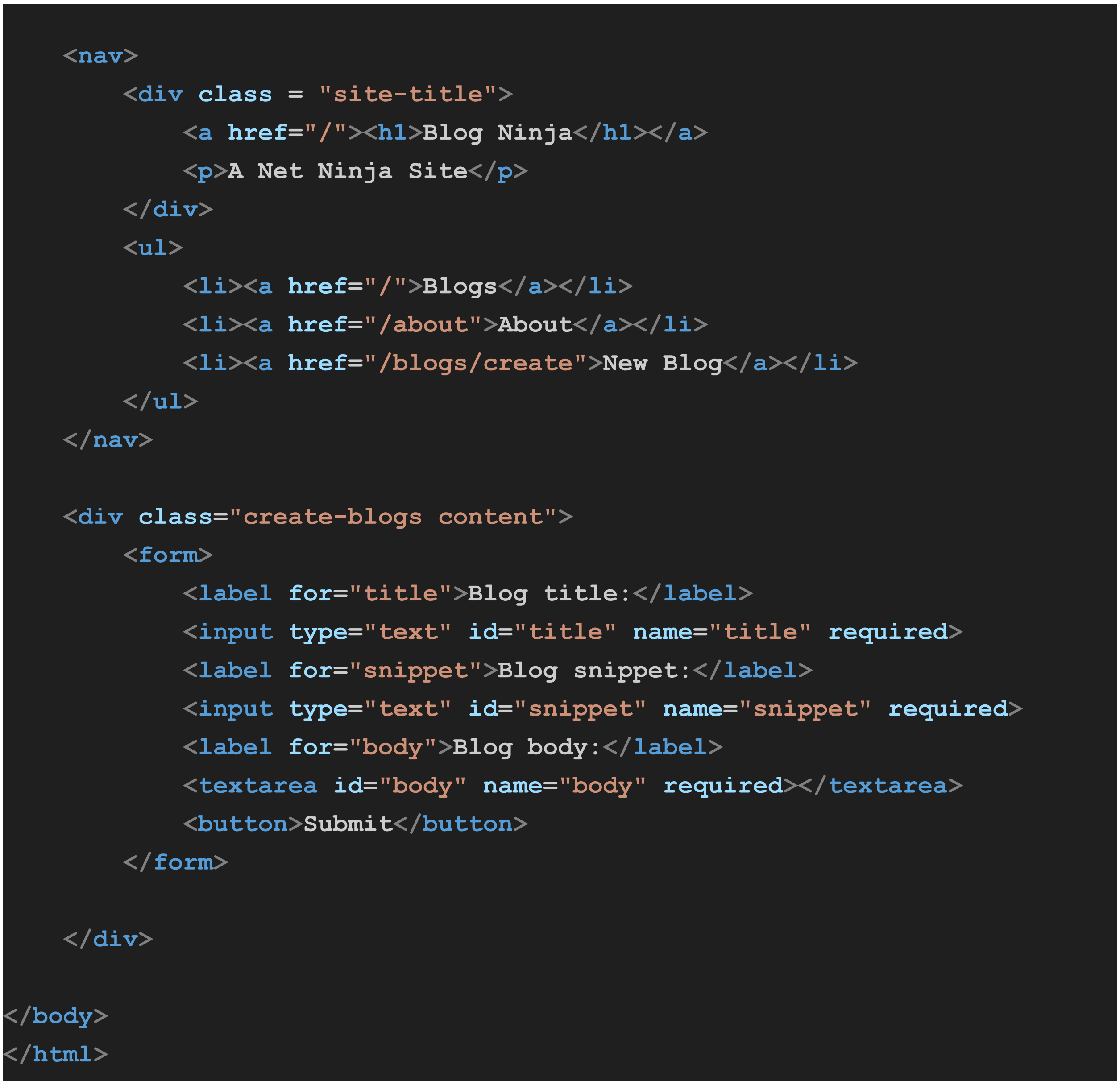
```
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Blog Ninja</title>
</head>
<body>

    <nav>
        <div class = "site-title">
            <a href="/"><h1>Blog Ninja</h1></a>
            <p>A Net Ninja Site</p>
        </div>
        <ul>
            <li><a href="/">Blogs</a></li>
            <li><a href="/about">About</a></li>
            <li><a href="/blogs/create">New Blog</a></li>
        </ul>
    </nav>

    <div class="about content">
        <h2>About Us</h2>
        <p>Lorem ipsum dolor, sit amet consectetur adipisicing elit. Quia quibusdam quaerat illo a modi tenetur ut blanditiis, illum quo consectetur recusandae excepturi natus impedit rem, tempora cum fuga quae in.</p>
        <p>Lorem ipsum dolor, sit amet consectetur adipisicing elit. Quia quibusdam quaerat illo a modi tenetur ut blanditiis, illum quo consectetur recusandae excepturi natus impedit rem, tempora cum fuga quae in.</p>
        <p>Lorem ipsum dolor, sit amet consectetur adipisicing elit. Quia quibusdam quaerat illo a modi tenetur ut blanditiis, illum quo consectetur recusandae excepturi natus impedit rem, tempora cum fuga quae in.</p>
    </div>

</body>
</html>
```

404.ejs

```
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Blog Ninja</title>
</head>
<body>

    <nav>
        <div class = "site-title">
            <a href="/"><h1>Blog Ninja</h1></a>
            <p>A Net Ninja Site</p>
        </div>
        <ul>
            <li><a href="/">Blogs</a></li>
            <li><a href="/about">About</a></li>
            <li><a href="/blogs/create">New Blog</a></li>
        </ul>
    </nav>

    <div class="not-found content">
        OOPS, page not found :)
    </div>

</body>
</html>
```

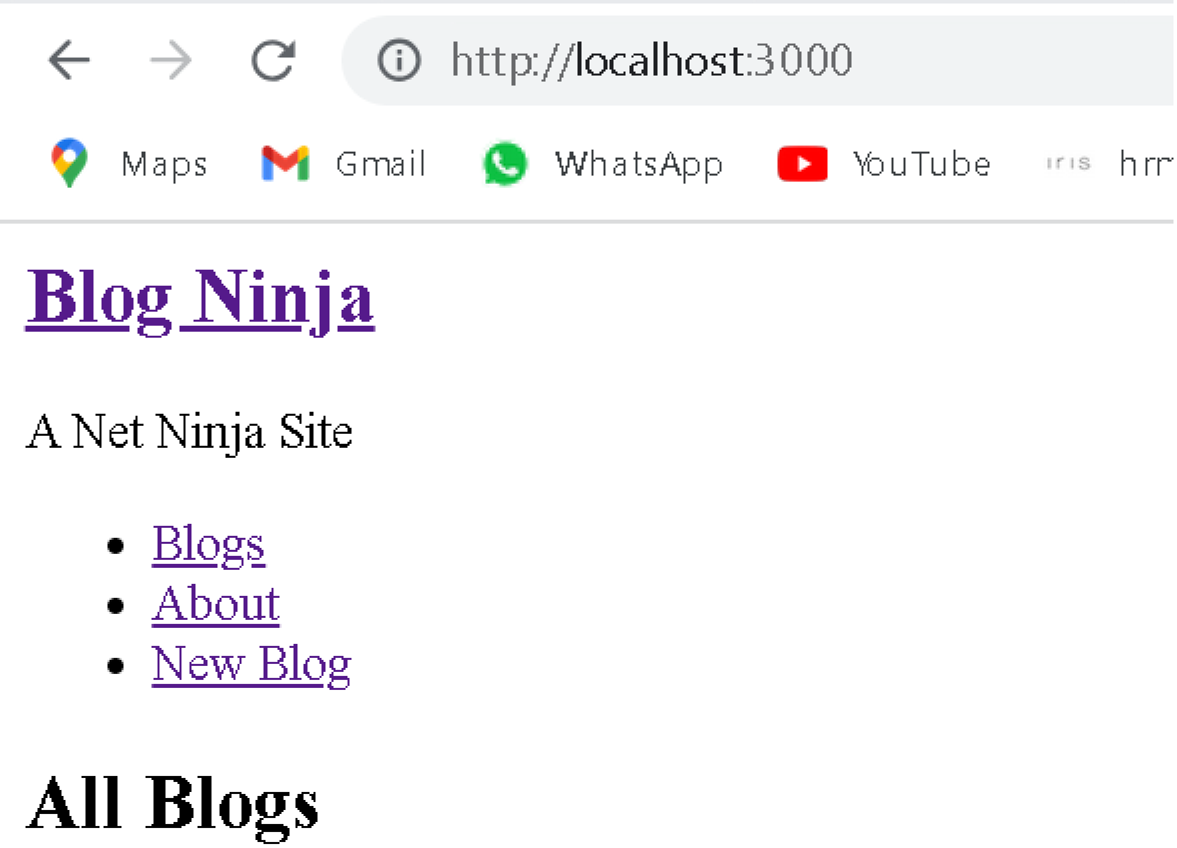
create.ejs

```
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Blog Ninja</title>
</head>
<body>
```



```html
    <nav>
        <div class = "site-title">
            <a href="/"><h1>Blog Ninja</h1></a>
            <p>A Net Ninja Site</p>
        </div>
        <ul>
            <li><a href="/">Blogs</a></li>
            <li><a href="/about">About</a></li>
            <li><a href="/blogs/create">New Blog</a></li>
        </ul>
    </nav>

    <div class="create-blogs content">
        <form>
            <label for="title">Blog title:</label>
            <input type="text" id="title" name="title" required>
            <label for="snippet">Blog snippet:</label>
            <input type="text" id="snippet" name="snippet" required>
            <label for="body">Blog body:</label>
            <textarea id="body" name="body" required></textarea>
            <button>Submit</button>
        </form>

    </div>

</body>
</html>
```

O/P:

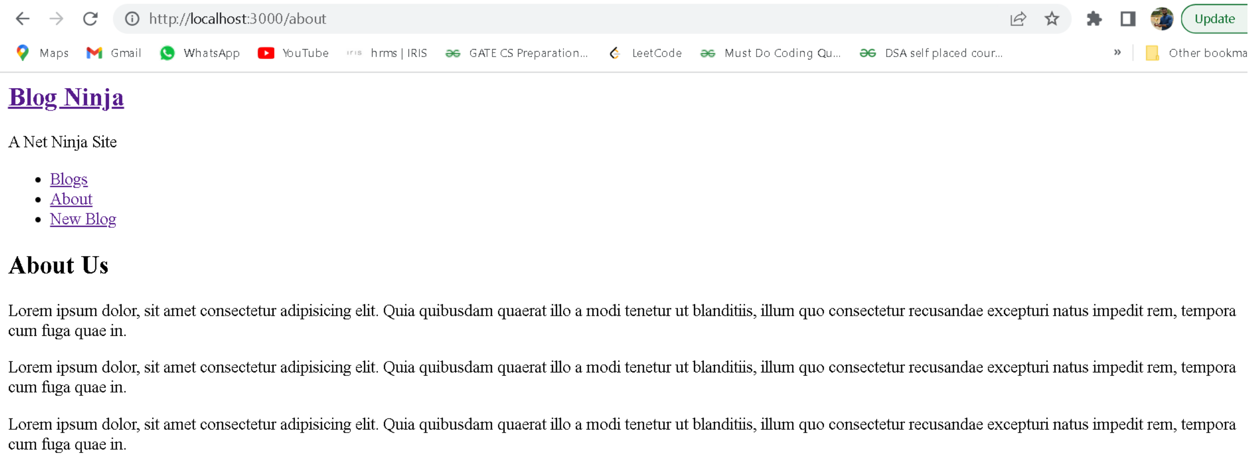


Blog Ninja

A Net Ninja Site

- Blogs
- About
- New Blog

All Blogs



Blog Ninja

A Net Ninja Site

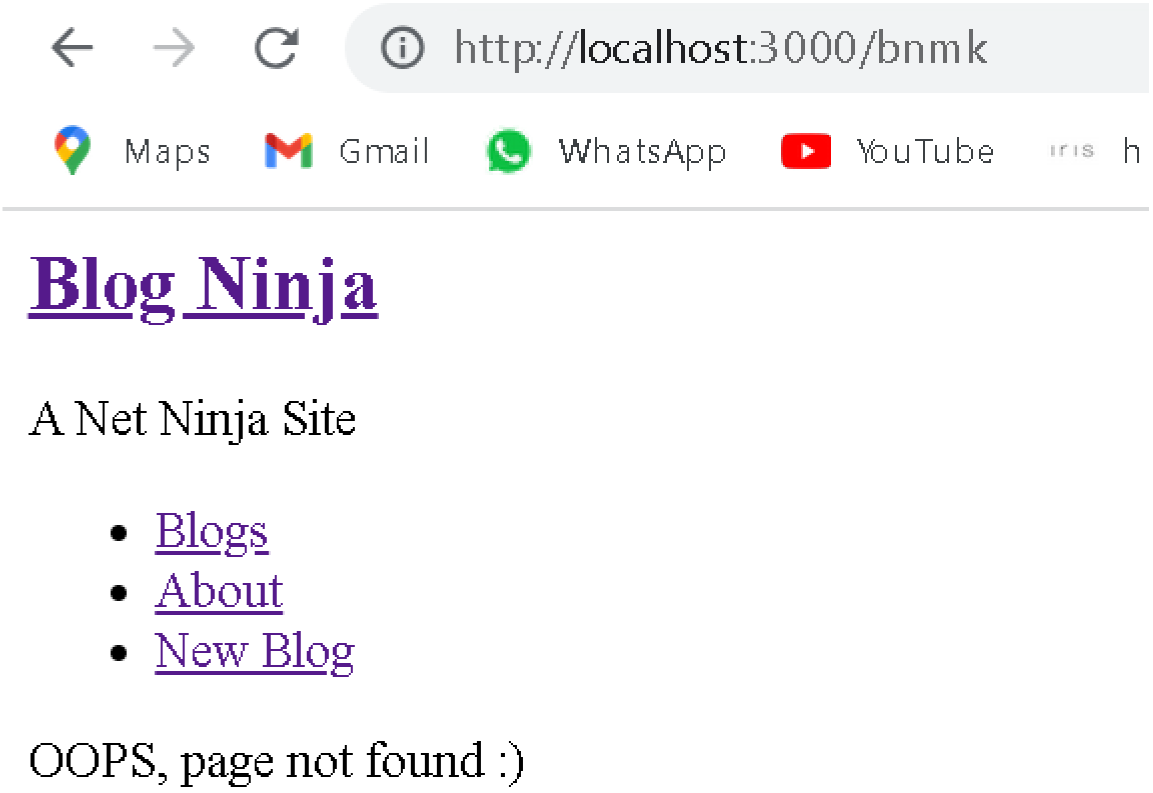
- Blogs
- About
- New Blog

About Us

Lorem ipsum dolor, sit amet consectetur adipisicing elit. Quia quibusdam quaerat illo a modi tenetur ut blanditiis, illum quo consectetur recusandae excepturi natus impedit rem, tempora cum fuga quae in.

Lorem ipsum dolor, sit amet consectetur adipisicing elit. Quia quibusdam quaerat illo a modi tenetur ut blanditiis, illum quo consectetur recusandae excepturi natus impedit rem, tempora cum fuga quae in.

Lorem ipsum dolor, sit amet consectetur adipisicing elit. Quia quibusdam quaerat illo a modi tenetur ut blanditiis, illum quo consectetur recusandae excepturi natus impedit rem, tempora cum fuga quae in.

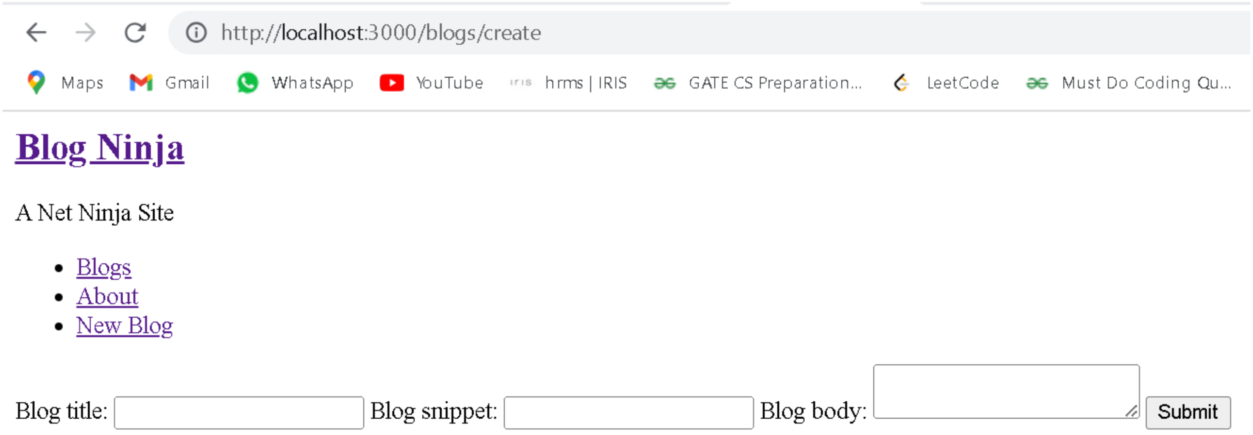


Blog Ninja

A Net Ninja Site

- Blogs
- About
- New Blog

OOPS, page not found :)



Blog Ninja

A Net Ninja Site

- Blogs
- About
- New Blog

Blog title: Blog snippet: Blog body: Submit

Passing Data into views

app.js

```
const express = require('express');

// express app
const app = express();

//register view engine
app.set('view engine', 'ejs');

// listen for requests
app.listen(3000);

app.get('/', (req, res)=>{
    const blogs = [
        {title: 'Sandy finds eggs', snippet: 'Lorem ipsum dolor sit amet
consectetur'},
        {title: 'Raghava finds stars', snippet: 'Lorem ipsum dolor sit
amet consectetur'},
        {title: 'How to defeat bowser', snippet: 'Lorem ipsum dolor sit
amet consectetur'},
    ];
    res.render('index', {title: 'Home', blogs});
});

app.get('/about', (req, res)=>{
    res.render('about', {title: 'About'});
});

app.get('/blogs/create', (req, res)=>{
    res.render('create', {title: 'Create a new Blog'});
});

// 404 page

app.use((req, res) =>{
    res.status(404).render('404', {title: '404'});
});
```

index.html

```
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Blog Ninja | <%= title %> </title>
</head>
<body>

    <nav>
        <div class = "site-title">
            <a href="/"><h1>Blog Ninja</h1></a>
            <p>A Net Ninja Site</p>
        </div>
        <ul>
            <li><a href="/">Blogs</a></li>
            <li><a href="/about">About</a></li>
            <li><a href="/blogs/create">New Blog</a></li>
        </ul>
    </nav>

    <div class="blogs content">
        <h2>All Blogs</h2>

        <% if(blogs.length > 0) { %>
            <% blogs.forEach(blog => { %>

                <h3 class="title"><%= blog.title %></h3>
                <p class="snippet"><%= blog.snippet %></p>

            <% }) %>
        <% } else { %>
            <p>There is no blogs to display...</p>
        <% } %>

    </div>

</body>
</html>
```

about.ejs

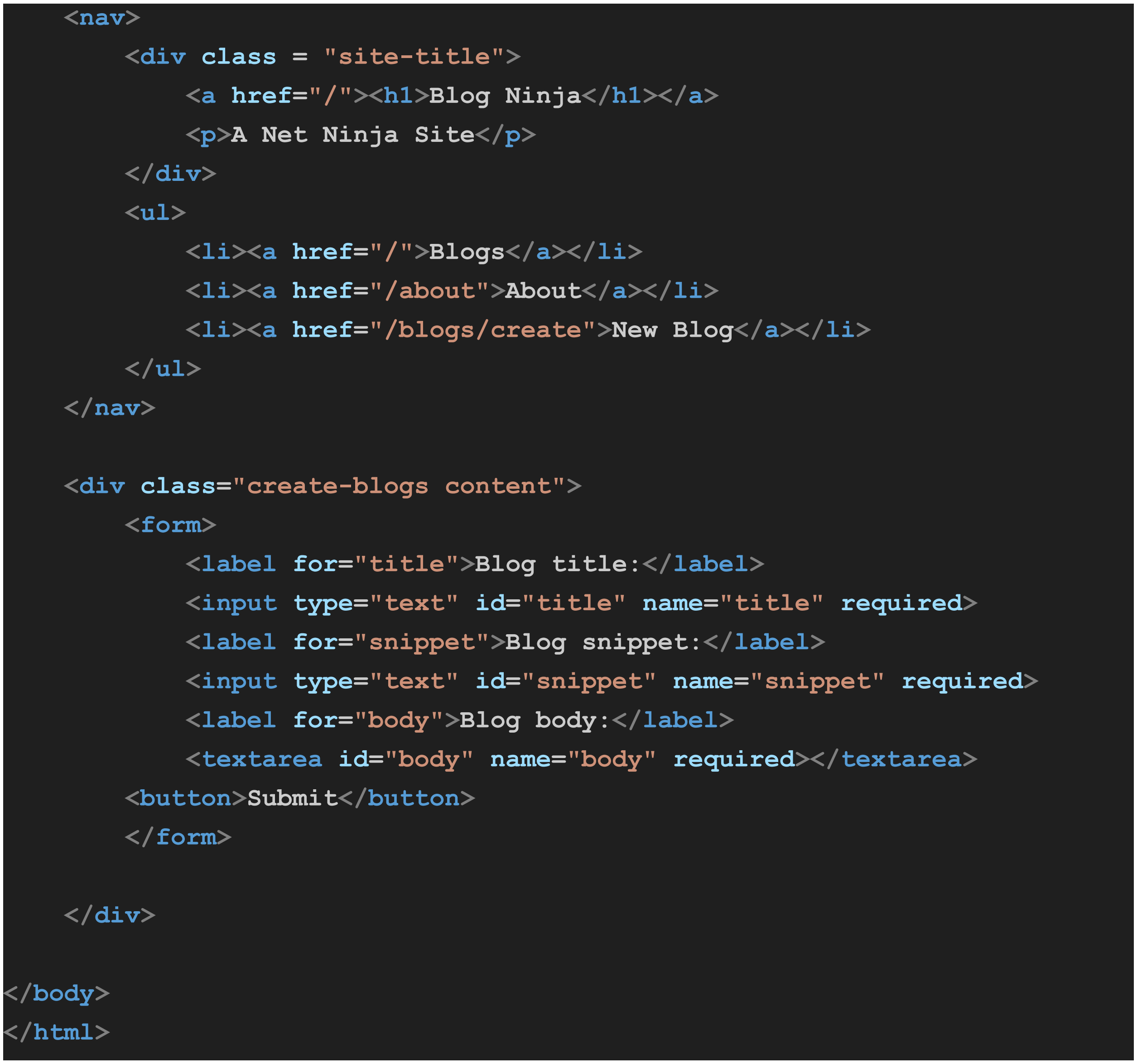
```
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Blog Ninja | <%= title %></title>
</head>
<body>

    <nav>
        <div class = "site-title">
            <a href="/"><h1>Blog Ninja</h1></a>
            <p>A Net Ninja Site</p>
        </div>
        <ul>
            <li><a href="/">Blogs</a></li>
            <li><a href="/about">About</a></li>
            <li><a href="/blogs/create">New Blog</a></li>
        </ul>
    </nav>

    <div class="about content">
        <h2>About Us</h2>
        <p>Lorem ipsum dolor, sit amet consectetur adipisicing elit. Quia
quibusdam quaerat illo a modi tenetur ut blanditiis, illum quo consectetur
recusandae excepturi natus impedit rem, tempora cum fuga quae in.</p>
        <p>Lorem ipsum dolor, sit amet consectetur adipisicing elit. Quia
quibusdam quaerat illo a modi tenetur ut blanditiis, illum quo consectetur
recusandae excepturi natus impedit rem, tempora cum fuga quae in.</p>
        <p>Lorem ipsum dolor, sit amet consectetur adipisicing elit. Quia
quibusdam quaerat illo a modi tenetur ut blanditiis, illum quo consectetur
recusandae excepturi natus impedit rem, tempora cum fuga quae in.</p>
    </div>

</body>
</html>
```

404.ejs

```
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Blog Ninja | <%= title %></title>
</head>
<body>

    <nav>
        <div class = "site-title">
            <a href="/"><h1>Blog Ninja</h1></a>
            <p>A Net Ninja Site</p>
        </div>
        <ul>
            <li><a href="/">Blogs</a></li>
            <li><a href="/about">About</a></li>
            <li><a href="/blogs/create">New Blog</a></li>
        </ul>
    </nav>

    <div class="not-found content">
        OOPS, page not found :)
    </div>

</body>
</html>
```

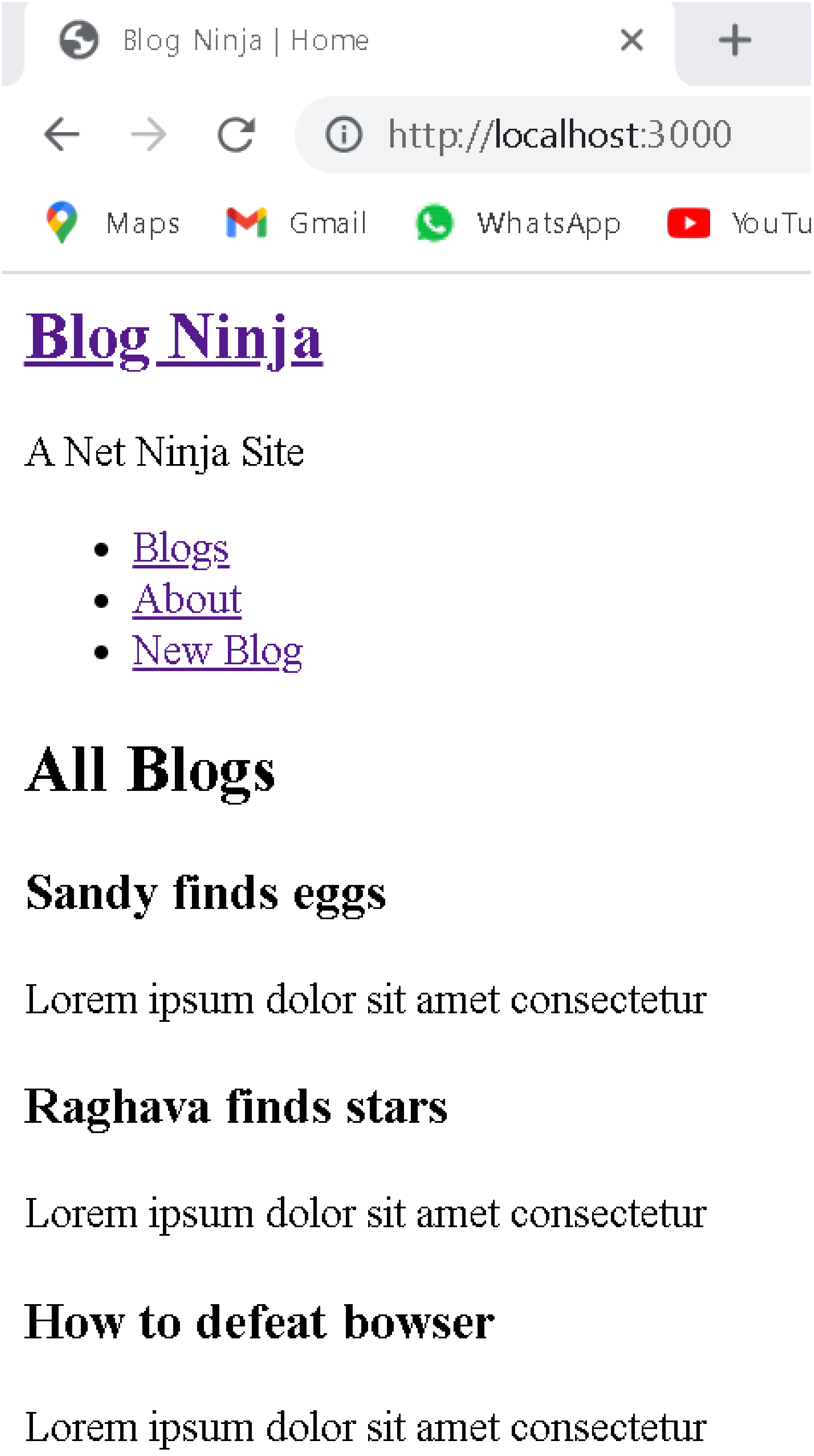
create.ejs

```
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Blog Ninja | <%= title %></title>
</head>
<body>
```



```
    <nav>
        <div class = "site-title">
            <a href="/"><h1>Blog Ninja</h1></a>
            <p>A Net Ninja Site</p>
        </div>
        <ul>
            <li><a href="/">Blogs</a></li>
            <li><a href="/about">About</a></li>
            <li><a href="/blogs/create">New Blog</a></li>
        </ul>
    </nav>

    <div class="create-blogs content">
        <form>
            <label for="title">Blog title:</label>
            <input type="text" id="title" name="title" required>
            <label for="snippet">Blog snippet:</label>
            <input type="text" id="snippet" name="snippet" required>
            <label for="body">Blog body:</label>
            <textarea id="body" name="body" required></textarea>
        <button>Submit</button>
        </form>

    </div>

</body>
</html>
```

O/P:



Blog Ninja

A Net Ninja Site

- Blogs
- About
- New Blog

All Blogs

Sandy finds eggs

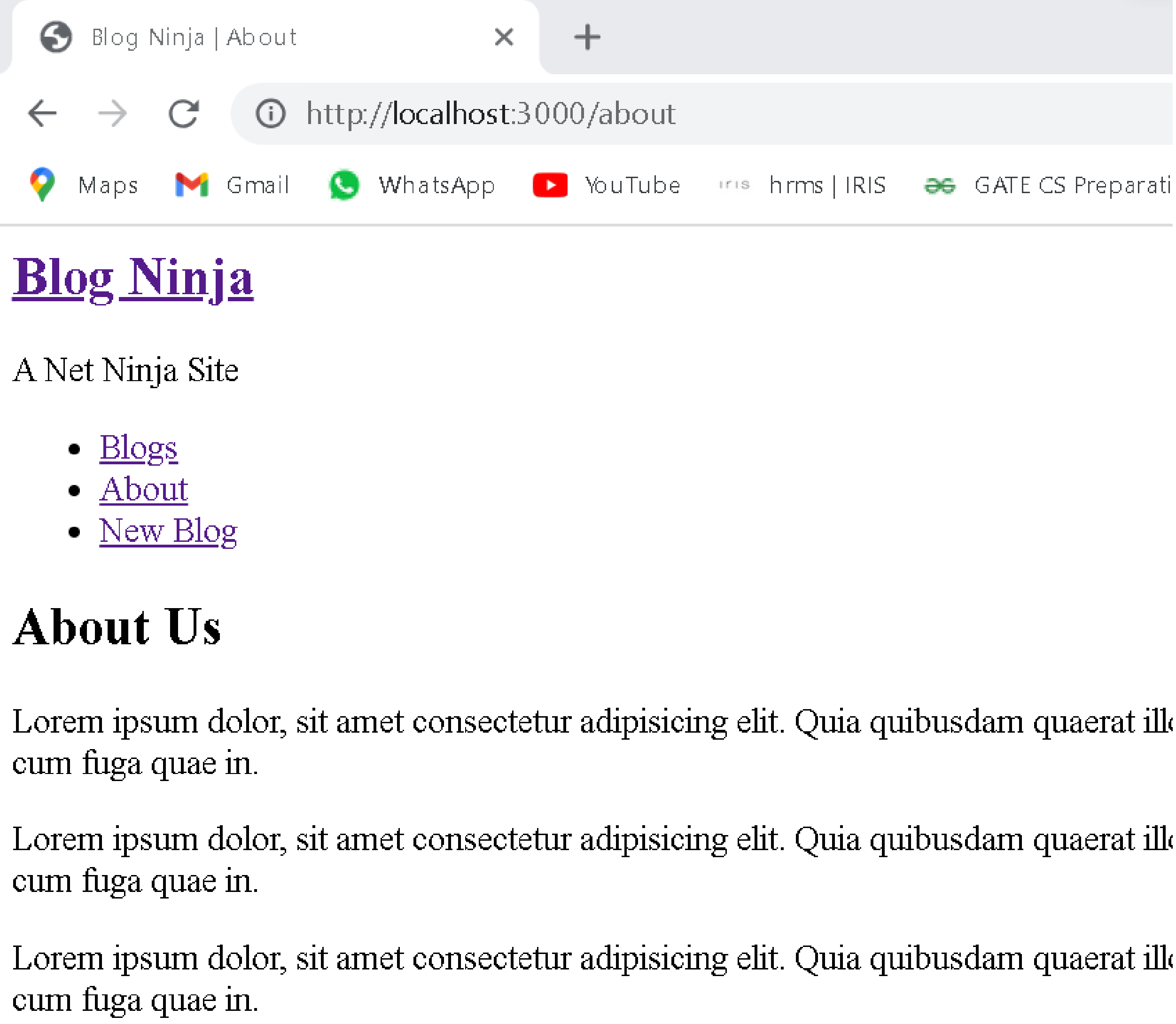
Lorem ipsum dolor sit amet consectetur

Raghava finds stars

Lorem ipsum dolor sit amet consectetur

How to defeat bowser

Lorem ipsum dolor sit amet consectetur



Blog Ninja

A Net Ninja Site

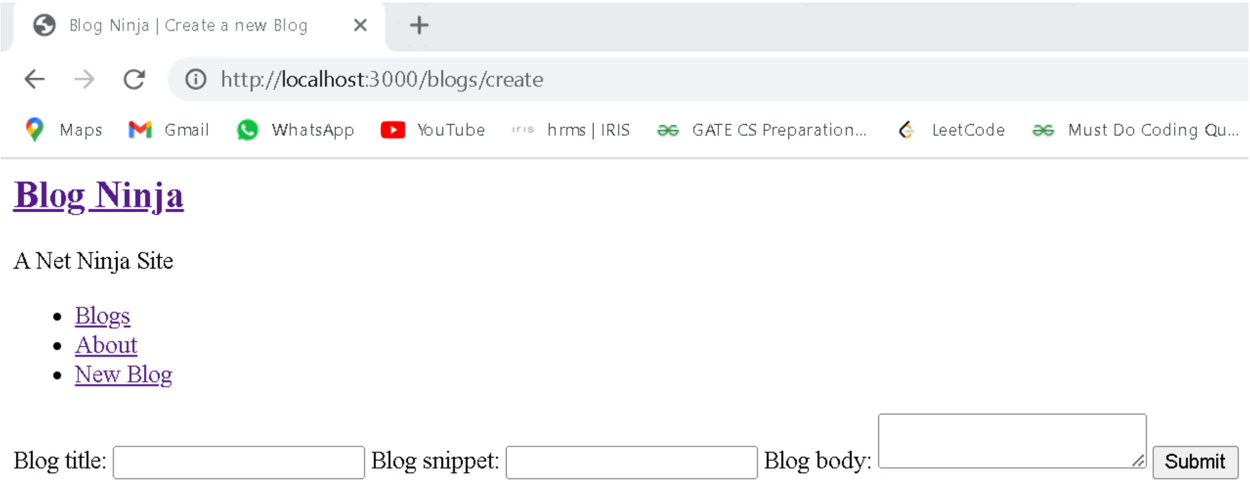
- Blogs
- About
- New Blog

About Us

Lorem ipsum dolor, sit amet consectetur adipisicing elit. Quia quibusdam quaerat ill
cum fuga quae in.

Lorem ipsum dolor, sit amet consectetur adipisicing elit. Quia quibusdam quaerat ill
cum fuga quae in.

Lorem ipsum dolor, sit amet consectetur adipisicing elit. Quia quibusdam quaerat ill
cum fuga quae in.

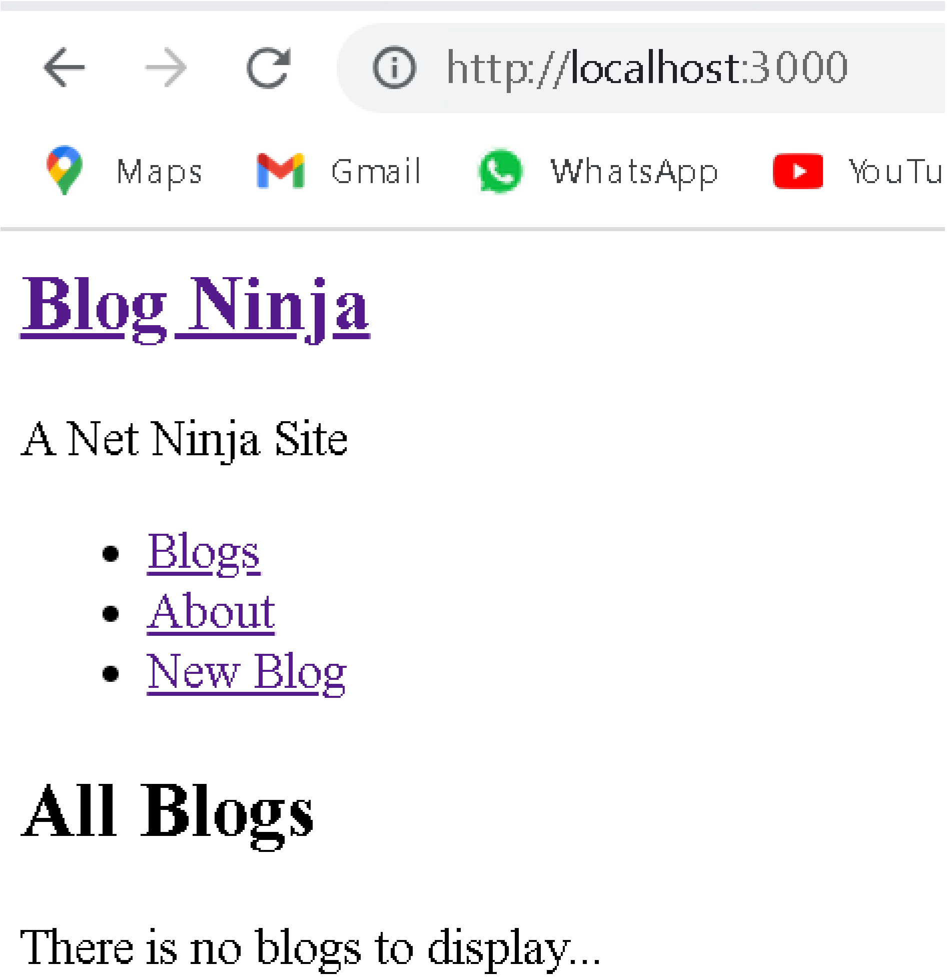


Blog Ninja

A Net Ninja Site

- Blogs
- About
- New Blog

Blog title: Blog snippet: Blog body:



Blog Ninja

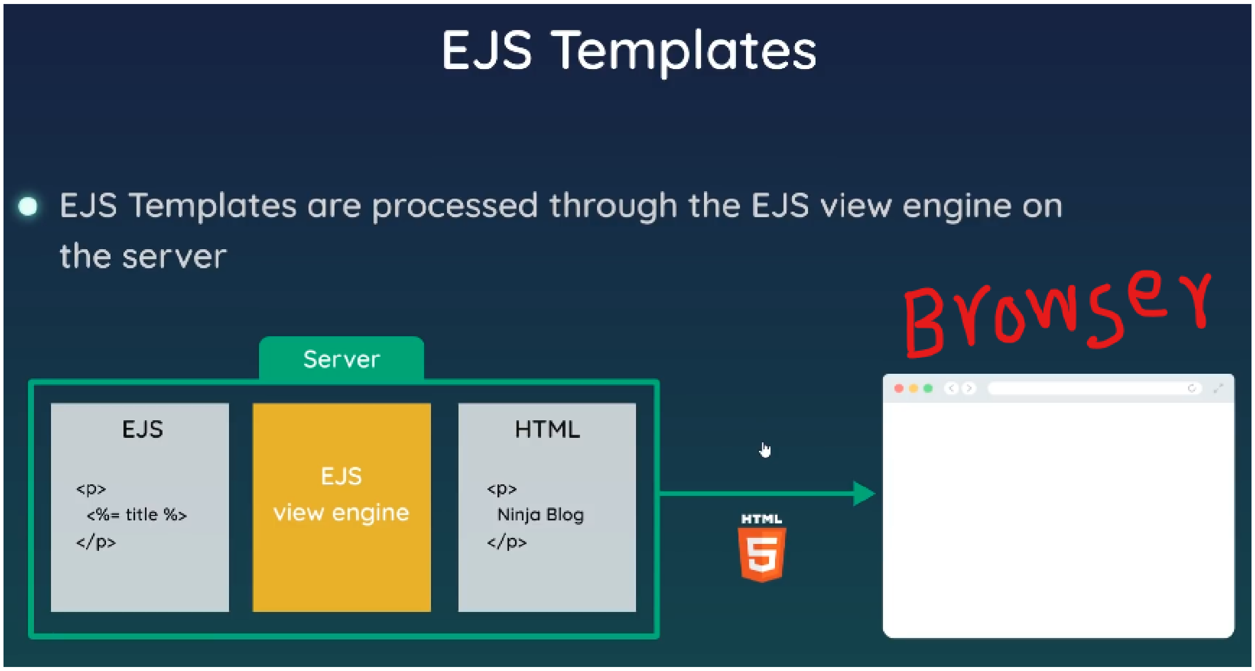
A Net Ninja Site

- Blogs
- About
- New Blog

All Blogs

There is no blogs to display...

EJS Template



EJS Templates

- EJS Templates are processed through the EJS view engine on the server

This process is called “Server side rendering”

Partials

view→partials →footer.ejs

→head.ejs

→nav.ejs

add.js

```
const express = require('express');

// express app
const app = express();

//register view engine
app.set('view engine', 'ejs');

// listen for requests
app.listen(3000);
app.get('/', (req, res)=>{
    const blogs = [
        {title: 'Sandy finds eggs', snippet: 'Lorem ipsum dolor sit amet consectetur'},
        {title: 'Raghava finds stars', snippet: 'Lorem ipsum dolor sit amet consectetur'},
        {title: 'How to defeat bowser', snippet: 'Lorem ipsum dolor sit amet consectetur'},
    ];
    res.render('index', {title: 'Home', blogs});
});

app.get('/about', (req, res)=>{
    res.render('about', {title: 'About'});
});

app.get('/blogs/create', (req, res)=>{
    res.render('create', {title: 'Create a new Blog'});
});

// 404 page

app.use((req, res) =>{
    res.status(404).render('404', {title: '404'});
});
```

index.html

```
<html lang="en">
    <%- include('./partials/head.ejs') %>
<body>

   <%- include('./partials/nav.ejs') %>

    <div class="blogs content">
        <h2>All Blogs</h2>

        <% if(blogs.length > 0) { %>
            <% blogs.forEach(blog => { %>
                <h3 class="title"><%= blog.title %></h3>
                <p class="snippet"><%= blog.snippet %></p>

            <% }) %>
        <% } else { %>
            <p>There is no blogs to display...</p>
        <% } %>

    </div>

    <%- include('./partials/footer.ejs') %>

</body>
</html>
```

about.html

```
<html lang="en">

<%- include('./partials/head.ejs') %>
<body>

    <%- include('./partials/nav.ejs') %>

    <div class="about content">
        <h2>About Us</h2>
        <p>Lorem ipsum dolor, sit amet consectetur adipisicing elit. Quia quibusdam quaerat illo a modi tenetur ut blanditiis, illum quo consectetur recusandae excepturi natus impedit rem, tempora cum fuga quae in.</p>
        <p>Lorem ipsum dolor, sit amet consectetur adipisicing elit. Quia quibusdam quaerat illo a modi tenetur ut blanditiis, illum quo consectetur recusandae excepturi natus impedit rem, tempora cum fuga quae in.</p>
        <p>Lorem ipsum dolor, sit amet consectetur adipisicing elit. Quia quibusdam quaerat illo a modi tenetur ut blanditiis, illum quo consectetur recusandae excepturi natus impedit rem, tempora cum fuga quae in.</p>
    </div>
    <%- include('./partials/footer.ejs') %>

</body>
</html>
```

404.html

```
<html lang="en">

    <%- include('./partials/head.ejs') %>
<body>

    <%- include('./partials/nav.ejs') %>

    <div class="not-found content">
        OOPS, page not found :)
    </div>

    <%- include('./partials/footer.ejs') %>

</body>
</html>
```

create.html

```
<html lang="en">

    <%- include('./partials/head.ejs') %>
<body>

    <%- include('./partials/nav.ejs') %>

    <div class="create-blogs content">
        <form>
            <label for="title">Blog title:</label>
            <input type="text" id="title" name="title" required>
            <label for="snippet">Blog snippet:</label>
            <input type="text" id="snippet" name="snippet" required>
            <label for="body">Blog body:</label>
            <textarea id="body" name="body" required></textarea>
        <button>Submit</button>
        </form>

    </div>

    <%- include('./partials/footer.ejs') %>

</body>
</html>
```

head.ejs

```
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Blog Ninja | <%= title %> </title>
  </head>
```

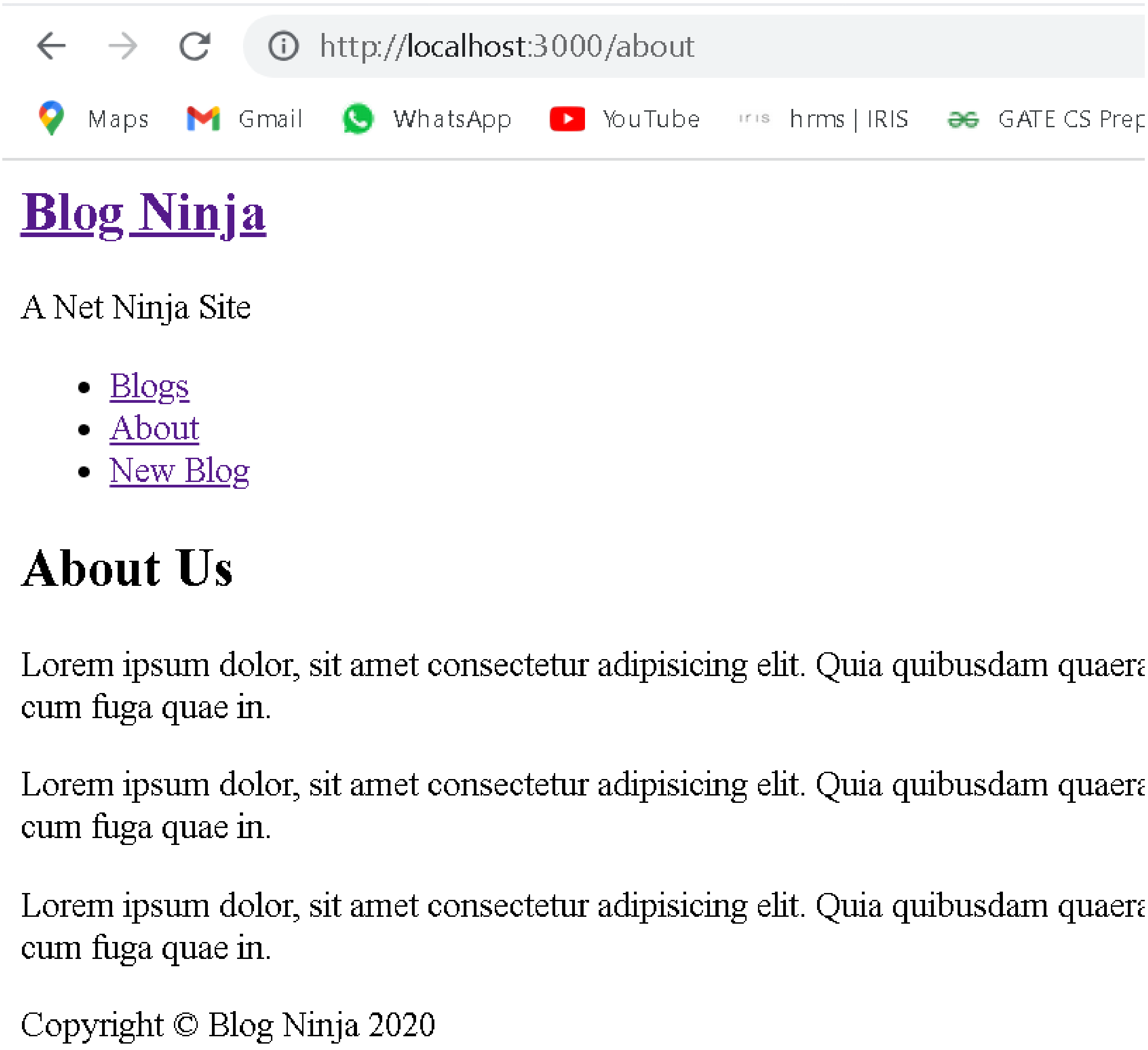
nav.ejs

```
<nav>
    <div class = "site-title">
        <a href="/"><h1>Blog Ninja</h1></a>
        <p>A Net Ninja Site</p>
    </div>
    <ul>
        <li><a href="/">Blogs</a></li>
        <li><a href="/about">About</a></li>
        <li><a href="/blogs/create">New Blog</a></li>
    </ul>
</nav>
```

footer.ejs

```
<footer>
    Copyright &copy; Blog Ninja 2020
</footer>
```

O/P:



Blog Ninja

A Net Ninja Site

- Blogs
- About
- New Blog

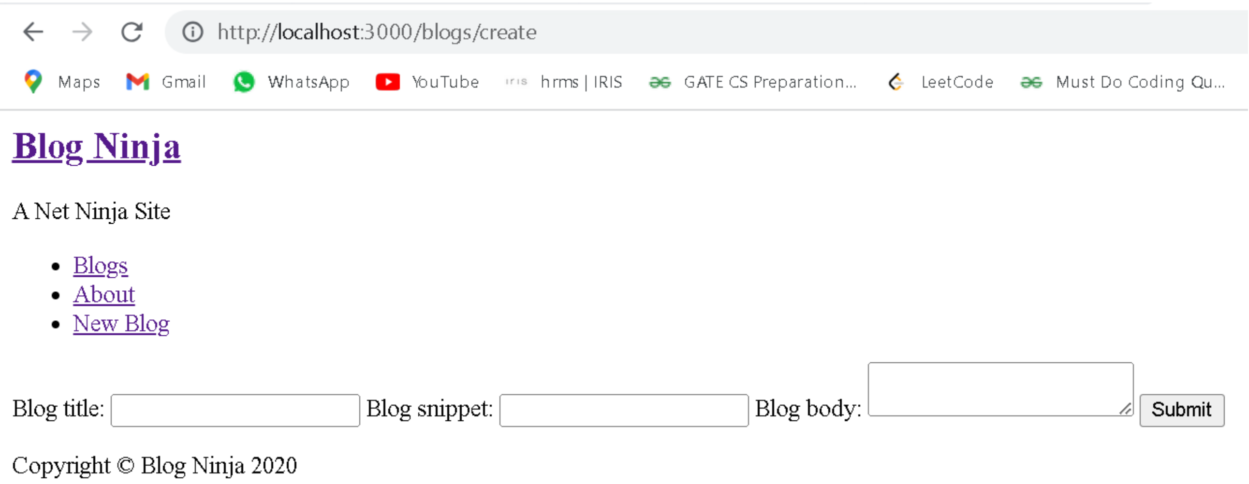
About Us

Lorem ipsum dolor, sit amet consectetur adipisicing elit. Quia quibusdam quaera
cum fuga quae in.

Lorem ipsum dolor, sit amet consectetur adipisicing elit. Quia quibusdam quaera
cum fuga quae in.

Lorem ipsum dolor, sit amet consectetur adipisicing elit. Quia quibusdam quaera
cum fuga quae in.

Copyright © Blog Ninja 2020



http://localhost:3000/blogs/create

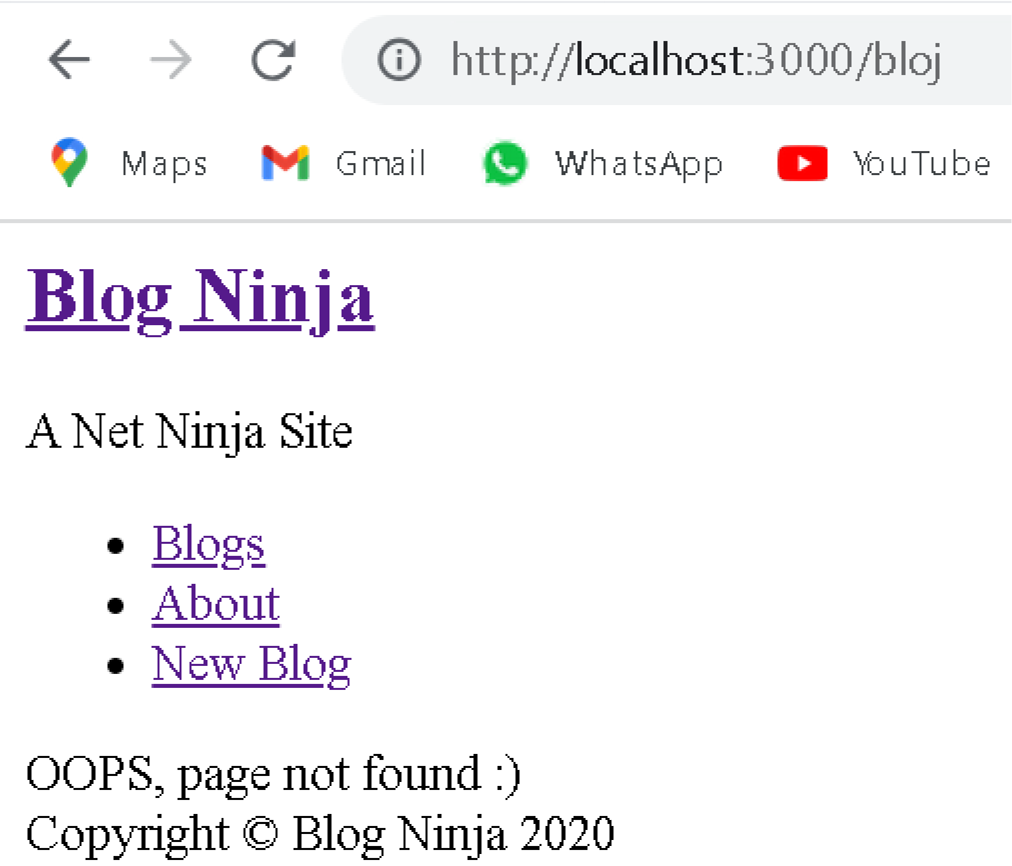
Blog Ninja

A Net Ninja Site

- Blogs
- About
- New Blog

Blog title: Blog snippet: Blog body:

Copyright © Blog Ninja 2020



http://localhost:3000/bloj

Blog Ninja

A Net Ninja Site

- Blogs
- About
- New Blog

OOPS, page not found :)
Copyright © Blog Ninja 2020

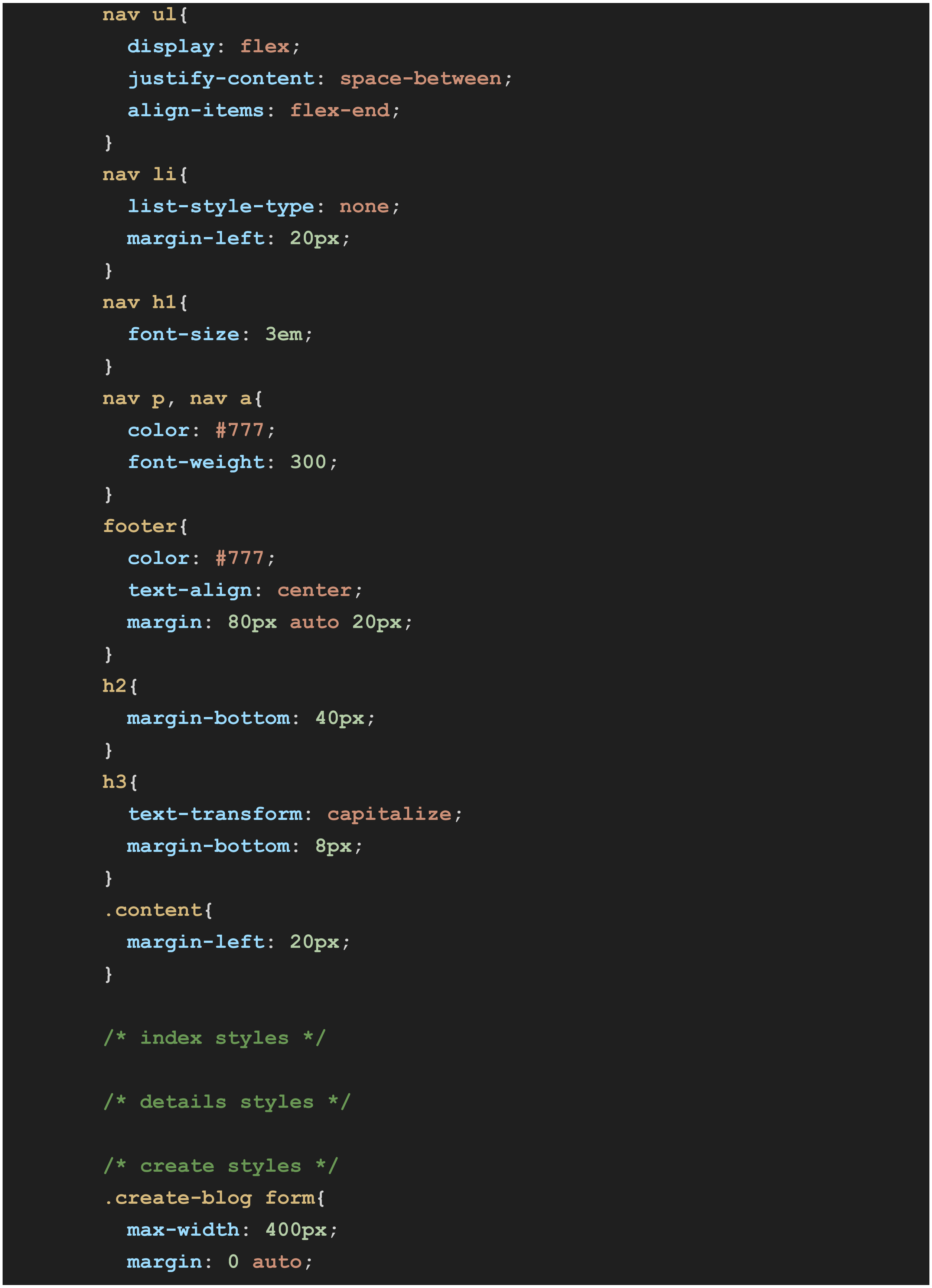
Adding CSS

→https://fonts.google.com/

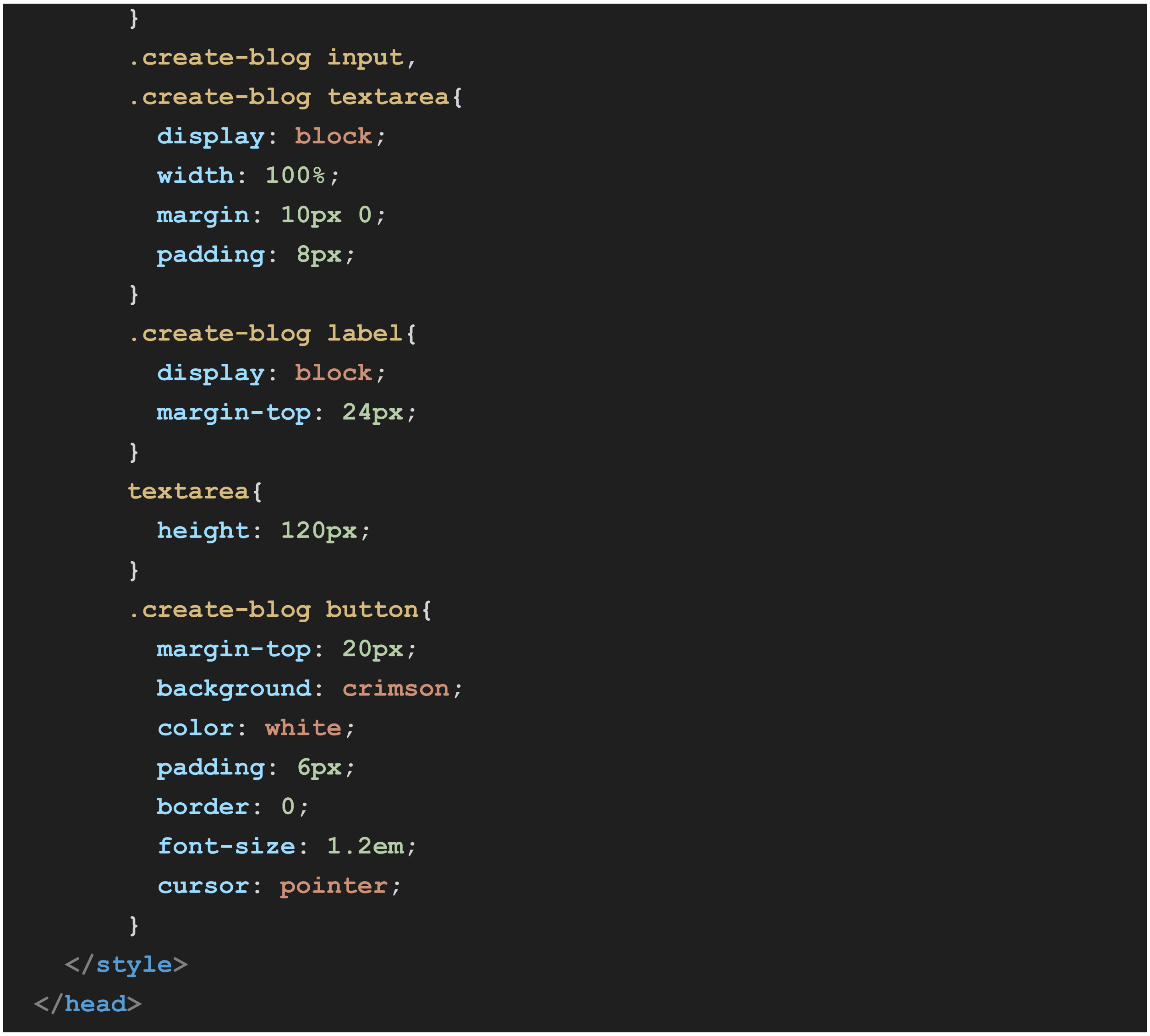
The above link selecting the style

head.ejs

```
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Blog Ninja | <%= title %> </title>
    <style>
        @import url('https://fonts.googleapis.com/css2?family=Noto+Serif:wght@400;700&display=swap');
        body{
          max-width: 1200px;
          margin: 20px auto;
          padding: 0 20px;
          font-family: 'Noto Serif', serif;
        }
        p, h1, h2, h3, a, ul{
          margin: 0;
          padding: 0;
          text-decoration: none;
          color: #222;
        }

        /* nav & footer styles */
        nav{
          display: flex;
          justify-content: space-between;
          margin-bottom: 60px;
          padding-bottom: 10px;
          border-bottom: 1px solid #ddd;
          text-transform: uppercase;
        }
```

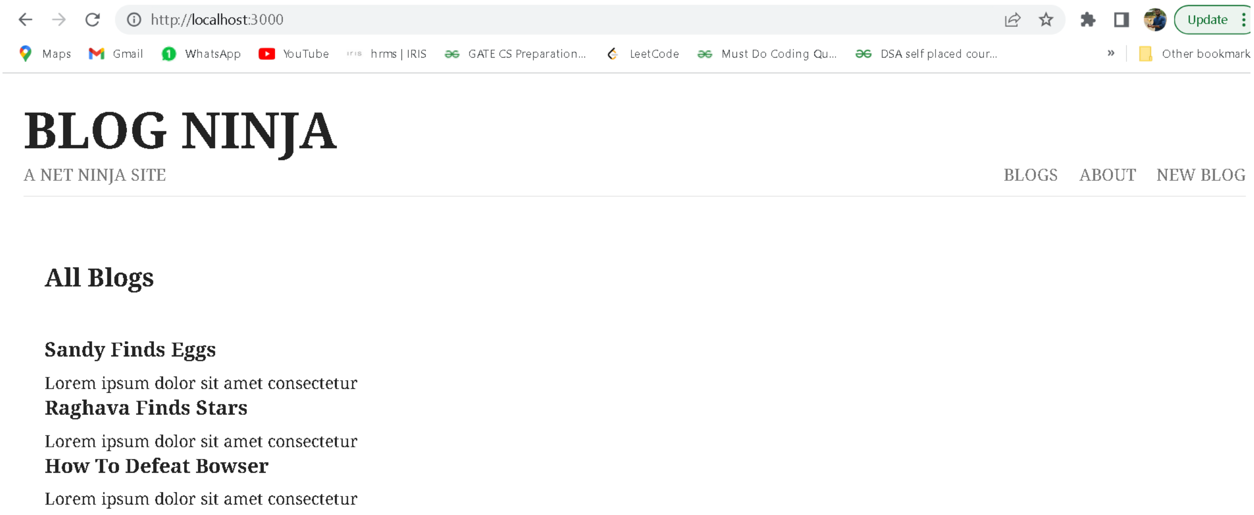


```
nav ul{
  display: flex;
  justify-content: space-between;
  align-items: flex-end;
}
nav li{
  list-style-type: none;
  margin-left: 20px;
}
nav h1{
  font-size: 3em;
}
nav p, nav a{
  color: #777;
  font-weight: 300;
}
footer{
  color: #777;
  text-align: center;
  margin: 80px auto 20px;
}
h2{
  margin-bottom: 40px;
}
h3{
  text-transform: capitalize;
  margin-bottom: 8px;
}
.content{
  margin-left: 20px;
}

/* index styles */

/* details styles */

/* create styles */
.create-blog form{
  max-width: 400px;
  margin: 0 auto;
```



```
        }
        .create-blog input,
        .create-blog textarea{
          display: block;
          width: 100%;
          margin: 10px 0;
          padding: 8px;
        }
        .create-blog label{
          display: block;
          margin-top: 24px;
        }
        textarea{
          height: 120px;
        }
        .create-blog button{
          margin-top: 20px;
          background: crimson;
          color: white;
          padding: 6px;
          border: 0;
          font-size: 1.2em;
          cursor: pointer;
        }
    </style>
  </head>
```

O/P:



BLOG NINJA

A NET NINJA SITE

BLOGS ABOUT NEW BLOG

All Blogs

Sandy Finds Eggs

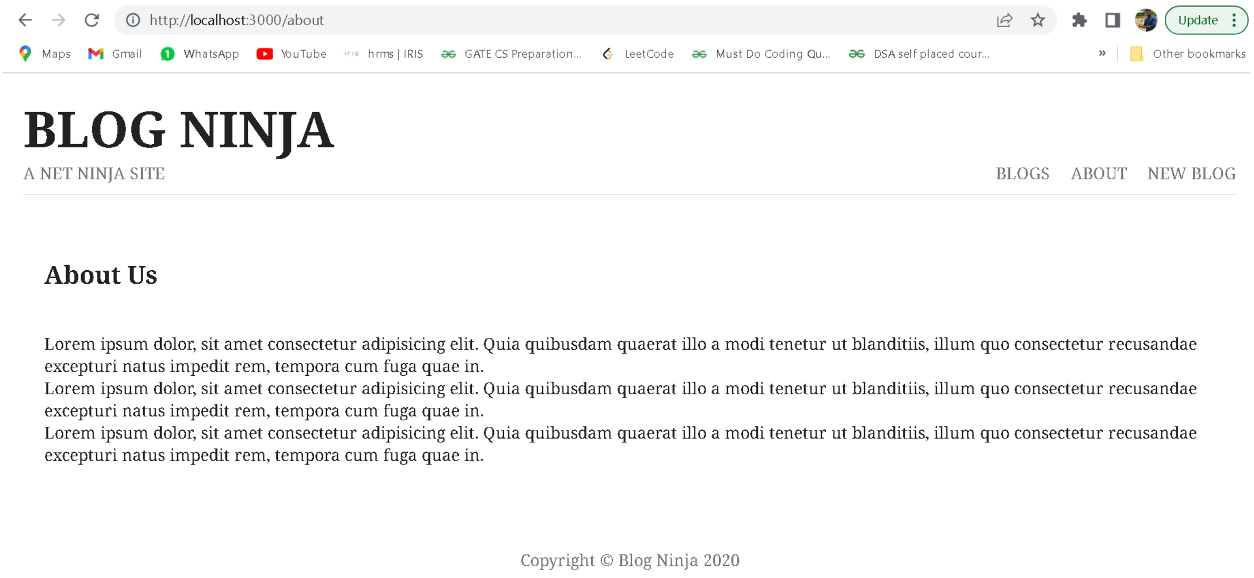
Lorem ipsum dolor sit amet consectetur

Raghava Finds Stars

Lorem ipsum dolor sit amet consectetur

How To Defeat Bowser

Lorem ipsum dolor sit amet consectetur



BLOG NINJA

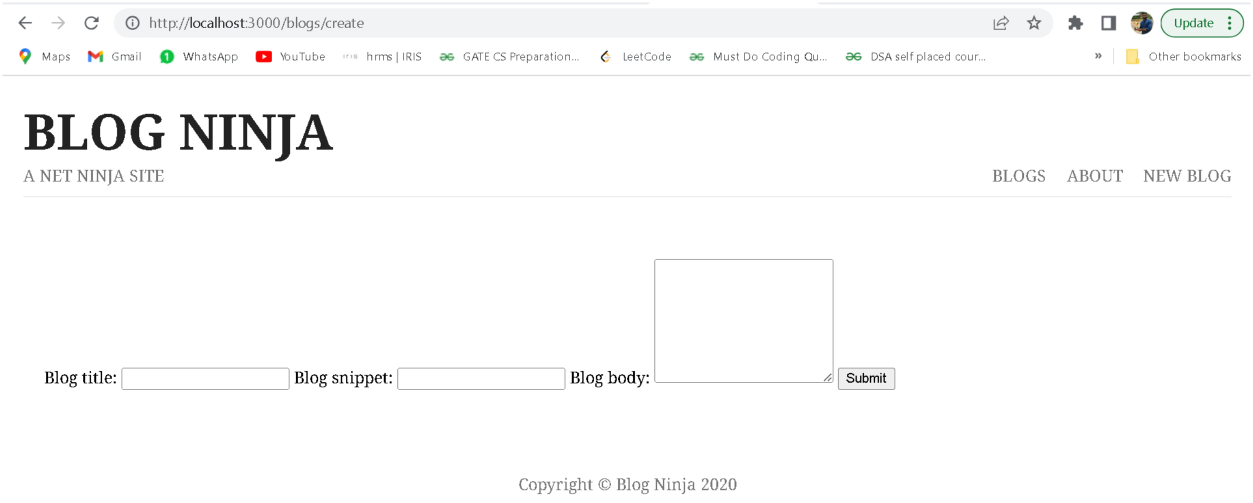
A NET NINJA SITE

BLOGS ABOUT NEW BLOG

About Us

Lorem ipsum dolor, sit amet consectetur adipisicing elit. Quia quibusdam quaerat illo a modi tenetur ut blanditiis, illum quo consectetur recusandae excepturi natus impedit rem, tempora cum fuga quae in.
Lorem ipsum dolor, sit amet consectetur adipisicing elit. Quia quibusdam quaerat illo a modi tenetur ut blanditiis, illum quo consectetur recusandae excepturi natus impedit rem, tempora cum fuga quae in.
Lorem ipsum dolor, sit amet consectetur adipisicing elit. Quia quibusdam quaerat illo a modi tenetur ut blanditiis, illum quo consectetur recusandae excepturi natus impedit rem, tempora cum fuga quae in.

Copyright © Blog Ninja 2020



BLOG NINJA

A NET NINJA SITE

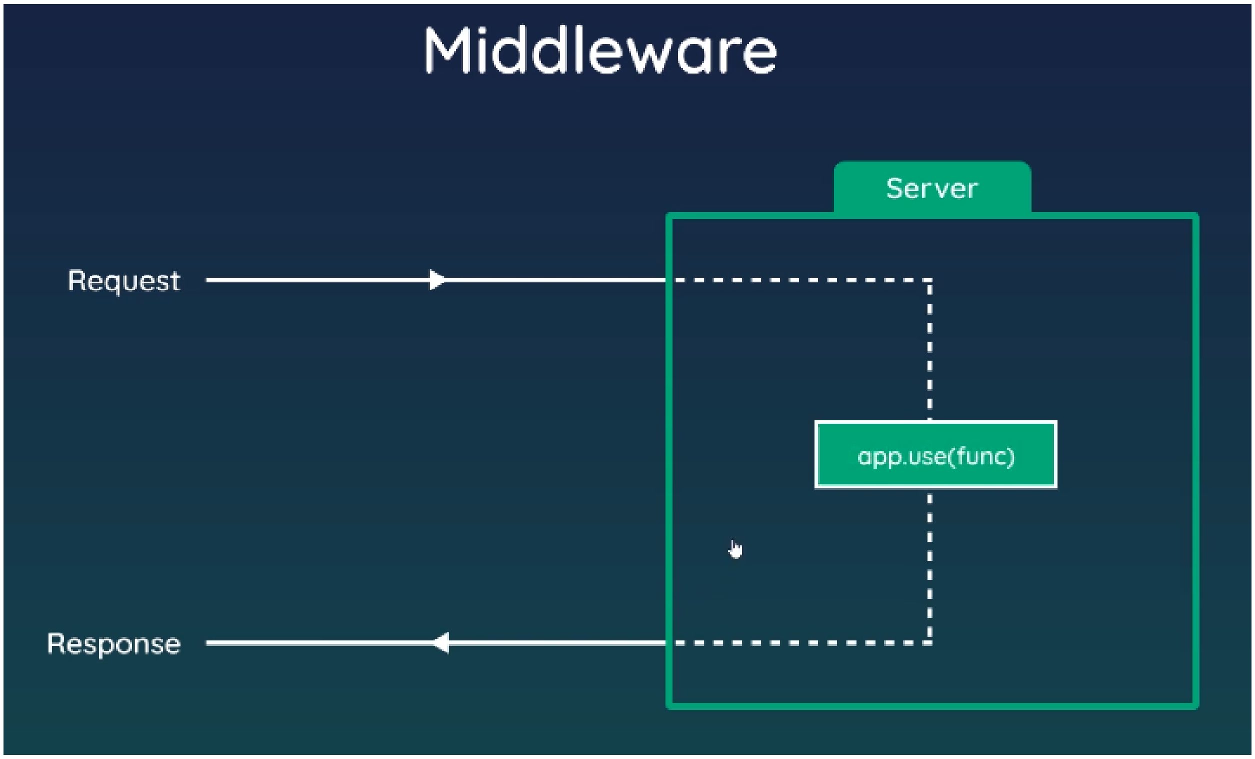
BLOGS ABOUT NEW BLOG

Blog title: Blog snippet: Blog body: Submit

Copyright © Blog Ninja 2020

Lesson 8

Middleware



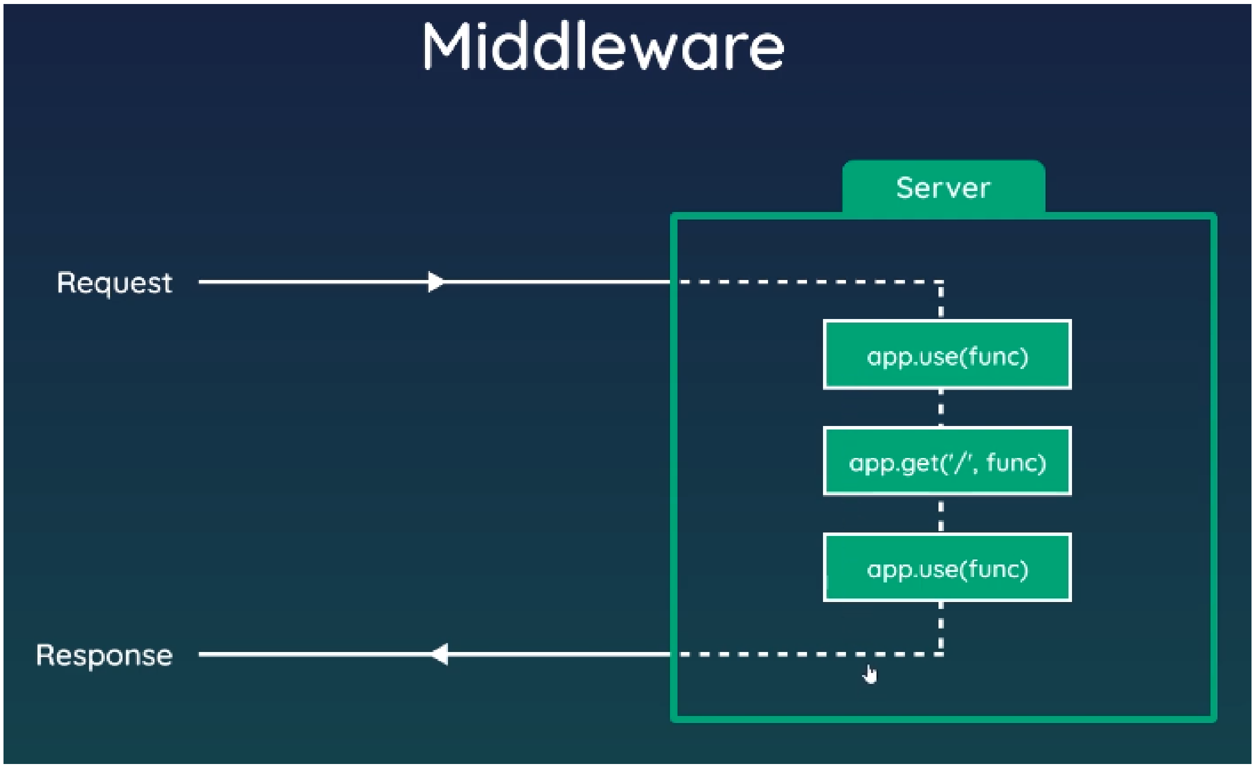
Middleware

Server

Request

app.use(func)

Response



Middleware

Server

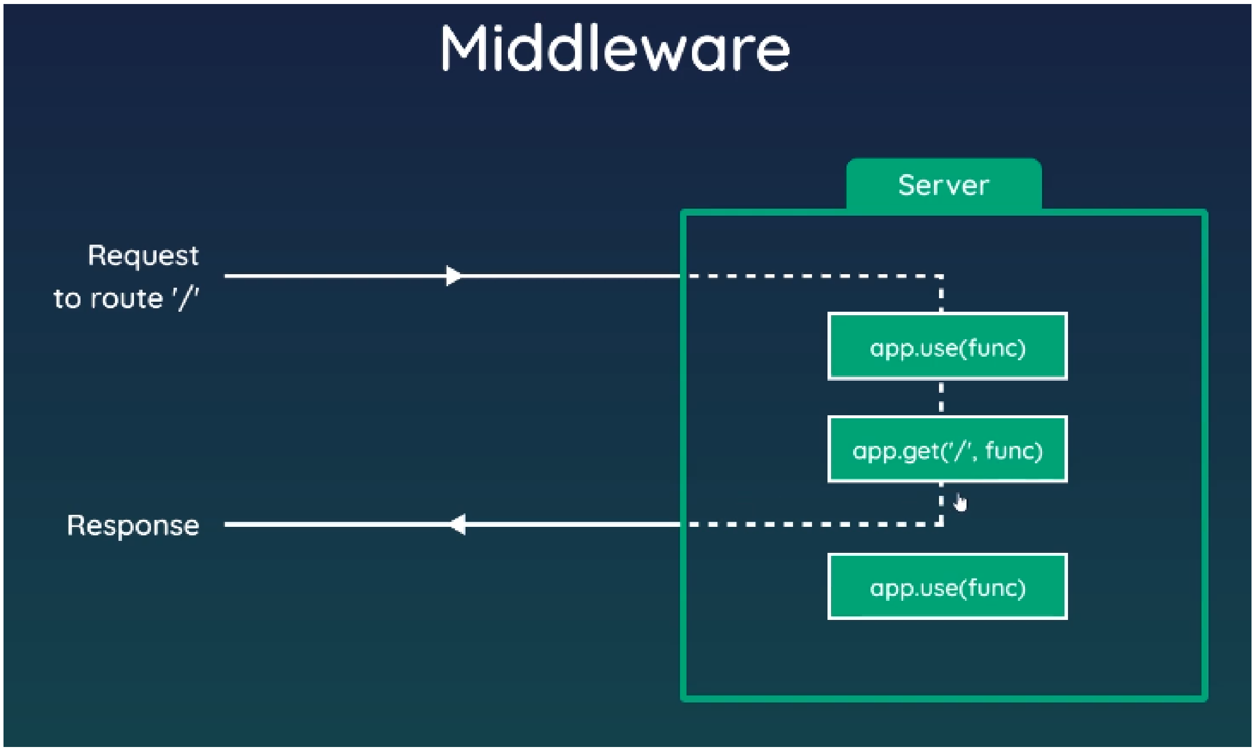
Request

app.use(func)

app.get('/', func)

app.use(func)

Response



Middleware

Server

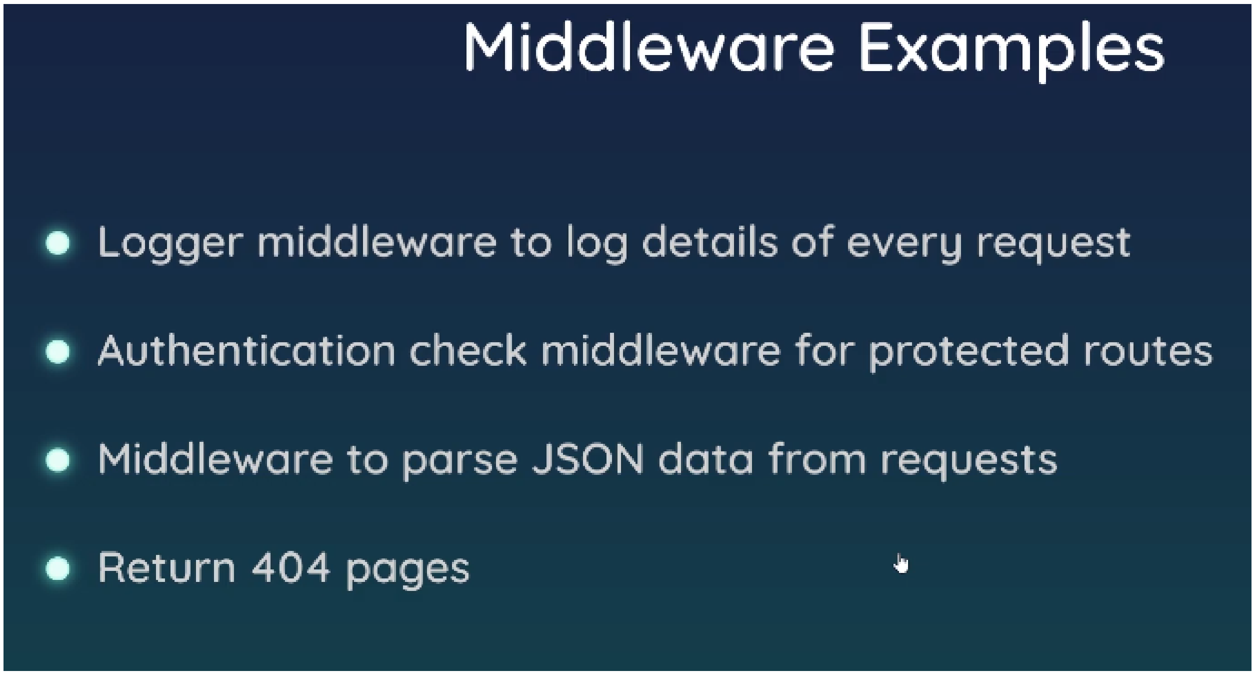
Request
to route '/'

app.use(func)

app.get('/', func)

Response

app.use(func)



Middleware Examples

- Logger middleware to log details of every request
- Authentication check middleware for protected routes
- Middleware to parse JSON data from requests
- Return 404 pages

If i use middleware then it can’t be stopped

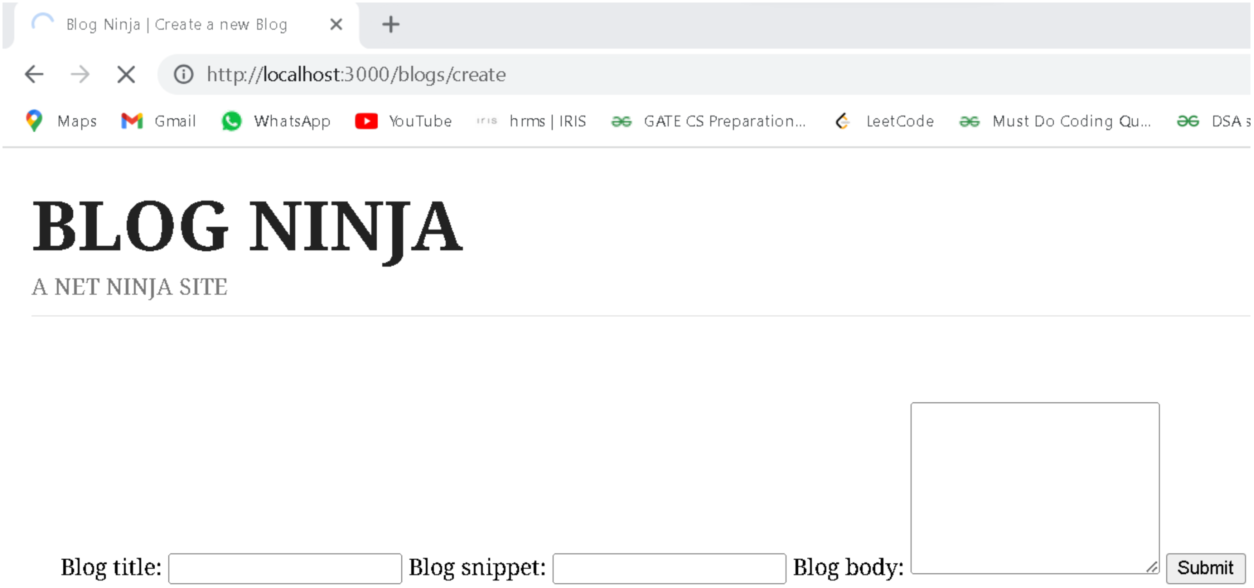
app.js

```
const express = require('express');

// express app
const app = express();

//register view engine
app.set('view engine', 'ejs');

// listen for requests
app.listen(3000);

app.use((req, res) => {
    console.log('new request made:');
    console.log('host: ', req.hostname);
    console.log('path: ', req.path);
    console.log('method: ', req.method);
    //next();
  });
```

```
app.get('/', (req, res)=>{
    const blogs = [
        {title: 'Sandy finds eggs', snippet: 'Lorem ipsum dolor sit amet
consectetur'},
        {title: 'Raghava finds stars', snippet: 'Lorem ipsum dolor sit
amet consectetur'},
        {title: 'How to defeat bowser', snippet: 'Lorem ipsum dolor sit
amet consectetur'},
    ];
    res.render('index', {title: 'Home', blogs});
});

app.get('/about', (req, res)=>{
    res.render('about', {title: 'About'});
});

app.get('/blogs/create', (req, res)=>{
    res.render('create', {title: 'Create a new Blog'});
});

// 404 page

app.use((req, res) =>{
    res.status(404).render('404', {title: '404'});
});
```

O/P:

```
PS C:\Users\gudiw\Documents\tuts\node-crash-course>  powershell
-ExecutionPolicy Bypass -Command "nodemon app"
[nodemon] restarting due to changes...
[nodemon] starting `node app.js`
[nodemon] restarting due to changes...
[nodemon] starting `node app.js`
[nodemon] restarting due to changes...
new request made:
host:  localhost
path:  /blogs/create
method:  GET
```



BLOG NINJA

A NET NINJA SITE

Blog title: Blog snippet: Blog body:

→web page is still hanging

Using next()

```
const express = require('express');

// express app
const app = express();

//register view engine
app.set('view engine', 'ejs');

// listen for requests
app.listen(3000);

app.use((req, res, next) => {
    console.log('new request made:');
    console.log('host: ', req.hostname);
    console.log('path: ', req.path);
    console.log('method: ', req.method);
    next();
});
```

```
app.use((req, res, next) => {
    console.log('in the next middleware');
    next();
});

app.get('/', (req, res)=>{
    const blogs = [
        {title: 'Sandy finds eggs', snippet: 'Lorem ipsum dolor sit amet
consectetur'},
        {title: 'Raghava finds stars', snippet: 'Lorem ipsum dolor sit
amet consectetur'},
        {title: 'How to defeat bowser', snippet: 'Lorem ipsum dolor sit
amet consectetur'},
    ];
    res.render('index', {title: 'Home', blogs});
});

app.get('/about', (req, res)=>{
    res.render('about', {title: 'About'});
});

app.get('/blogs/create', (req, res)=>{
    res.render('create', {title: 'Create a new Blog'});
});

// 404 page

app.use((req, res) =>{
    res.status(404).render('404', {title: '404'});
});
```

O/P:

PS C:\Users\gudiw\Documents\tuts\node-crash-course> **powershell -ExecutionPolicy Bypass -Command "nodemon app"**

```
[nodemon] restarting due to changes...
[nodemon] starting `node app.js`
```

```
new request made:
host:  localhost
path:  /blogs/create
method:  GET
new request made:
host:  localhost
path:  /
method:  GET
[nodemon] restarting due to changes...
[nodemon] starting `node app.js`
new request made:
host:  localhost
path:  /
method:  GET
in the next middleware
new request made:
host:  localhost
path:  /about
method:  GET
in the next middleware
```

3rd-Party Middleware

PS C:\Users\gudiw\Documents\tuts\node-crash-course> **npm install morgan**

```
const express = require('express');
const morgan = require('morgan');

// express app
const app = express();

//register view engine
app.set('view engine', 'ejs');

// listen for requests
app.listen(3000);
app.use(morgan('dev'));
//app.use(morgan('tiny'));    //slight change

app.get('/', (req, res)=>{
    const blogs = [
        {title: 'Sandy finds eggs', snippet: 'Lorem ipsum dolor sit amet consectetur'},
        {title: 'Raghava finds stars', snippet: 'Lorem ipsum dolor sit amet consectetur'},
        {title: 'How to defeat bowser', snippet: 'Lorem ipsum dolor sit amet consectetur'},
    ];
    res.render('index', {title: 'Home', blogs});
});

app.get('/about', (req, res)=>{
    res.render('about', {title: 'About'});
});

app.get('/blogs/create', (req, res)=>{
    res.render('create', {title: 'Create a new Blog'});
});

// 404 page

app.use((req, res) =>{
    res.status(404).render('404', {title: '404'});
});
```

O/P:

```
PS C:\Users\gudiw\Documents\tuts\node-crash-course>powershell -ExecutionPolicy Bypass -Command "nodemon app"
[nodemon] restarting due to changes...
[nodemon] starting `node app.js`
GET / 304 - - 13.821 ms
```

Static Files

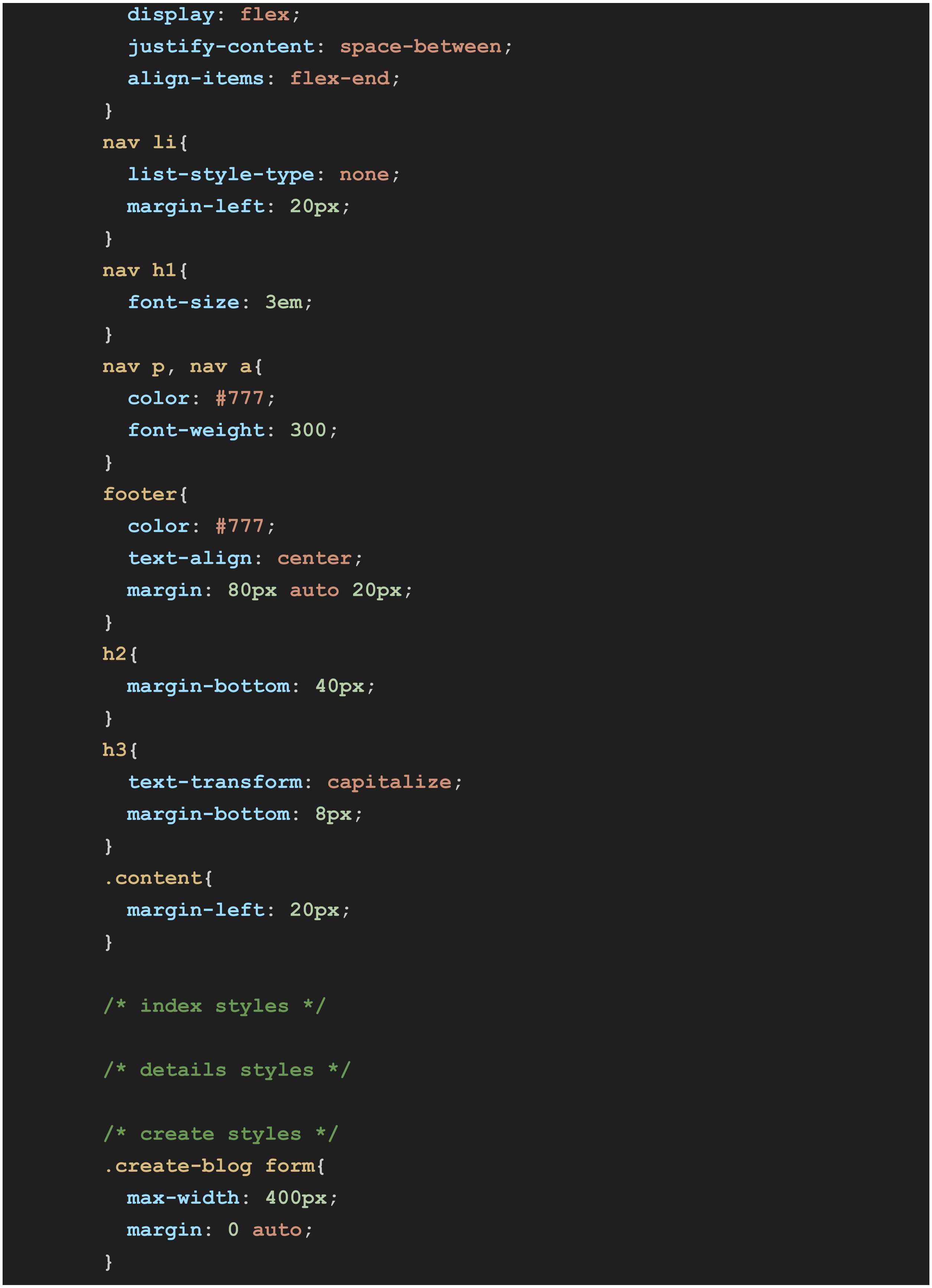
app.js

```
const express = require('express');
const morgan = require('morgan');

// express app
const app = express();

//register view engine
app.set('view engine', 'ejs');

// listen for requests
app.listen(3000);

// middleware & static files
app.use(express.static('public'));
app.use(morgan('dev'));
//app.use(morgan('tiny'));    //slight change

app.get('/', (req, res)=>{
    const blogs = [
        {title: 'Sandy finds eggs', snippet: 'Lorem ipsum dolor sit amet consectetur'},
        {title: 'Raghava finds stars', snippet: 'Lorem ipsum dolor sit amet consectetur'},
        {title: 'How to defeat bowser', snippet: 'Lorem ipsum dolor sit amet consectetur'},
    ];
    res.render('index', {title: 'Home', blogs});
});

app.get('/about', (req, res)=>{
    res.render('about', {title: 'About'});
});

app.get('/blogs/create', (req, res)=>{
    res.render('create', {title: 'Create a new Blog'});
});

// 404 page

app.use((req, res) =>{
    res.status(404).render('404', {title: '404'});
});
```

public →styles.css

style.css

```
@import
url('https://fonts.googleapis.com/css2?family=Noto+Serif:wght@400;700&display=swap');
        body{
          max-width: 1200px;
          margin: 20px auto;
          padding: 0 20px;
          font-family: 'Noto Serif', serif;
          max-width: 1200px;
        }
        p, h1, h2, h3, a, ul{
          margin: 0;
          padding: 0;
          text-decoration: none;
          color: #222;
        }

        /* nav & footer styles */
        nav{
          display: flex;
          justify-content: space-between;
          margin-bottom: 60px;
          padding-bottom: 10px;
          border-bottom: 1px solid #ddd;
          text-transform: uppercase;
        }
        nav ul{
```



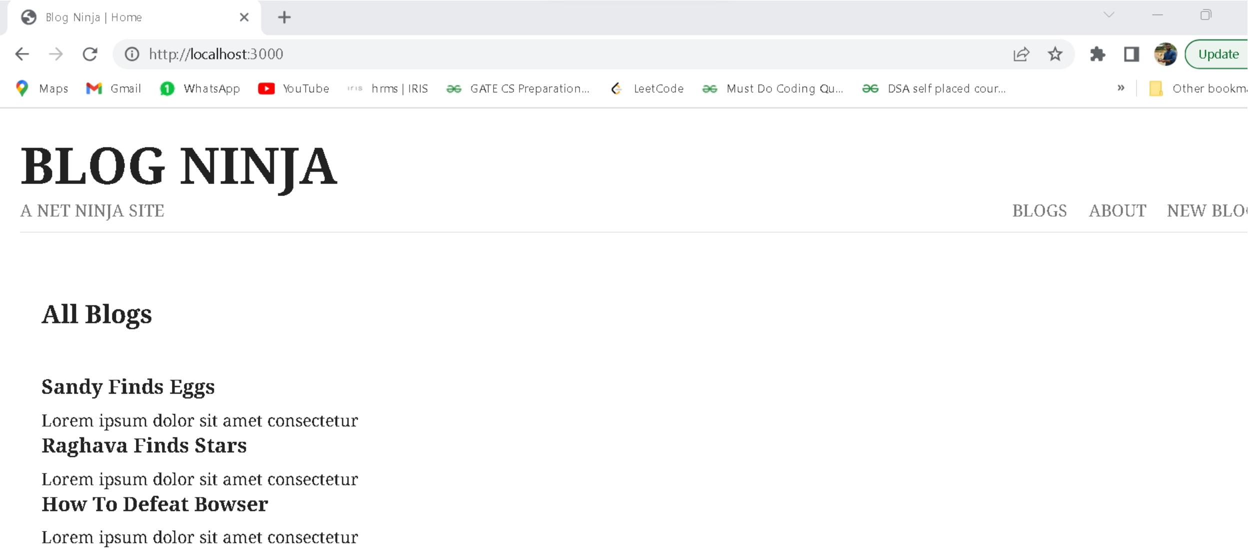
```
  display: flex;
  justify-content: space-between;
  align-items: flex-end;
}
nav li{
  list-style-type: none;
  margin-left: 20px;
}
nav h1{
  font-size: 3em;
}
nav p, nav a{
  color: #777;
  font-weight: 300;
}
footer{
  color: #777;
  text-align: center;
  margin: 80px auto 20px;
}
h2{
  margin-bottom: 40px;
}
h3{
  text-transform: capitalize;
  margin-bottom: 8px;
}
.content{
  margin-left: 20px;
}

/* index styles */

/* details styles */

/* create styles */
.create-blog form{
  max-width: 400px;
  margin: 0 auto;
}
```

```
        .create-blog input,
        .create-blog textarea{
          display: block;
          width: 100%;
          margin: 10px 0;
          padding: 8px;
        }
        .create-blog label{
          display: block;
          margin-top: 24px;
        }
        textarea{
          height: 120px;
        }
        .create-blog button{
          margin-top: 20px;
          background: crimson;
          color: white;
          padding: 6px;
          border: 0;
          font-size: 1.2em;
          cursor: pointer;
        }
```

head.ejs

```
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Blog Ninja | <%= title %> </title>

    <link rel="stylesheet" href="/styles.css">

</head>
```

O/P:



BLOG NINJA

A NET NINJA SITE

BLOGS ABOUT NEW BLO

All Blogs

Sandy Finds Eggs

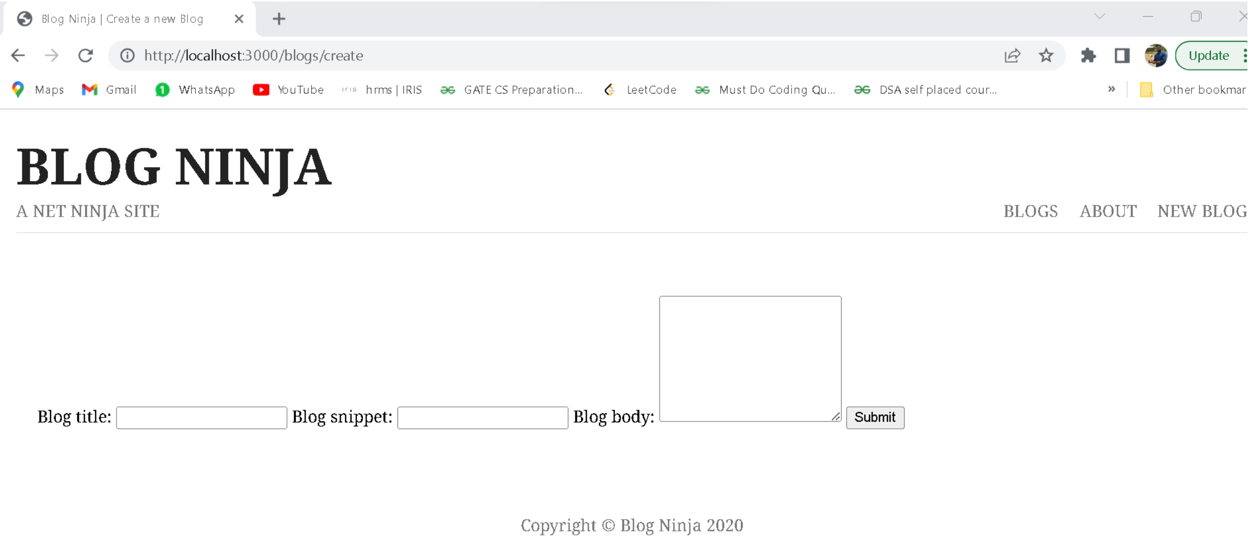
Lorem ipsum dolor sit amet consectetur

Raghava Finds Stars

Lorem ipsum dolor sit amet consectetur

How To Defeat Bowser

Lorem ipsum dolor sit amet consectetur



BLOG NINJA

A NET NINJA SITE

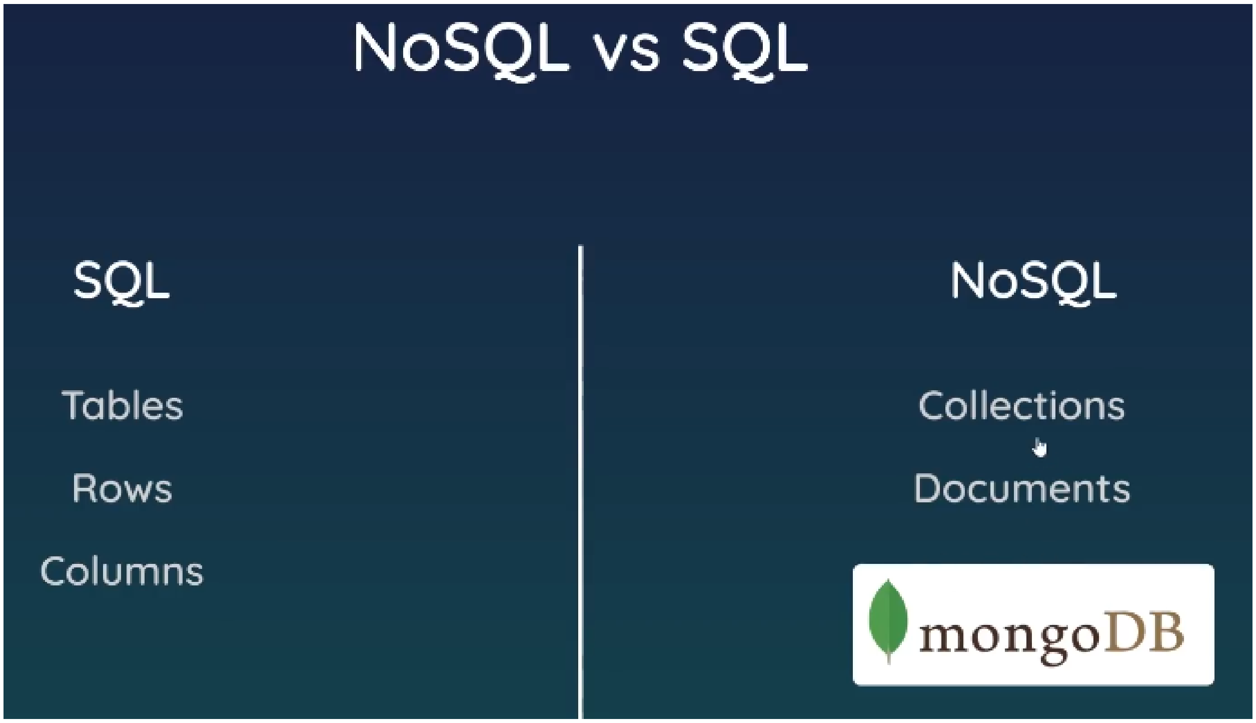
BLOGS ABOUT NEW BLOG

Blog title: ______ Blog snippet: ______ Blog body: ______ Submit

Copyright © Blog Ninja 2020

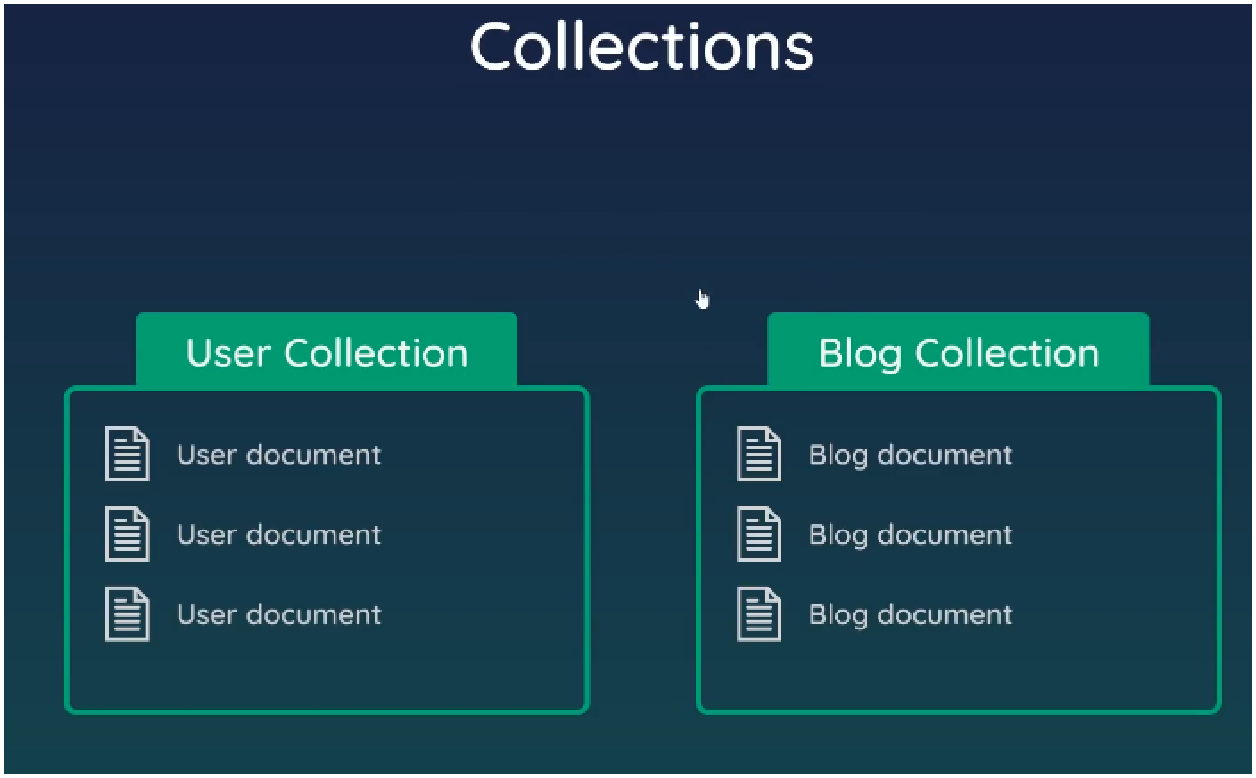
Lesson 9

MangoDB



NoSQL vs SQL

SQL	NoSQL
Tables	Collections
Rows	Documents
Columns	mongoDB



Collections

User Collection

User document

User document

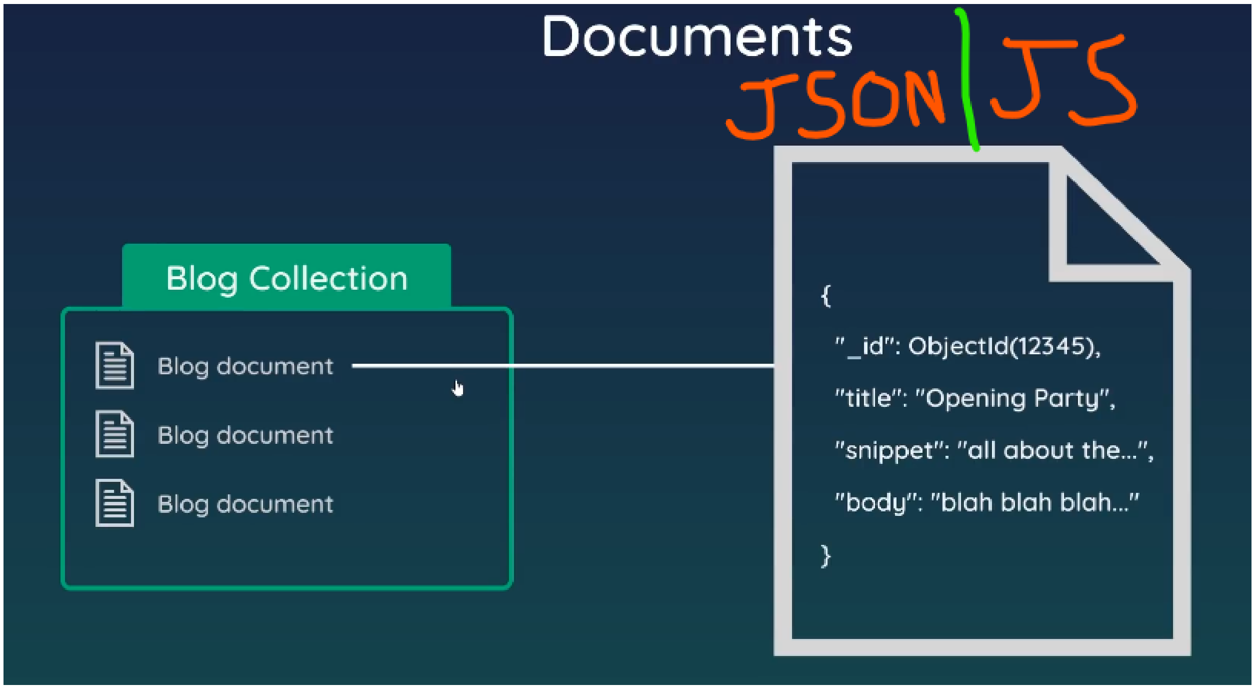
User document

Blog Collection

Blog document

Blog document

Blog document



Documents

JSON | JS

Blog Collection

Blog document

Blog document

Blog document

```
{
  "_id": ObjectId(12345),
  "title": "Opening Party",
  "snippet": "all about the...",
  "body": "blah blah blah..."
}
```

→Each document will store one block.

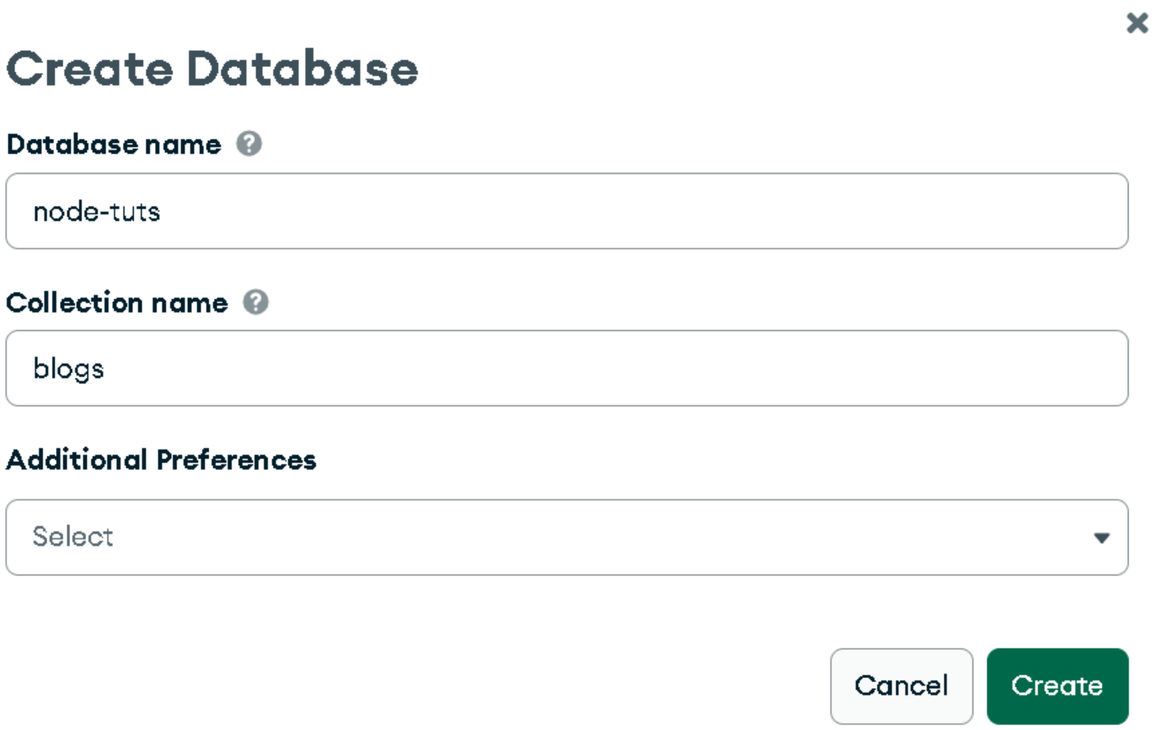
MangoDB Setup & Atlas

https://www.mongodb.com/atlas/database

Go to the Website signUP

1. Create cluster   (given cluster name → nodetuts)

2. Create database



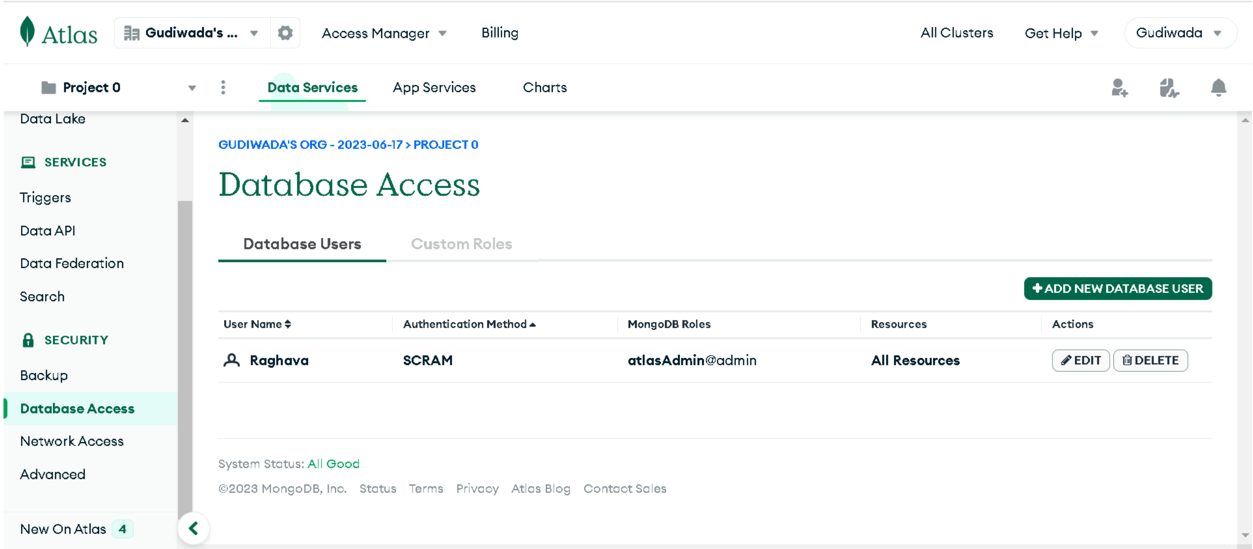
Create Database

Database name: node-tuts

Collection name: blogs

Additional Preferences: Select

Go to Database Access



GUDIWADA'S ORG - 2023-06-17 > PROJECT 0

Database Access

Database Users | Custom Roles

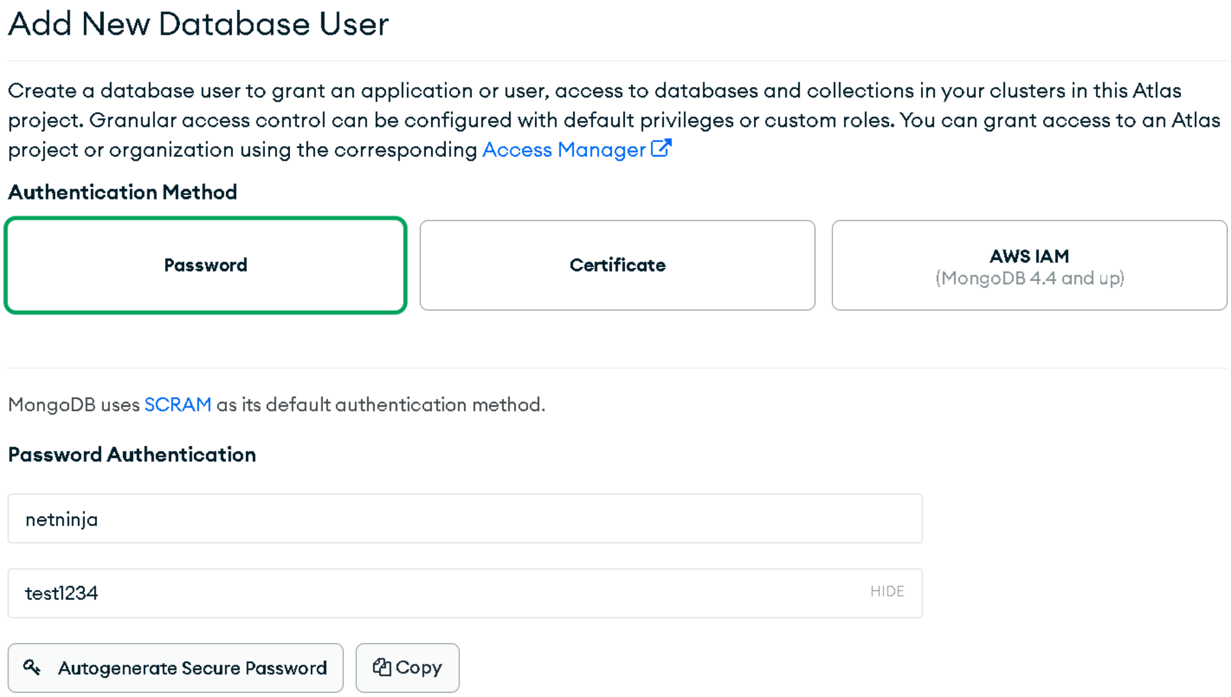
+ ADD NEW DATABASE USER

User Name	Authentication Method	MongoDB Roles	Resources	Actions
Raghava	SCRAM	atlasAdmin@admin	All Resources	EDIT DELETE

System Status: All Good

©2023 MongoDB, Inc. Status Terms Privacy Atlas Blog Contact Sales

Add new database user



Add New Database User

Create a database user to grant an application or user, access to databases and collections in your clusters in this Atlas project. Granular access control can be configured with default privileges or custom roles. You can grant access to an Atlas project or organization using the corresponding Access Manager

Authentication Method

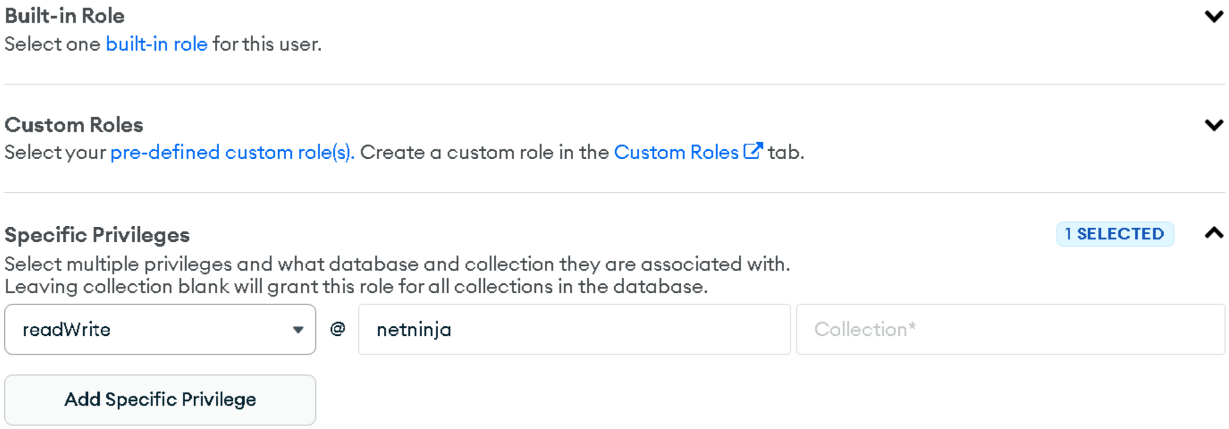
MongoDB uses SCRAM as its default authentication method.

Password Authentication

netninja

test1234

→while adding database user give access to the privileges



Built-in Role

Select one built-in role for this user.

Custom Roles

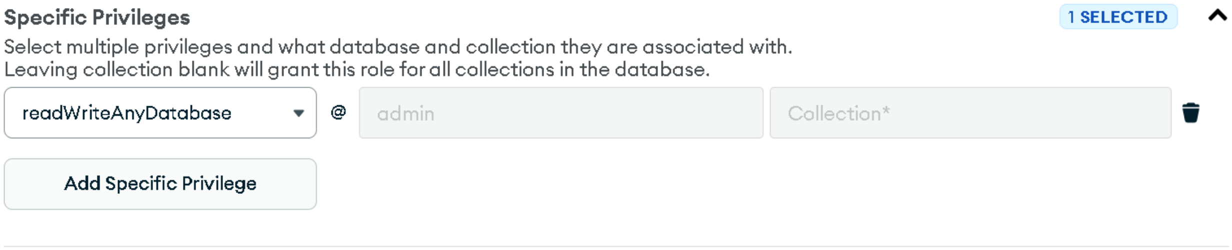
Select your pre-defined custom role(s). Create a custom role in the Custom Roles tab.

Specific Privileges

Select multiple privileges and what database and collection they are associated with.
Leaving collection blank will grant this role for all collections in the database.

readWrite @ netninja

We can also mention whatever the database we want

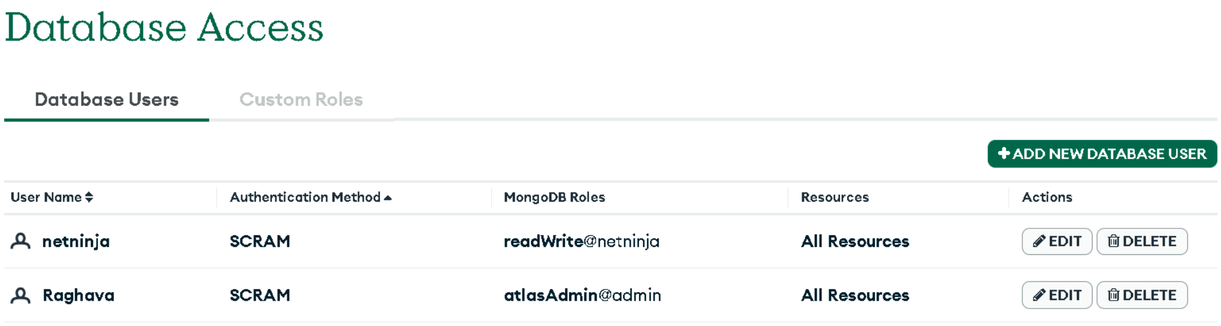


Specific Privileges

Select multiple privileges and what database and collection they are associated with.
Leaving collection blank will grant this role for all collections in the database.

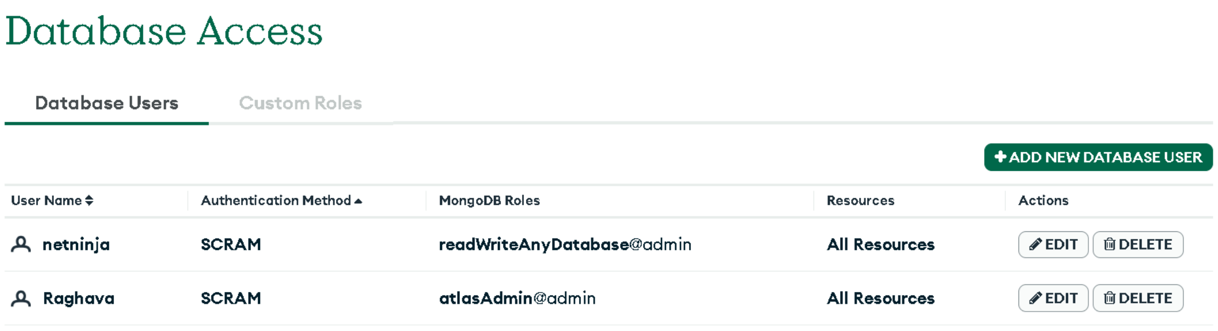
readWriteAnyDatabase @ admin

→After creation of Database user look like this



Database Access

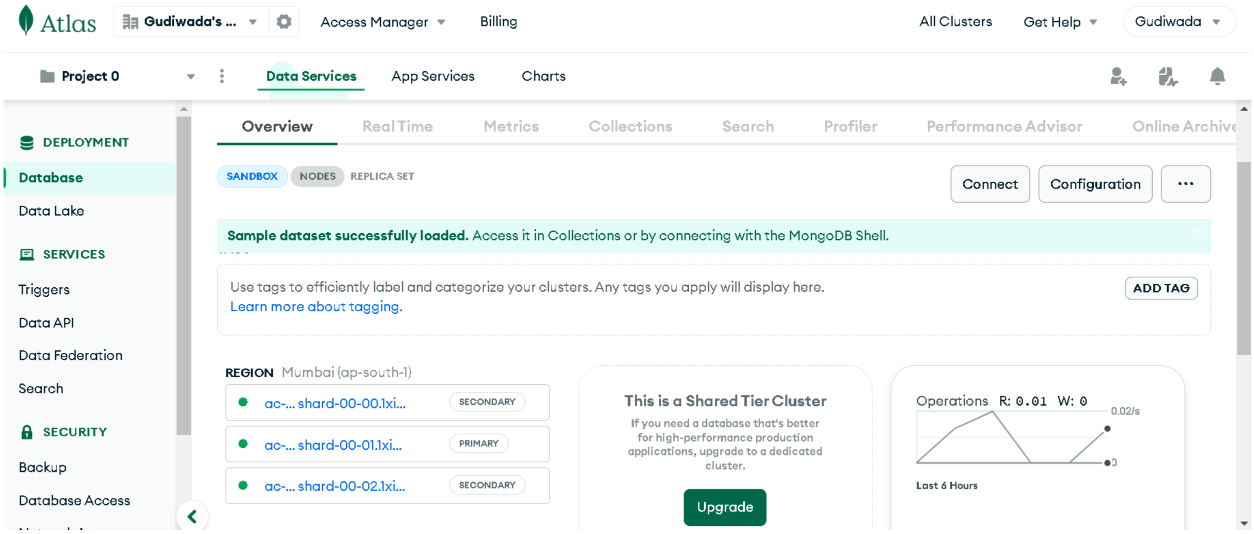
User Name	Authentication Method	MongoDB Roles	Resources	Actions
netninja	SCRAM	readWrite@netninja	All Resources	EDIT DELETE
Raghava	SCRAM	atlasAdmin@admin	All Resources	EDIT DELETE



Database Access

User Name	Authentication Method	MongoDB Roles	Resources	Actions
netninja	SCRAM	readWriteAnyDatabase@admin	All Resources	EDIT DELETE
Raghava	SCRAM	atlasAdmin@admin	All Resources	EDIT DELETE

—>Go To Connect



Sample dataset successfully loaded. Access it in Collections or by connecting with the MongoDB Shell.

Use tags to efficiently label and categorize your clusters. Any tags you apply will display here.
Learn more about tagging.

REGION Mumbai (ap-south-1)

● ac-... shard-00-00.1xi...	SECONDARY
● ac-... shard-00-01.1xi...	PRIMARY
● ac-... shard-00-02.1xi...	SECONDARY

This is a Shared Tier Cluster

If you need a database that's better for high-performance production applications, upgrade to a dedicated cluster.

→Drivers (select go to your Application)

mongodb+srv://<username>:<password>@nodetuts.1xigocw.mongodb.net/test?retryWrites=true&w=majority

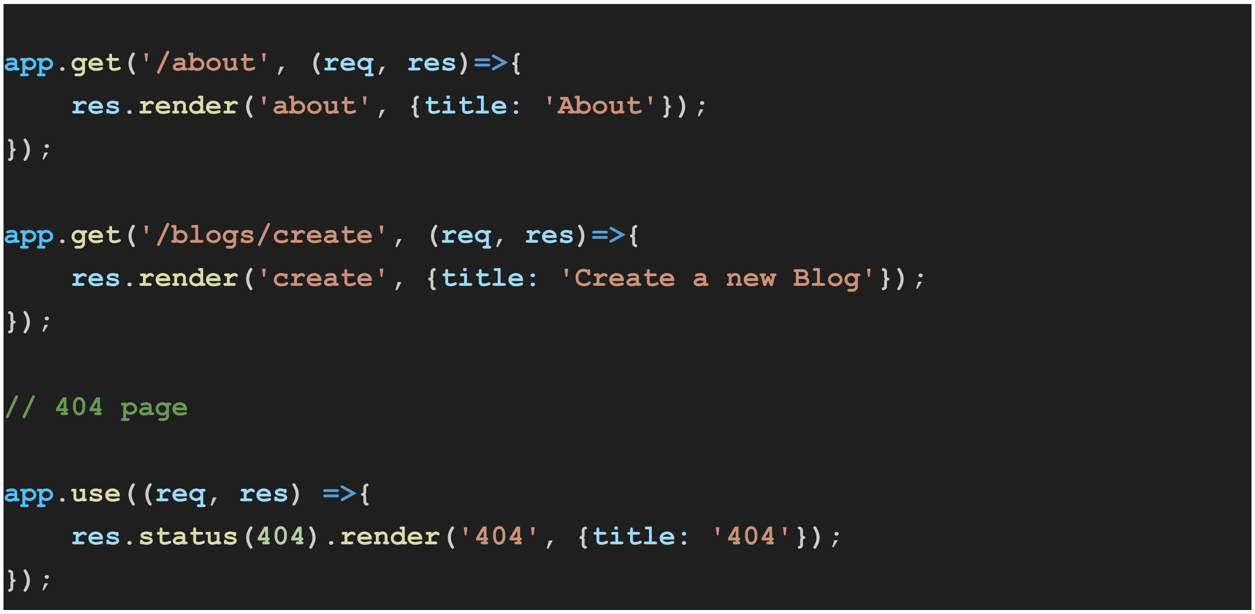
Example:

mongodb+srv://netninja:test1234@nodetuts.1xigocw.mongodb.net/node-tuts?retryWrites=true&w=majority

<username>   → netninja   (add new database user)
<password>   → test1234
test   →node-tuts   (database name)

app.js

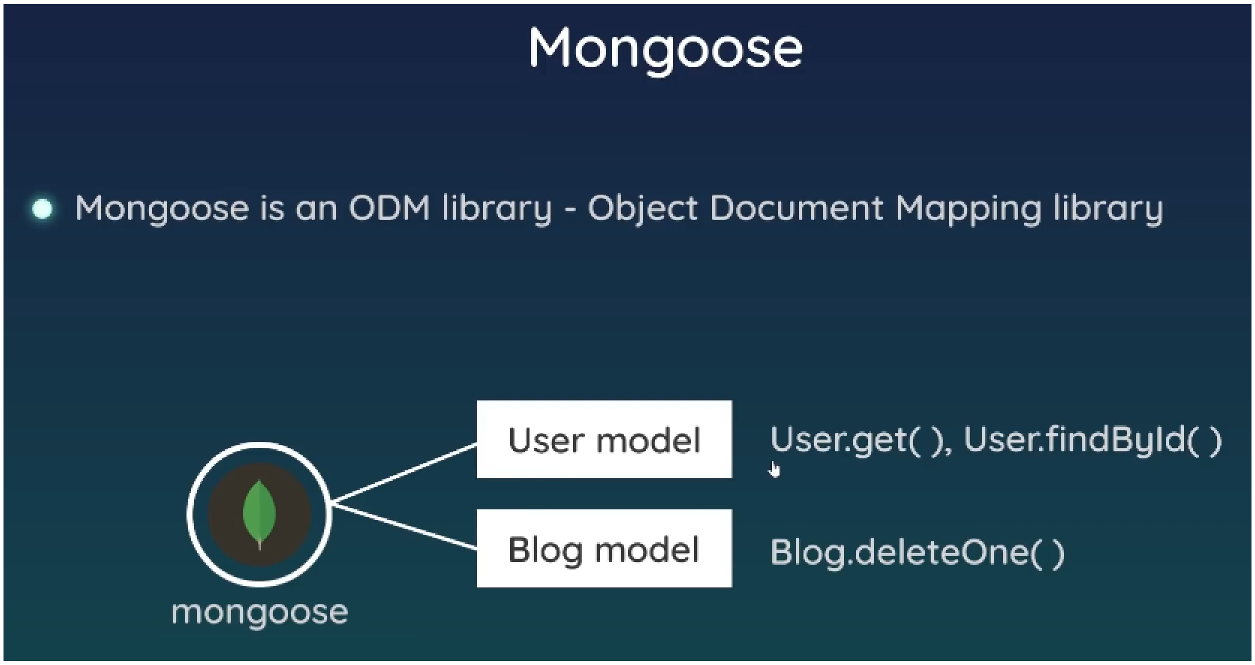
```
const express = require('express');
const morgan = require('morgan');


// express app
const app = express();

// connect to mongodb
const dbURI =
'mongodb+srv://netninja:test1234@nodetuts.1xigocw.mongodb.net/?retryWrites=true&w=majority';

//register view engine
app.set('view engine', 'ejs');

// listen for requests
app.listen(3000);

// middleware & static files
app.use(express.static('public'));
app.use(morgan('dev'));
//app.use(morgan('tiny'));    //slight change

app.get('/', (req, res)=>{
    const blogs = [
        {title: 'Sandy finds eggs', snippet: 'Lorem ipsum dolor sit amet consectetur'},
        {title: 'Raghava finds stars', snippet: 'Lorem ipsum dolor sit amet consectetur'},
        {title: 'How to defeat bowser', snippet: 'Lorem ipsum dolor sit amet consectetur'},
    ];
    res.render('index', {title: 'Home', blogs});
});
```



```
app.get('/about', (req, res)=>{
    res.render('about', {title: 'About'});
});

app.get('/blogs/create', (req, res)=>{
    res.render('create', {title: 'Create a new Blog'});
});

// 404 page

app.use((req, res) =>{
    res.status(404).render('404', {title: '404'});
});
```

Mongoose, Models & Schemas

PS C:\Users\gudiw\Documents\tuts\node-crash-course> **npm install mongoose**



Mongoose

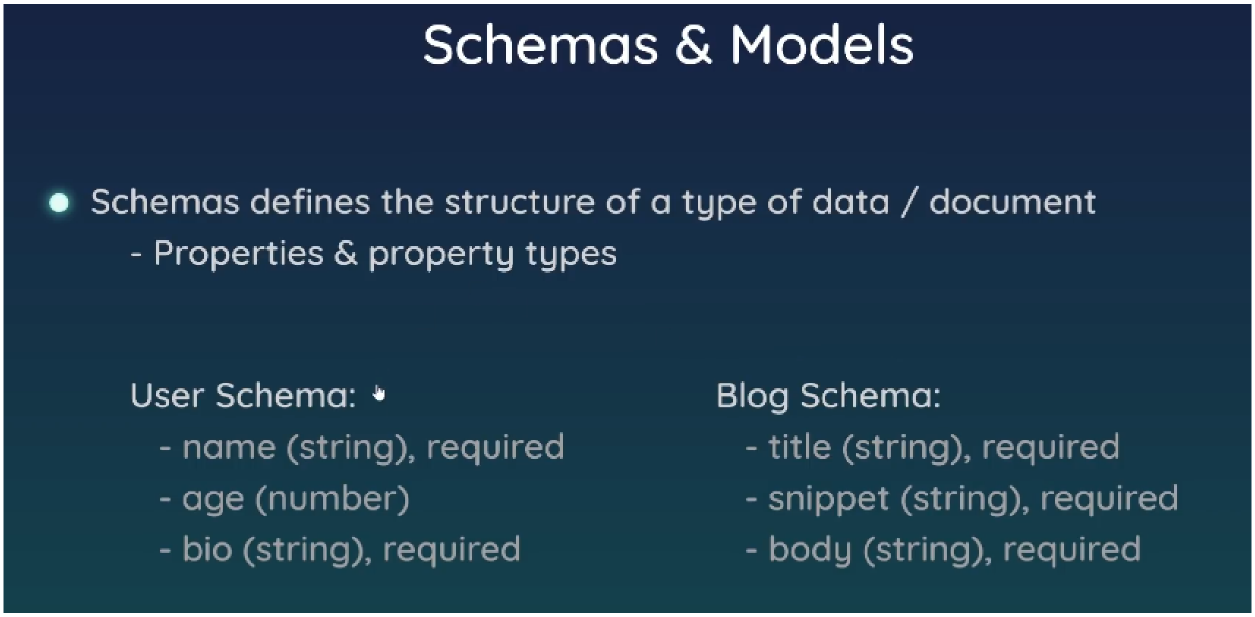
- Mongoose is an ODM library - Object Document Mapping library

app.js

```
const express = require('express');
const morgan = require('morgan');
const mongoose = require('mongoose');

// express app
const app = express();

// connect to mongodb
const dbURI =
'mongodb+srv://netninja:test1234@nodetuts.1xigocw.mongodb.net/node-tuts?re
tryWrites=true&w=majority';

//mongoose.connect(dbURI, { useNewUrlParser: true, useUnifiedTopology:
true })  //if u see deprecationwarning then use this line

mongoose.connect(dbURI)
  //.then(result => console.log('connected to db'))
  .then(result => app.listen(3000))
  .catch((err) => console.log(err));

//register view engine
app.set('view engine', 'ejs');


// middleware & static files
app.use(express.static('public'));
app.use(morgan('dev'));
//app.use(morgan('tiny'));    //slight change

app.get('/', (req, res)=>{
    const blogs = [
        {title: 'Sandy finds eggs', snippet: 'Lorem ipsum dolor sit amet
consectetur'},
        {title: 'Raghava finds stars', snippet: 'Lorem ipsum dolor sit
amet consectetur'},
        {title: 'How to defeat bowser', snippet: 'Lorem ipsum dolor sit
amet consectetur'},
    ];
    res.render('index', {title: 'Home', blogs});
});

app.get('/about', (req, res)=>{
    res.render('about', {title: 'About'});
});

app.get('/blogs/create', (req, res)=>{
    res.render('create', {title: 'Create a new Blog'});
});

// 404 page

app.use((req, res) =>{
    res.status(404).render('404', {title: '404'});
});
```

O/P:

```
PS C:\Users\gudiw\Documents\tuts\node-crash-course>  powershell -ExecutionPolicy Bypass -Command "nodemon app"
[nodemon] 2.0.22
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node app.js`
[nodemon] restarting due to changes...
[nodemon] starting `node app.js`
C:\Users\gudiw\Documents\tuts\node-crash-course\app.js:12
  .then(result => console.log('connected to db'))
connected to db
[nodemon] restarting due to changes...
[nodemon] starting `node app.js`
connected to db
[nodemon] restarting due to changes...
[nodemon] restarting due to changes...
[nodemon] starting `node app.js`
```

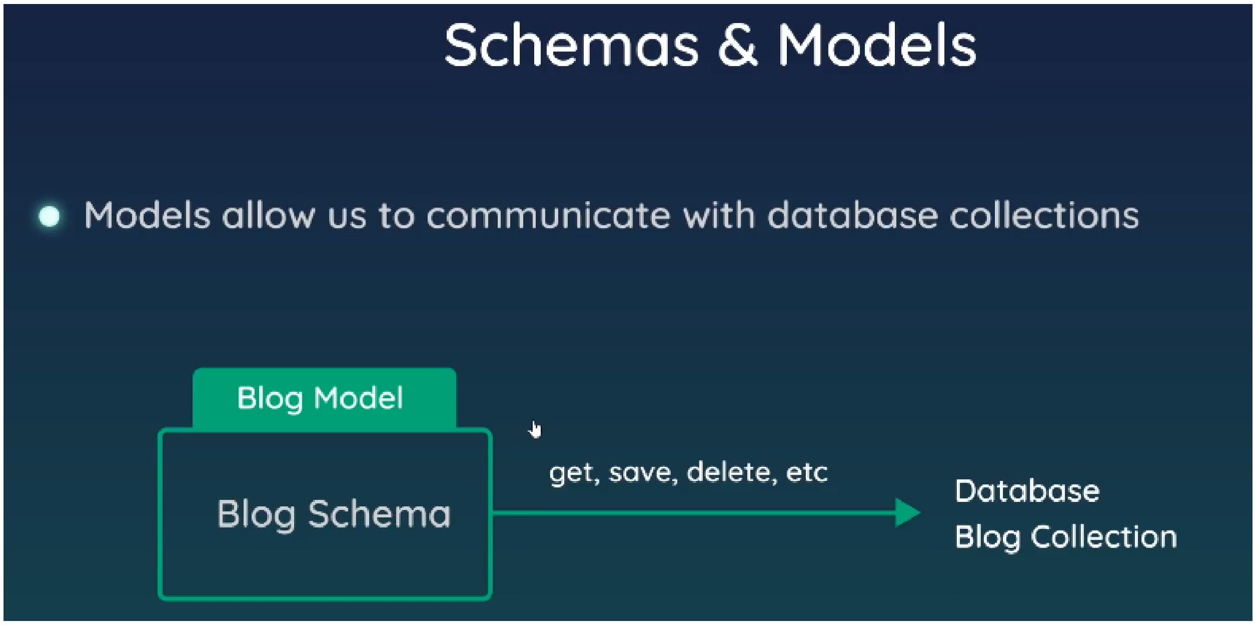


Schemas & Models

- Schemas defines the structure of a type of data / document
  - Properties & property types

User Schema:
- name (string), required
- age (number)
- bio (string), required

Blog Schema:
- title (string), required
- snippet (string), required
- body (string), required



Schemas & Models

- Models allow us to communicate with database collections

Blog Model

Blog Schema

get, save, delete, etc

Database
Blog Collection

Models →blog.js

blog.js

```
const mongoose = require('mongoose');
const Schema = mongoose.Schema;

// Schema is a structure
const blogSchema = new Schema({
    title: {
        type: String,
        required: true
    },
    snippet: {
        type: String,
        required: true
    },
    body: {
        type: String,
        required: true
    }
}, {timestamps: true });

// name of the model is Captial letter Here Blog
const Blog = mongoose.model('Blog', blogSchema);
module.exports = Blog;
```

Getting & Saving Data

→while executing the code if you see this kind of error
Following steps to resolve the error.

MongooseServerSelectionError: Could not connect to any servers in your MongoDB Atlas cluster. One common reason is that you're trying to access the database from an IP that isn't whitelisted. Make sure your current IP address is on

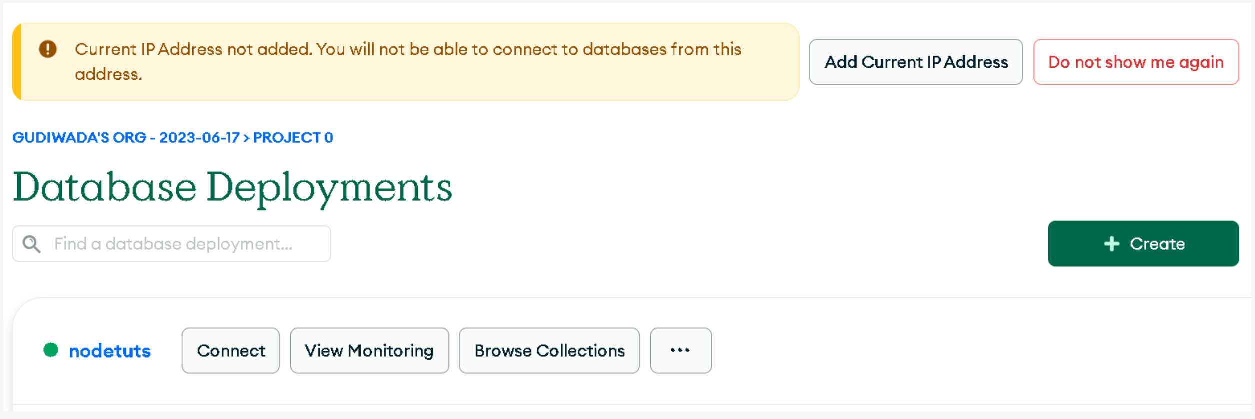
```
your Atlas cluster's IP whitelist:
https://www.mongodb.com/docs/atlas/security-whitelist/
    at _handleConnectionErrors
(C:\Users\gudiw\Documents\tuts\node-crash-course\node_modules\mongoose\lib\connection.js:792:11)
```

→If the database is not connecting then

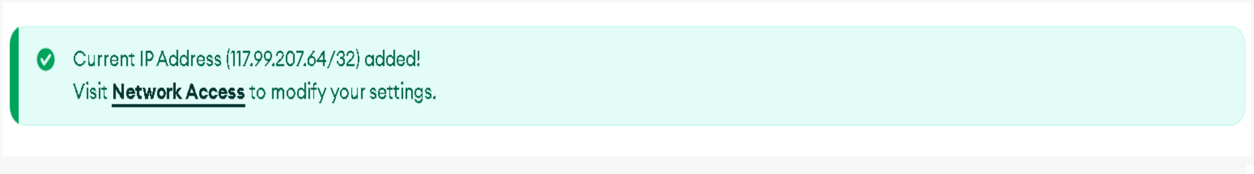
Here are the steps to whitelist your IP address in MongoDB Atlas:

1. **Open the MongoDB Atlas website and log in to your account.**
2. **Go to your project and navigate to the "Network Access" or "Security" section.**
3. **Find the IP Whitelist section and click on the "Add IP Address" button.**
4. **Choose "Add Your Current IP Address" to automatically add your current IP address to the whitelist. Alternatively, you can specify a specific IP address or range manually.**
5. **Save the changes and ensure that your IP address is now included in the whitelist.**

By whitelisting your IP address, you should be able to connect to your MongoDB Atlas cluster successfully without encountering the "MongooseServerSelectionError" related to IP whitelisting issues.



Current IP Address not added. You will not be able to connect to databases from this address.

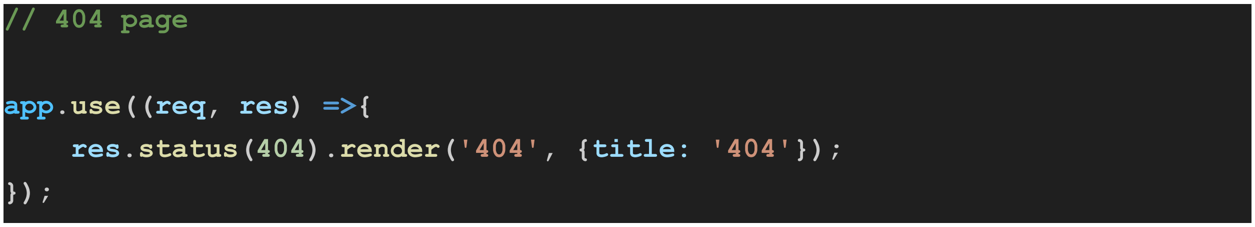


Current IP Address (117.99.207.64/32) added!
Visit **Network Access** to modify your settings.

app.js

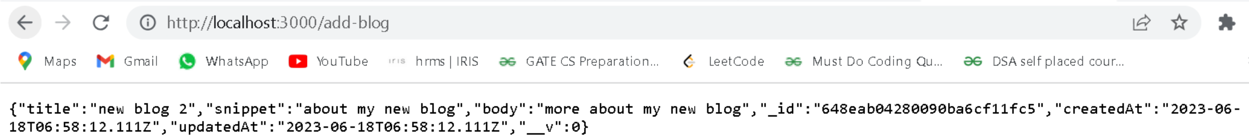
```
const express = require('express');
const morgan = require('morgan');
const mongoose = require('mongoose');
const Blog = require('./models/blog');

// express app
const app = express();

// connect to mongodb
const dbURI =
'mongodb+srv://netninja:test1234@nodetuts.1xigocw.mongodb.net/node-tuts?retryWrites=true&w=majority';

//mongoose.connect(dbURI, { useNewUrlParser: true, useUnifiedTopology:
true })  //if u see deprecationwarning then use this line

mongoose.connect(dbURI)
  //.then(result => console.log('connected to db'))
  .then(result => app.listen(3000))
  .catch((err) => console.log(err));

//register view engine
app.set('view engine', 'ejs');


// middleware & static files
app.use(express.static('public'));
app.use(morgan('dev'));
//app.use(morgan('tiny'));    //slight change
```

```
// mongoose and mondo sandbox routes
app.get('/add-blog', (req, res) =>{
    const blog = new Blog({
        title: 'new blog 2',
        snippet: 'about my new blog',
        body: 'more about my new blog'
    });

    // save documents to database
    blog.save()
     .then((result) => {
        res.send(result)
     })
     .catch((err) => {
        console.log(err);
     });

})

// routes
app.get('/', (req, res)=>{
    const blogs = [
        {title: 'Sandy finds eggs', snippet: 'Lorem ipsum dolor sit amet
consectetur'},
        {title: 'Raghava finds stars', snippet: 'Lorem ipsum dolor sit
amet consectetur'},
        {title: 'How to defeat bowser', snippet: 'Lorem ipsum dolor sit
amet consectetur'},
    ];
    res.render('index', {title: 'Home', blogs});
});

app.get('/about', (req, res)=>{
    res.render('about', {title: 'About'});
});

app.get('/blogs/create', (req, res)=>{
    res.render('create', {title: 'Create a new Blog'});
});
```



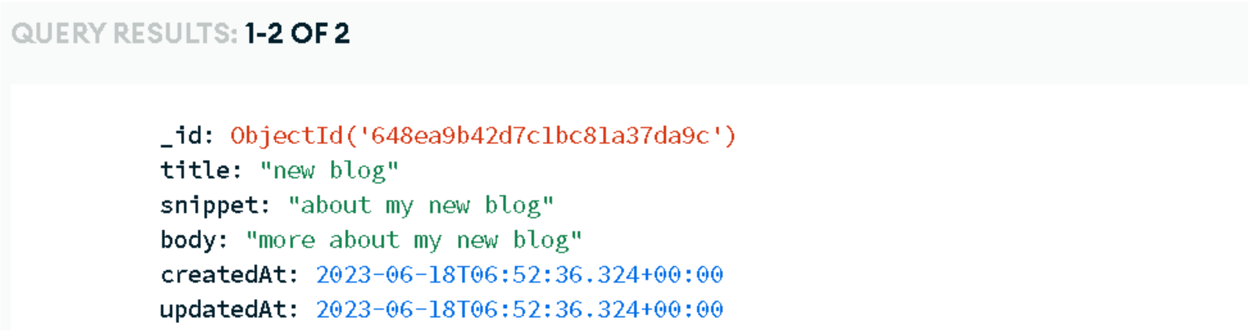
```
// 404 page

app.use((req, res) =>{
    res.status(404).render('404', {title: '404'});
});
```

O/P:



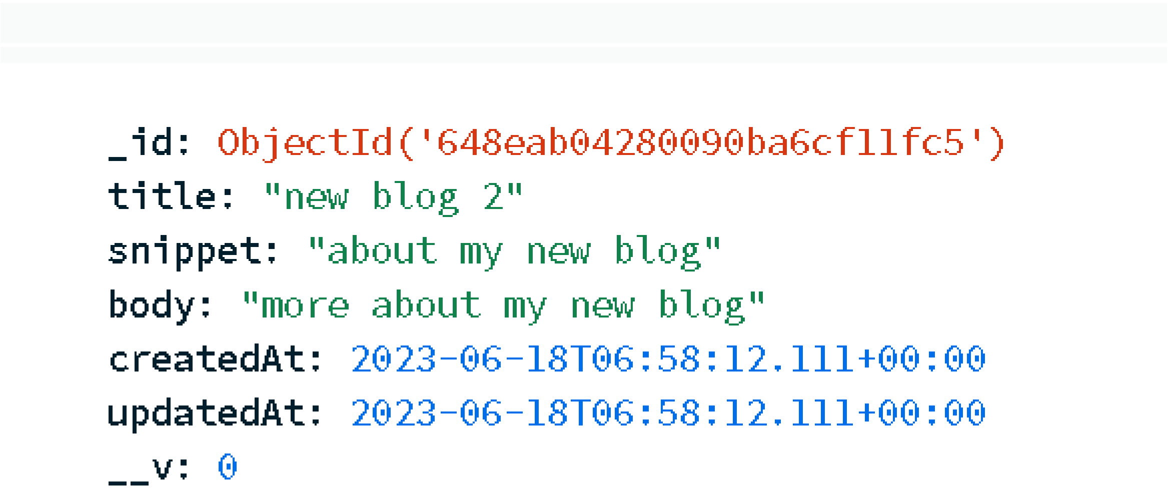
```
{"title":"new blog 2","snippet":"about my new blog","body":"more about my new blog","_id":"648eab04280090ba6cf11fc5","createdAt":"2023-06-18T06:58:12.111Z","updatedAt":"2023-06-18T06:58:12.111Z","__v":0}
```

Database mongodb



QUERY RESULTS: 1-2 OF 2

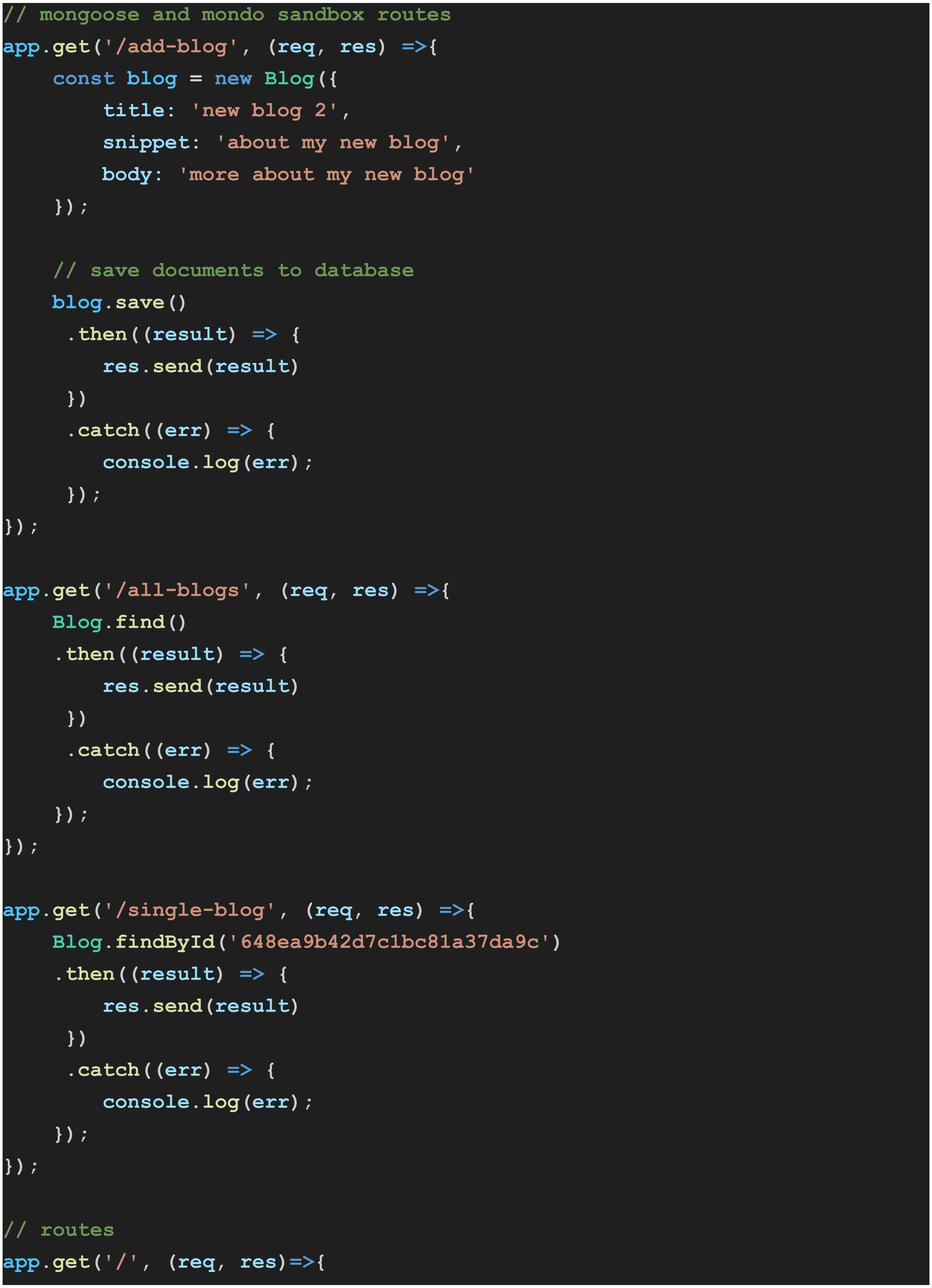
```
_id: ObjectId('648ea9b42d7c1bc81a37da9c')
title: "new blog"
snippet: "about my new blog"
body: "more about my new blog"
createdAt: 2023-06-18T06:52:36.324+00:00
updatedAt: 2023-06-18T06:52:36.324+00:00
```



```
_id: ObjectId('648eab04280090ba6cf11fc5')
title: "new blog 2"
snippet: "about my new blog"
body: "more about my new blog"
createdAt: 2023-06-18T06:58:12.111+00:00
updatedAt: 2023-06-18T06:58:12.111+00:00
__v: 0
```

app.js

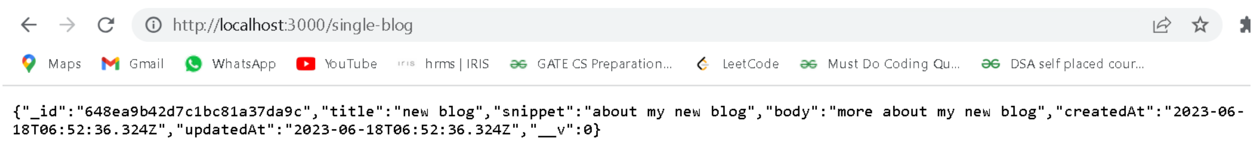
```
const express = require('express');
const morgan = require('morgan');
const mongoose = require('mongoose');
const Blog = require('./models/blog');

// express app
const app = express();

// connect to mongodb
const dbURI =
'mongodb+srv://netninja:test1234@nodetuts.1xigocw.mongodb.net/node-tuts?retryWrites=true&w=majority';

//mongoose.connect(dbURI, { useNewUrlParser: true, useUnifiedTopology: true })  //if u see deprecationwarning then use this line

mongoose.connect(dbURI)
  //.then(result => console.log('connected to db'))
  .then(result => app.listen(3000))
  .catch((err) => console.log(err));

//register view engine
app.set('view engine', 'ejs');


// middleware & static files
app.use(express.static('public'));
app.use(morgan('dev'));
//app.use(morgan('tiny'));    //slight change
```



```
// mongoose and mondo sandbox routes
app.get('/add-blog', (req, res) =>{
    const blog = new Blog({
        title: 'new blog 2',
        snippet: 'about my new blog',
        body: 'more about my new blog'
    });

    // save documents to database
    blog.save()
     .then((result) => {
        res.send(result)
     })
     .catch((err) => {
        console.log(err);
     });
});

app.get('/all-blogs', (req, res) =>{
    Blog.find()
    .then((result) => {
        res.send(result)
     })
     .catch((err) => {
        console.log(err);
    });
});

app.get('/single-blog', (req, res) =>{
    Blog.findById('648ea9b42d7c1bc81a37da9c')
    .then((result) => {
        res.send(result)
     })
     .catch((err) => {
        console.log(err);
    });
});

// routes
app.get('/', (req, res)=>{
    const blogs = [
        {title: 'Sandy finds eggs', snippet: 'Lorem ipsum dolor sit amet consectetur'},
        {title: 'Raghava finds stars', snippet: 'Lorem ipsum dolor sit amet consectetur'},
        {title: 'How to defeat bowser', snippet: 'Lorem ipsum dolor sit amet consectetur'},
    ];
    res.render('index', {title: 'Home', blogs});
});

app.get('/about', (req, res)=>{
    res.render('about', {title: 'About'});
});

app.get('/blogs/create', (req, res)=>{
    res.render('create', {title: 'Create a new Blog'});
});

// 404 page

app.use((req, res) =>{
    res.status(404).render('404', {title: '404'});
});
```

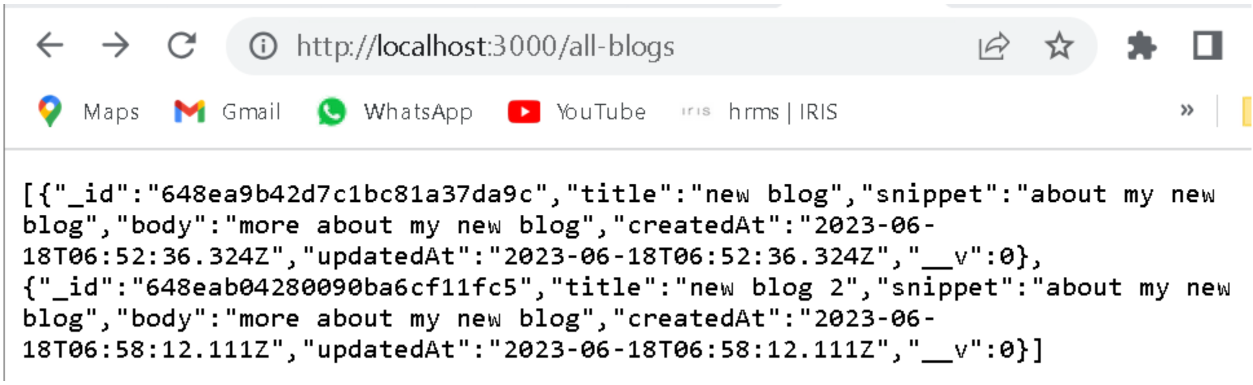
O/P:

single-blog



{"_id":"648ea9b42d7c1bc81a37da9c","title":"new blog","snippet":"about my new blog","body":"more about my new blog","createdAt":"2023-06-18T06:52:36.324Z","updatedAt":"2023-06-18T06:52:36.324Z","__v":0}

all-blogs

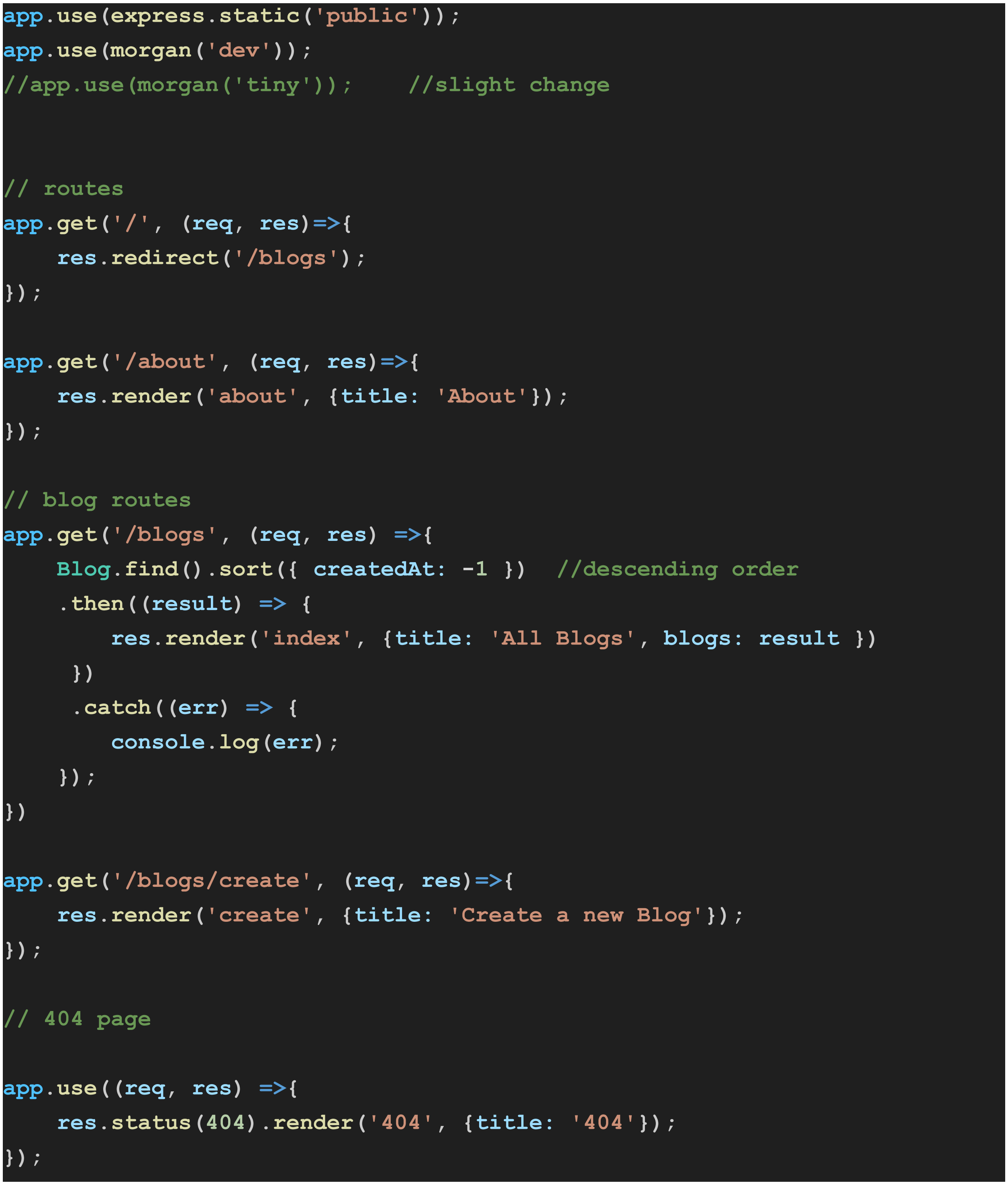


```
[{"_id":"648ea9b42d7c1bc81a37da9c","title":"new blog","snippet":"about my new
blog","body":"more about my new blog","createdAt":"2023-06-
18T06:52:36.324Z","updatedAt":"2023-06-18T06:52:36.324Z","__v":0},
{"_id":"648eab04280090ba6cf11fc5","title":"new blog 2","snippet":"about my new
blog","body":"more about my new blog","createdAt":"2023-06-
18T06:58:12.111Z","updatedAt":"2023-06-18T06:58:12.111Z","__v":0}]
```

Outputting Documents in Views

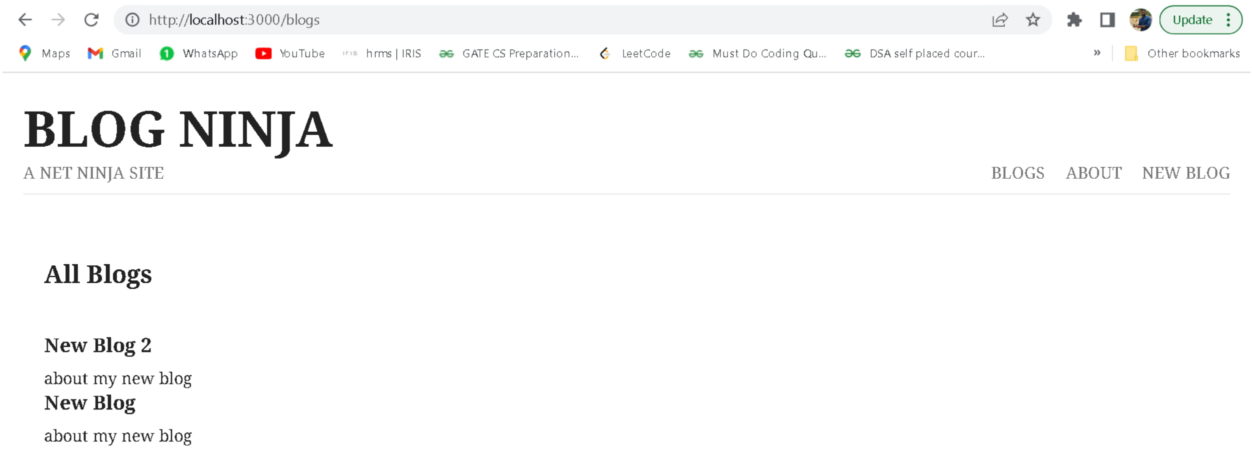
app.js

```
const express = require('express');
const morgan = require('morgan');
const mongoose = require('mongoose');
const Blog = require('./models/blog');

// express app
const app = express();

// connect to mongodb
const dbURI =
'mongodb+srv://netninja:test1234@nodetuts.1xigocw.mongodb.net/node-tuts?re
tryWrites=true&w=majority';

//mongoose.connect(dbURI, { useNewUrlParser: true, useUnifiedTopology:
true })  //if u see deprecationwarning then use this line

mongoose.connect(dbURI)
  //.then(result => console.log('connected to db'))
  .then(result => app.listen(3000))
  .catch((err) => console.log(err));

//register view engine
app.set('view engine', 'ejs');


// middleware & static files
```



```
app.use(express.static('public'));
app.use(morgan('dev'));
//app.use(morgan('tiny'));    //slight change


// routes
app.get('/', (req, res)=>{
    res.redirect('/blogs');
});

app.get('/about', (req, res)=>{
    res.render('about', {title: 'About'});
});

// blog routes
app.get('/blogs', (req, res) =>{
    Blog.find().sort({ createdAt: -1 })  //descending order
    .then((result) => {
        res.render('index', {title: 'All Blogs', blogs: result })
     })
     .catch((err) => {
        console.log(err);
    });
})

app.get('/blogs/create', (req, res)=>{
    res.render('create', {title: 'Create a new Blog'});
});

// 404 page

app.use((req, res) =>{
    res.status(404).render('404', {title: '404'});
});
```

O/P:



BLOG NINJA

A NET NINJA SITE

All Blogs

New Blog 2

about my new blog

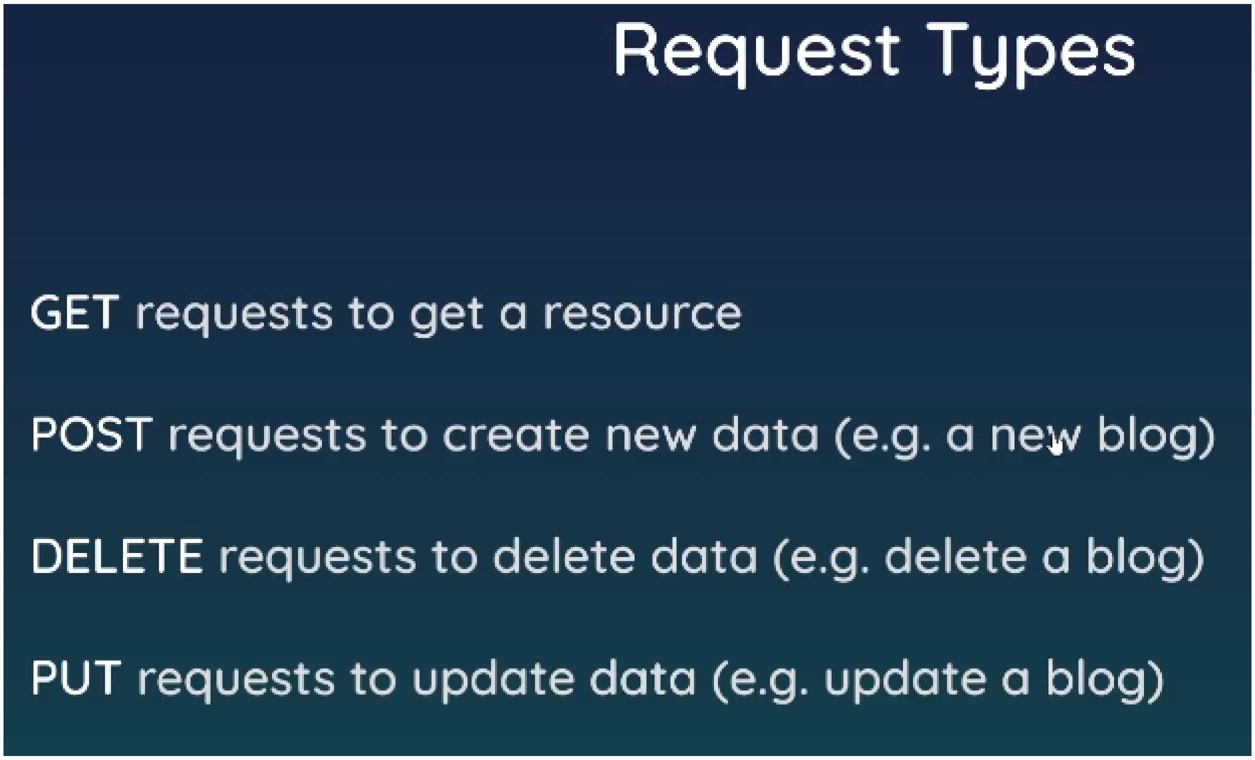
New Blog

about my new blog

Lesson 10

Get, Post & Delete Requests

Request Types



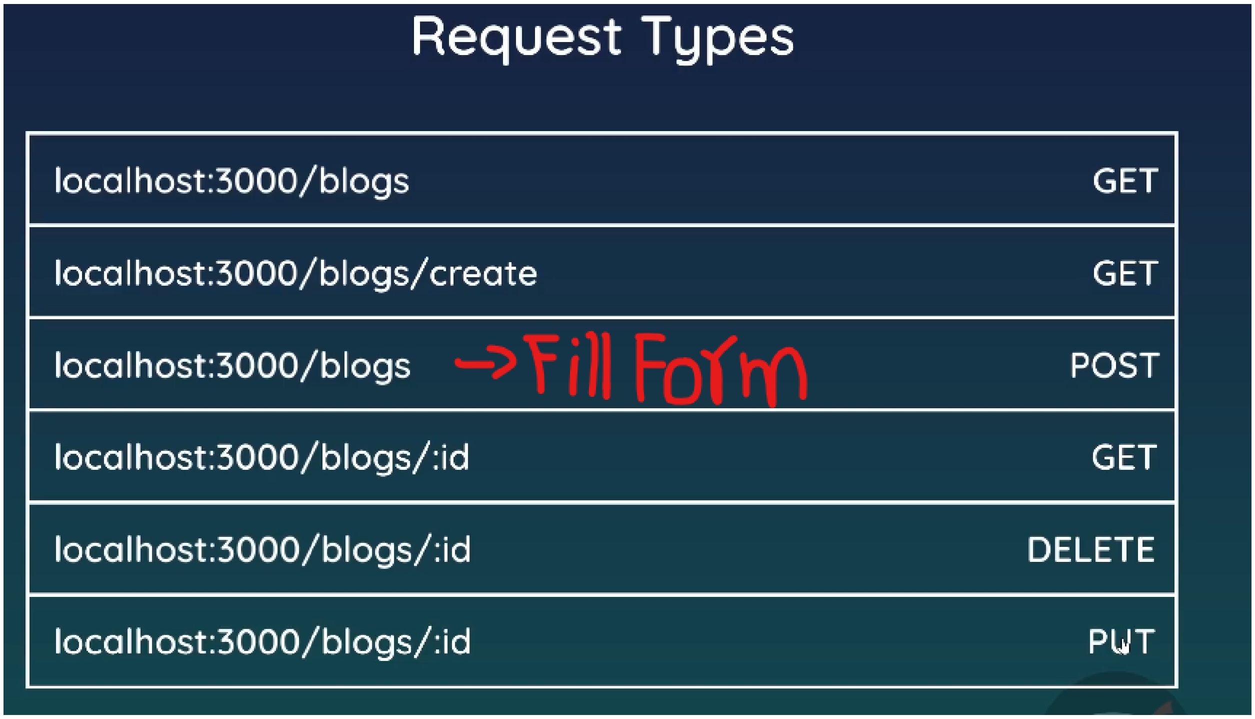
Request Types

GET requests to get a resource

POST requests to create new data (e.g. a new blog)

DELETE requests to delete data (e.g. delete a blog)

PUT requests to update data (e.g. update a blog)



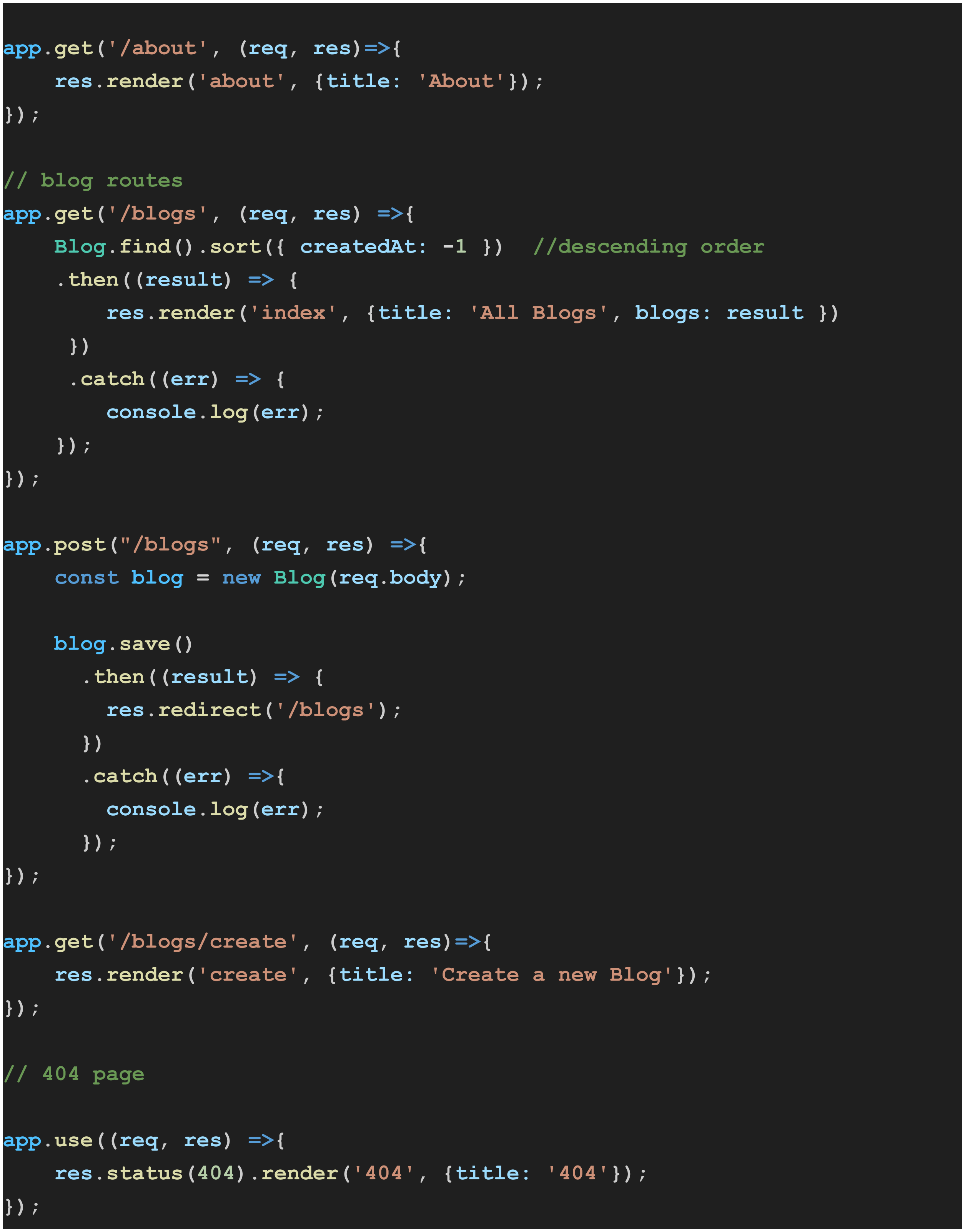
Request Types

localhost:3000/blogs	GET
localhost:3000/blogs/create	GET
localhost:3000/blogs ->Fill Form	POST
localhost:3000/blogs/:id	GET
localhost:3000/blogs/:id	DELETE
localhost:3000/blogs/:id	PUT

Post Request

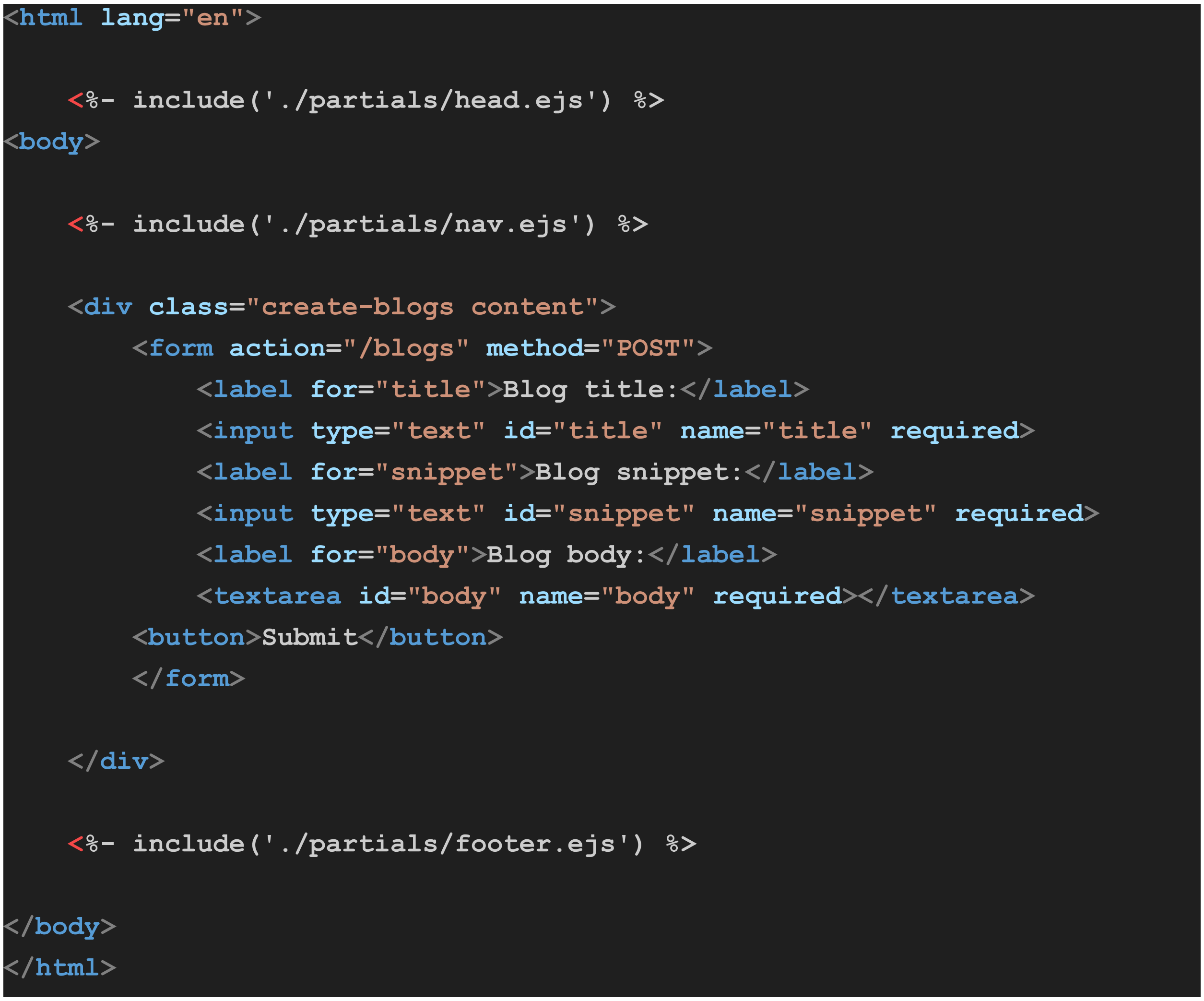
app.js

```
const express = require('express');
const morgan = require('morgan');
const mongoose = require('mongoose');
const Blog = require('./models/blog');

// express app
const app = express();

// connect to mongodb
const dbURI =
'mongodb+srv://netninja:test1234@nodetuts.1xigocw.mongodb.net/node-tuts?re
tryWrites=true&w=majority';

//mongoose.connect(dbURI, { useNewUrlParser: true, useUnifiedTopology:
true })  //if u see deprecationwarning then use this line

mongoose.connect(dbURI)
  //.then(result => console.log('connected to db'))
  .then(result => app.listen(3000))
  .catch((err) => console.log(err));

//register view engine
app.set('view engine', 'ejs');


// middleware & static files
app.use(express.static('public'));
app.use(express.urlencoded({extended: true}));
app.use(morgan('dev'));
//app.use(morgan('tiny'));    //slight change


// routes
app.get('/', (req, res)=>{
    res.redirect('/blogs');
});
```



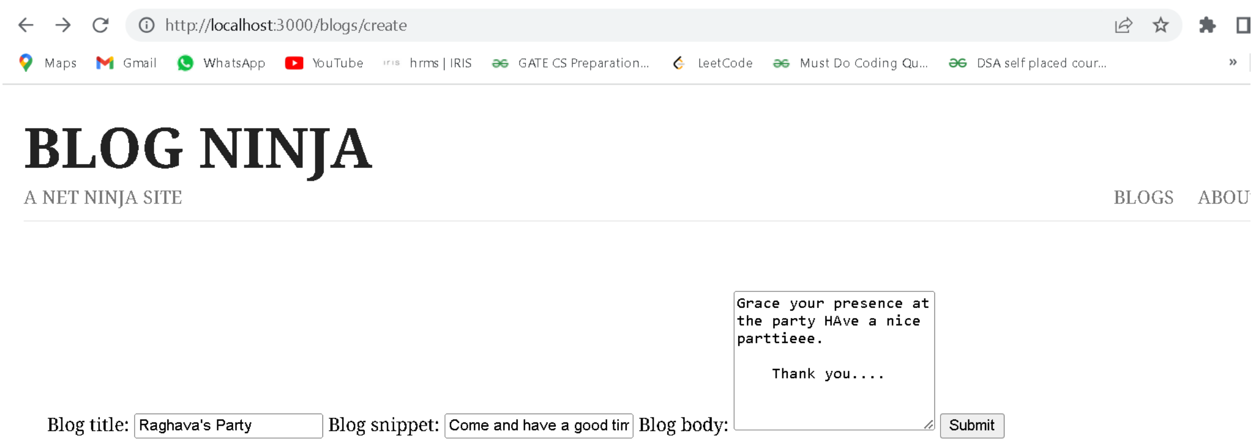
```
app.get('/about', (req, res)=>{
    res.render('about', {title: 'About'});
});

// blog routes
app.get('/blogs', (req, res) =>{
    Blog.find().sort({ createdAt: -1 })  //descending order
    .then((result) => {
        res.render('index', {title: 'All Blogs', blogs: result })
     })
     .catch((err) => {
        console.log(err);
    });
});

app.post("/blogs", (req, res) =>{
    const blog = new Blog(req.body);

    blog.save()
      .then((result) => {
        res.redirect('/blogs');
      })
      .catch((err) =>{
        console.log(err);
      });
});

app.get('/blogs/create', (req, res)=>{
    res.render('create', {title: 'Create a new Blog'});
});

// 404 page

app.use((req, res) =>{
    res.status(404).render('404', {title: '404'});
});
```

create.js



```
<html lang="en">

    <%- include('./partials/head.ejs') %>
<body>

    <%- include('./partials/nav.ejs') %>

    <div class="create-blogs content">
        <form action="/blogs" method="POST">
            <label for="title">Blog title:</label>
            <input type="text" id="title" name="title" required>
            <label for="snippet">Blog snippet:</label>
            <input type="text" id="snippet" name="snippet" required>
            <label for="body">Blog body:</label>
            <textarea id="body" name="body" required></textarea>
        <button>Submit</button>
        </form>

    </div>

    <%- include('./partials/footer.ejs') %>

</body>
</html>
```

O/P:

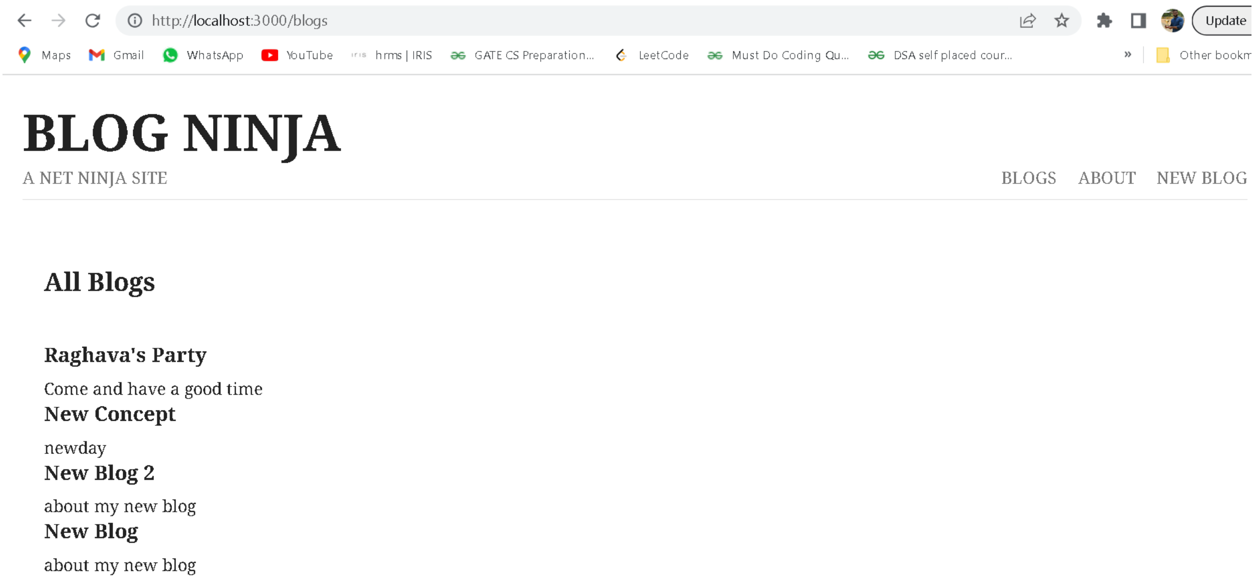


BLOG NINJA

A NET NINJA SITE

Blog title: Raghava's Party Blog snippet: Come and have a good tim Blog body:

```
Grace your presence at
the party HAve a nice
parttieee.

    Thank you....
```



BLOG NINJA

A NET NINJA SITE

BLOGS ABOUT NEW BLOG

All Blogs

Raghava's Party

Come and have a good time

New Concept

newday

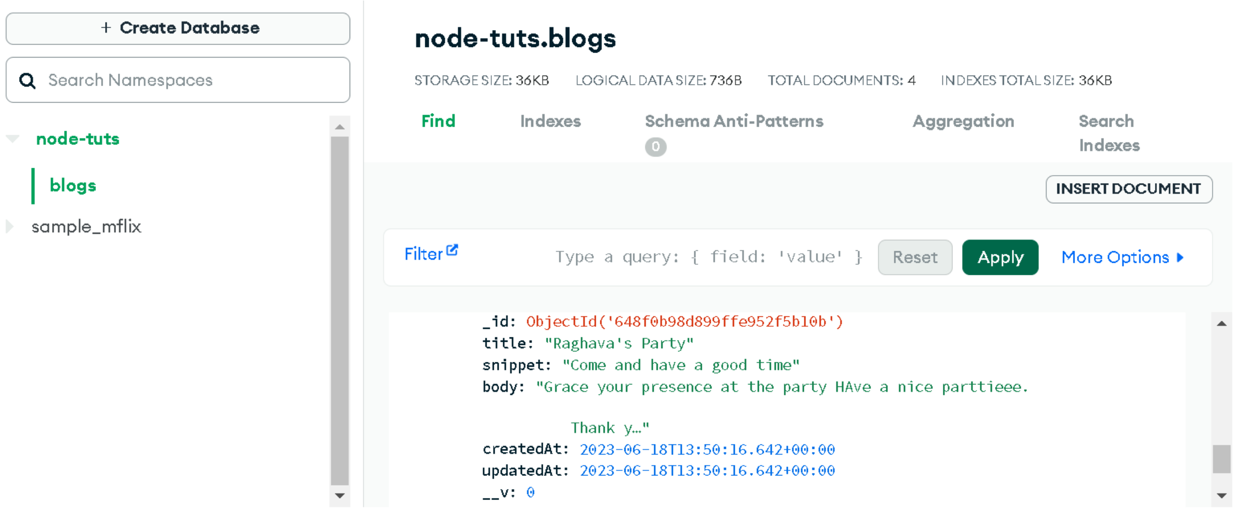
New Blog 2

about my new blog

New Blog

about my new blog

MongoDB

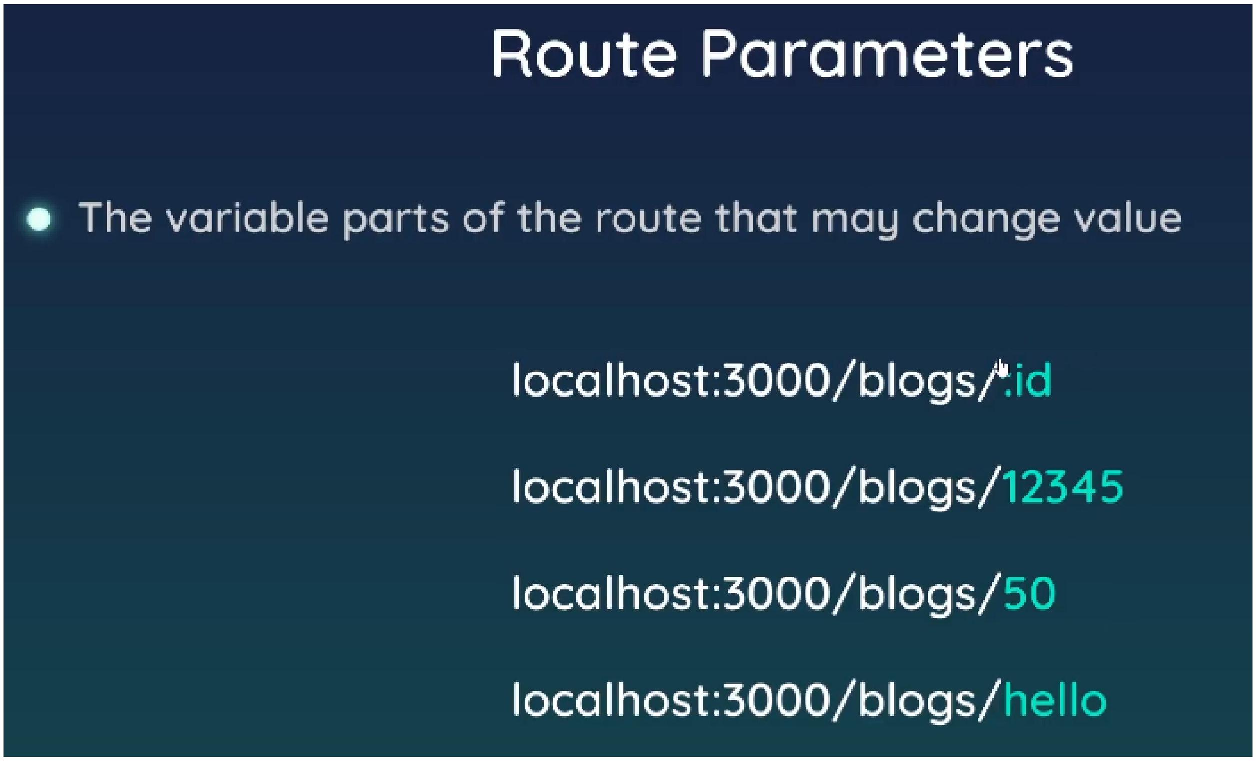


node-tuts.blogs

STORAGE SIZE: 36KB LOGICAL DATA SIZE: 736B TOTAL DOCUMENTS: 4 INDEXES TOTAL SIZE: 36KB

```
_id: ObjectId('648f0b98d899ffe952f5b10b')
title: "Raghava's Party"
snippet: "Come and have a good time"
body: "Grace your presence at the party HAve a nice parttieee.

          Thank y…"
createdAt: 2023-06-18T13:50:16.642+00:00
updatedAt: 2023-06-18T13:50:16.642+00:00
__v: 0
```

Route Parameters



Route Parameters

- The variable parts of the route that may change value

localhost:3000/blogs/:id

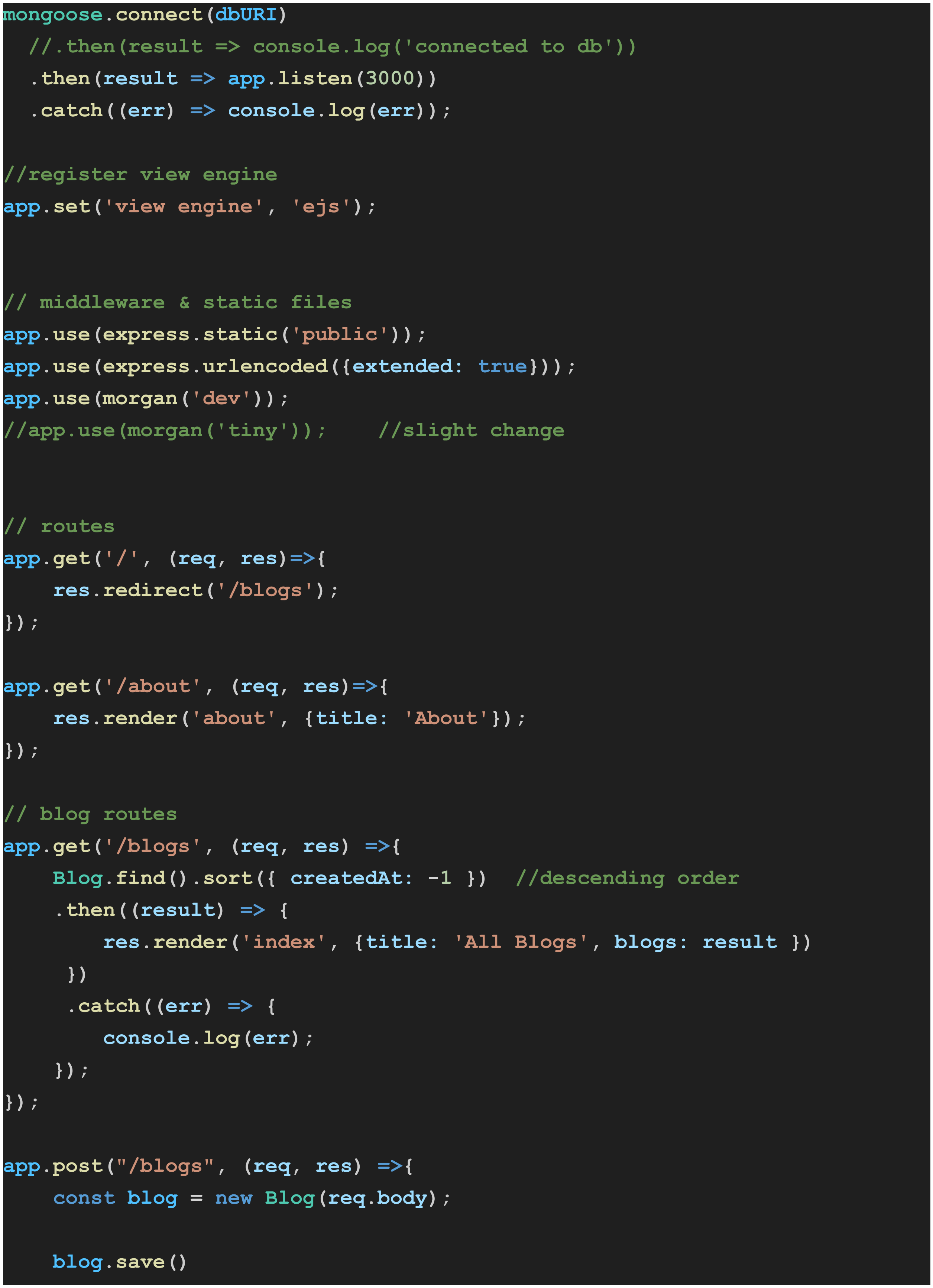
localhost:3000/blogs/12345

localhost:3000/blogs/50

localhost:3000/blogs/hello

app.js

```
const express = require('express');
const morgan = require('morgan');
const mongoose = require('mongoose');
const Blog = require('./models/blog');

// express app
const app = express();

// connect to mongodb
const dbURI =
'mongodb+srv://netninja:test1234@nodetuts.1xigocw.mongodb.net/node-tuts?retryWrites=true&w=majority';

//mongoose.connect(dbURI, { useNewUrlParser: true, useUnifiedTopology: true })  //if u see deprecationwarning then use this line
```



```
mongoose.connect(dbURI)
  //.then(result => console.log('connected to db'))
  .then(result => app.listen(3000))
  .catch((err) => console.log(err));

//register view engine
app.set('view engine', 'ejs');


// middleware & static files
app.use(express.static('public'));
app.use(express.urlencoded({extended: true}));
app.use(morgan('dev'));
//app.use(morgan('tiny'));    //slight change


// routes
app.get('/', (req, res)=>{
    res.redirect('/blogs');
});

app.get('/about', (req, res)=>{
    res.render('about', {title: 'About'});
});

// blog routes
app.get('/blogs', (req, res) =>{
    Blog.find().sort({ createdAt: -1 })  //descending order
    .then((result) => {
        res.render('index', {title: 'All Blogs', blogs: result })
     })
     .catch((err) => {
        console.log(err);
    });
});

app.post("/blogs", (req, res) =>{
    const blog = new Blog(req.body);

    blog.save()
```

```
      .then((result) => {
        res.redirect('/blogs');
      })
      .catch((err) =>{
        console.log(err);
      });
});

app.get('/blogs/:id', (req, res) =>{
    const id = req.params.id;
    Blog.findById(id)
      .then((result) =>{
        res.render('details', { blog: result, title: 'Blog Details' });
      })
      .catch((err)=>{
        console.log(err);
      });
});

app.get('/blogs/create', (req, res)=>{
    res.render('create', {title: 'Create a new Blog'});
});

// 404 page

app.use((req, res) =>{
    res.status(404).render('404', {title: '404'});
});
```

details.ejs

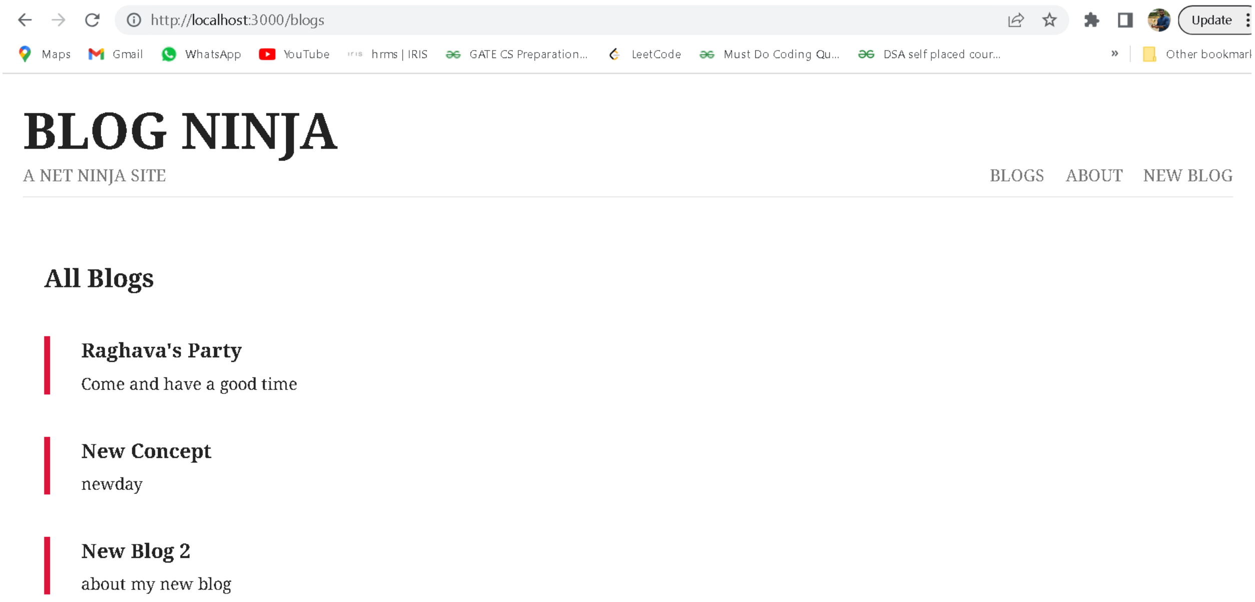
```
<html lang="en">
<%- include("./partials/head.ejs") %>

<body>
  <%- include("./partials/nav.ejs") %>

  <div class="details content">
    <h2><%= blog.title %></h2>
    <div class="content">
      <p><%= blog.body %></p>
    </div>

  </div>

  <%- include("./partials/footer.ejs") %>

</body>
</html>
```

style.css

→Only extra added for index styles

```
/* index styles */
        .blogs a{
          display: block;
          margin: 40px 0;
          padding-left: 30px;
          border-left: 6px solid crimson;
        }
        .blogs a:hover h3{
          color: crimson;
        }
```

O/P:



BLOG NINJA

A NET NINJA SITE

BLOGS ABOUT NEW BLOG

All Blogs

Raghava's Party

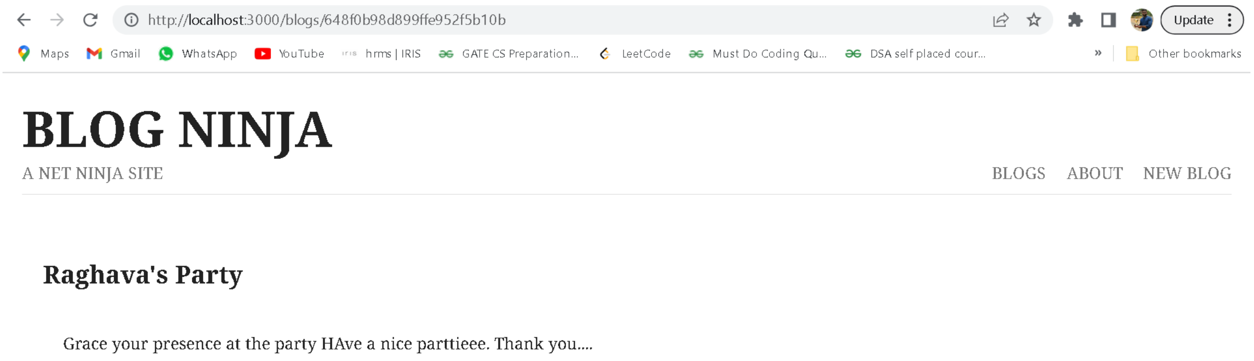
Come and have a good time

New Concept

newday

New Blog 2

about my new blog



BLOG NINJA

A NET NINJA SITE

BLOGS ABOUT NEW BLOG

Raghava's Party

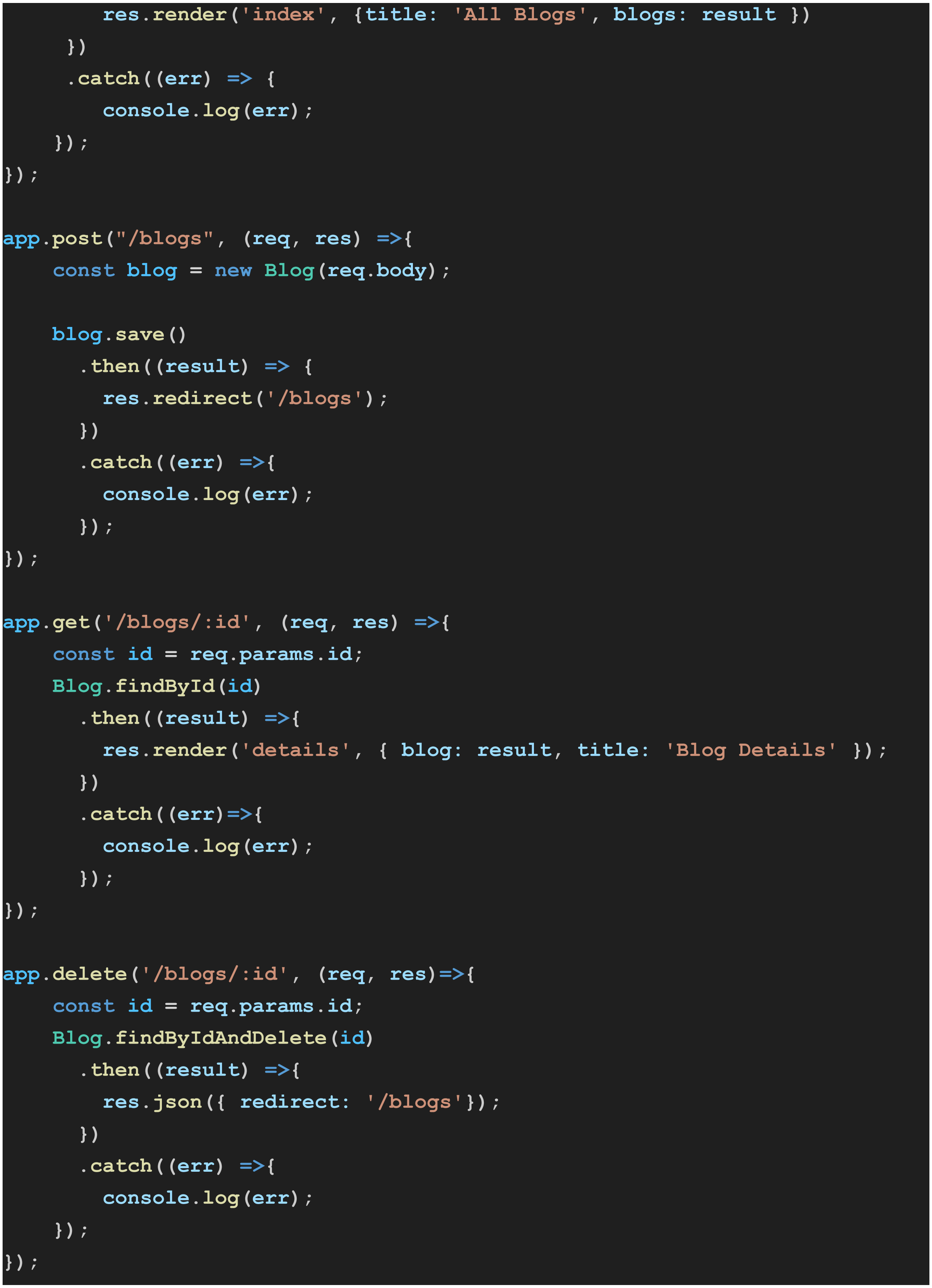
Grace your presence at the party HAve a nice parttieee. Thank you....

Delete Requests

app.js

```
const express = require('express');
const morgan = require('morgan');
const mongoose = require('mongoose');
const Blog = require('./models/blog');
```

```
// express app
const app = express();

// connect to mongodb
const dbURI =
'mongodb+srv://netninja:test1234@nodetuts.1xigocw.mongodb.net/node-tuts?re
tryWrites=true&w=majority';

//mongoose.connect(dbURI, { useNewUrlParser: true, useUnifiedTopology:
true })  //if u see deprecationwarning then use this line

mongoose.connect(dbURI)
  //.then(result => console.log('connected to db'))
  .then(result => app.listen(3000))
  .catch((err) => console.log(err));

//register view engine
app.set('view engine', 'ejs');


// middleware & static files
app.use(express.static('public'));
app.use(express.urlencoded({extended: true}));
app.use(morgan('dev'));
//app.use(morgan('tiny'));    //slight change


// routes
app.get('/', (req, res)=>{
    res.redirect('/blogs');
});

app.get('/about', (req, res)=>{
    res.render('about', {title: 'About'});
});

// blog routes
app.get('/blogs', (req, res) =>{
    Blog.find().sort({ createdAt: -1 })  //descending order
    .then((result) => {
```



```
        res.render('index', {title: 'All Blogs', blogs: result })
     })
     .catch((err) => {
        console.log(err);
    });
});

app.post("/blogs", (req, res) =>{
    const blog = new Blog(req.body);

    blog.save()
      .then((result) => {
        res.redirect('/blogs');
      })
      .catch((err) =>{
        console.log(err);
      });
});

app.get('/blogs/:id', (req, res) =>{
    const id = req.params.id;
    Blog.findById(id)
      .then((result) =>{
        res.render('details', { blog: result, title: 'Blog Details' });
      })
      .catch((err)=>{
        console.log(err);
      });
});

app.delete('/blogs/:id', (req, res)=>{
    const id = req.params.id;
    Blog.findByIdAndDelete(id)
      .then((result) =>{
        res.json({ redirect: '/blogs'});
      })
      .catch((err) =>{
        console.log(err);
    });
});
```

```
app.get('/blogs/create', (req, res)=>{
    res.render('create', {title: 'Create a new Blog'});
});

// 404 page

app.use((req, res) =>{
    res.status(404).render('404', {title: '404'});
});
```

details.ejs

```
<html lang="en">
<%- include("./partials/head.ejs") %>

<body>
  <%- include("./partials/nav.ejs") %>

  <div class="details content">
    <h2><%= blog.title %></h2>
    <div class="content">
      <p><%= blog.body %></p>
    </div>

    <a class="delete" data-doc="<%= blog._id %>">delete</a>
  </div>

  <%- include("./partials/footer.ejs") %>

  <script>
    const trashcan = document.querySelector('a.delete');

    trashcan.addEventListener('click', (e) => {
      const endpoint = `/blogs/${trashcan.dataset.doc}`;

      fetch(endpoint, {
        method: 'DELETE',
      })
      .then((response) => response.json())
      .then((data) => window.location.href = data.redirect)
      .catch((err) => console.log(err));
    });

  </script>
</body>
</html>
```

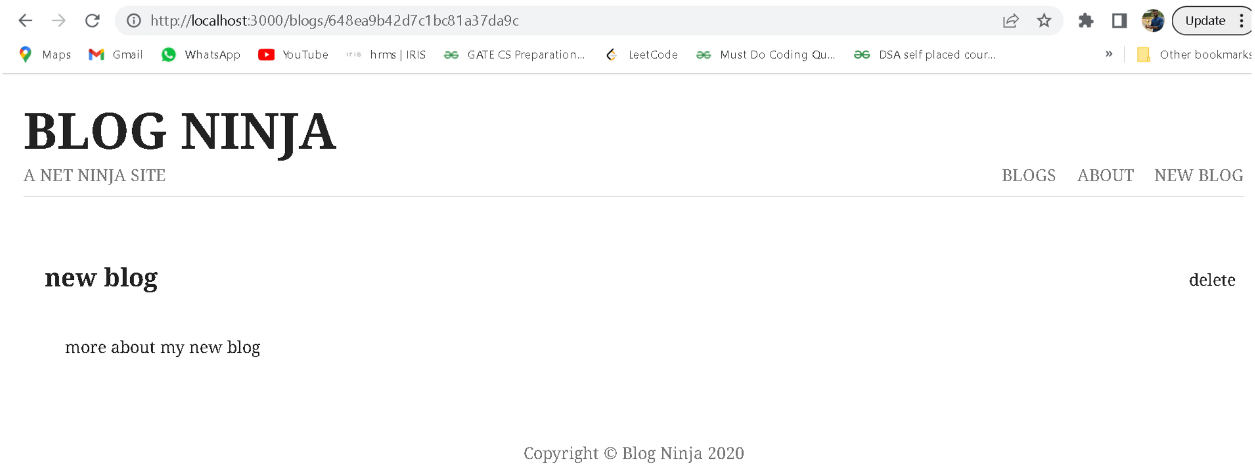
style.css

→Only extra added for details stye

```
/* details styles */
        .details{
          position: relative;
        }
        .delete{
          position: absolute;
          top: 0;
          right: 0;
          border-radius: 50%;
          padding: 8px;
        }
        .delete:hover{
          cursor: pointer;
          box-shadow: 1px 2px 3px rgba(0,0,0,0.2);
        }
```

O/P:

→Delete operation is working



BLOG NINJA

A NET NINJA SITE

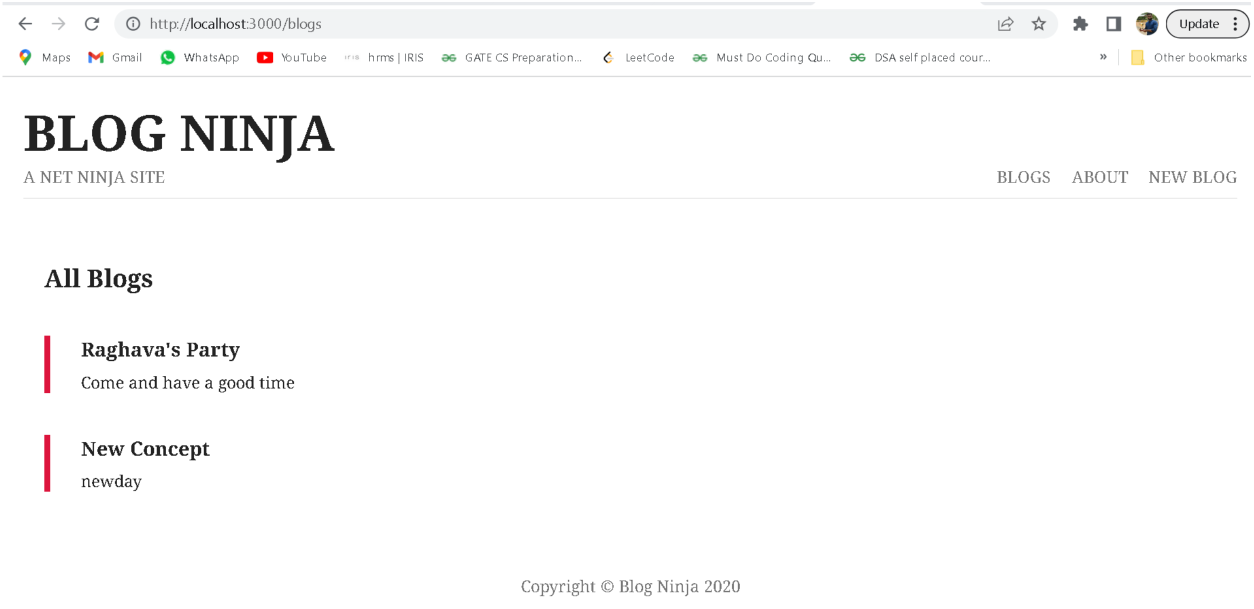
BLOGS ABOUT NEW BLOG

new blog

delete

more about my new blog

Copyright © Blog Ninja 2020



BLOG NINJA

A NET NINJA SITE

BLOGS ABOUT NEW BLOG

All Blogs

Raghava's Party

Come and have a good time

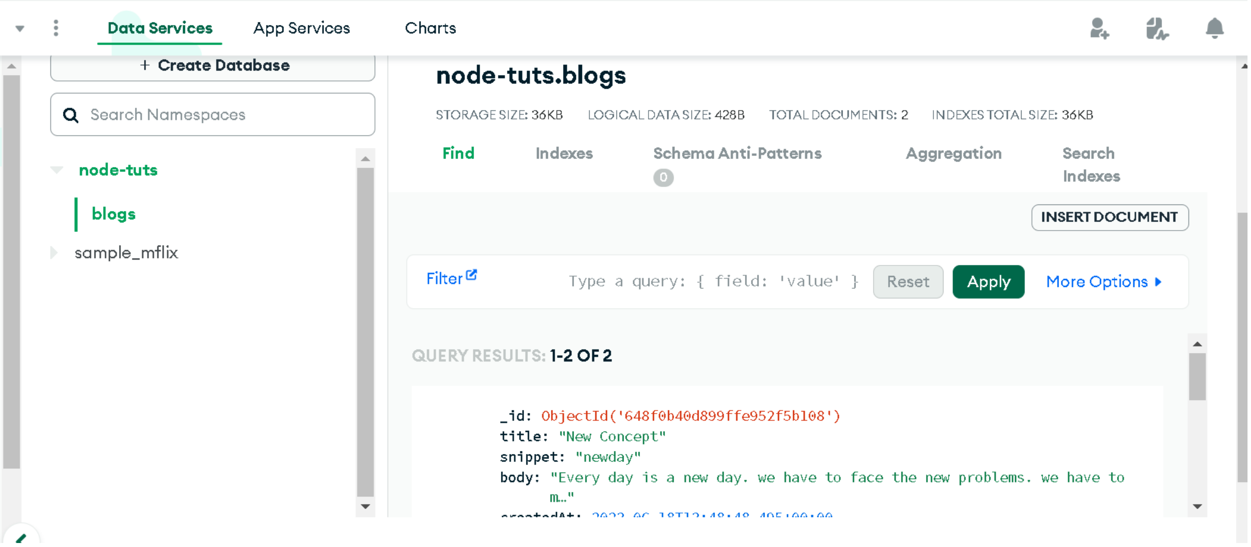
New Concept

newday

Copyright © Blog Ninja 2020

→MongoDb

Initially 4 out 4 are available after the deletion we have only 2



node-tuts.blogs

STORAGE SIZE: 36KB LOGICAL DATA SIZE: 428B TOTAL DOCUMENTS: 2 INDEXES TOTAL SIZE: 36KB

QUERY RESULTS: 1-2 OF 2

```
_id: ObjectId('648f0b40d899ffe952f5b108')
title: "New Concept"
snippet: "newday"
body: "Every day is a new day. we have to face the new problems. we have to
      m…"
createdAt: 2023-06-18T13:48:48.495+00:00
```

Lesson 11

Express Router & MVC

Express:

app.js

```
const express = require('express');
const morgan = require('morgan');
const mongoose = require('mongoose');
const blogRoutes = require('./routes/blogRoutes');

// express app
const app = express();

// connect to mongodb
const dbURI =
'mongodb+srv://netninja:test1234@nodetuts.1xigocw.mongodb.net/node-tuts?retryWrites=true&w=majority';

//mongoose.connect(dbURI, { useNewUrlParser: true, useUnifiedTopology: true })  //if u see deprecationwarning then use this line

mongoose.connect(dbURI)
  //.then(result => console.log('connected to db'))
  .then(result => app.listen(3000))
  .catch((err) => console.log(err));

//register view engine
app.set('view engine', 'ejs');


// middleware & static files
app.use(express.static('public'));
app.use(express.urlencoded({extended: true}));
app.use(morgan('dev'));
//app.use(morgan('tiny'));    //slight change


// routes
app.get('/', (req, res)=>{
    res.redirect('/blogs');
});
```

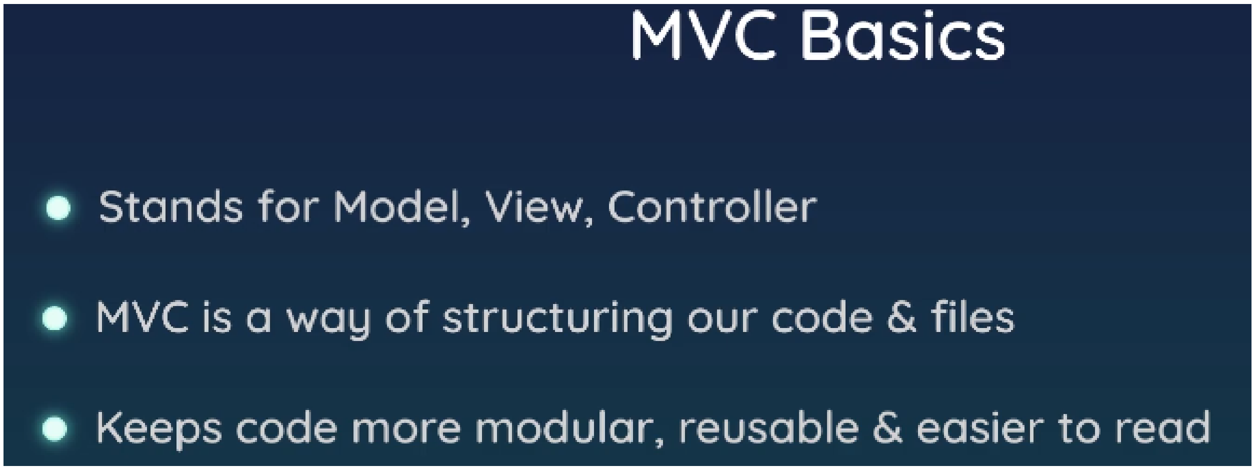
```
app.get('/about', (req, res)=>{
    res.render('about', {title: 'About'});
});

// blog routes
app.use('/blogs', blogRoutes);

// 404 page
app.use((req, res) =>{
    res.status(404).render('404', {title: '404'});
});
```

routes →blogRoutes.js

blogRoutes.js

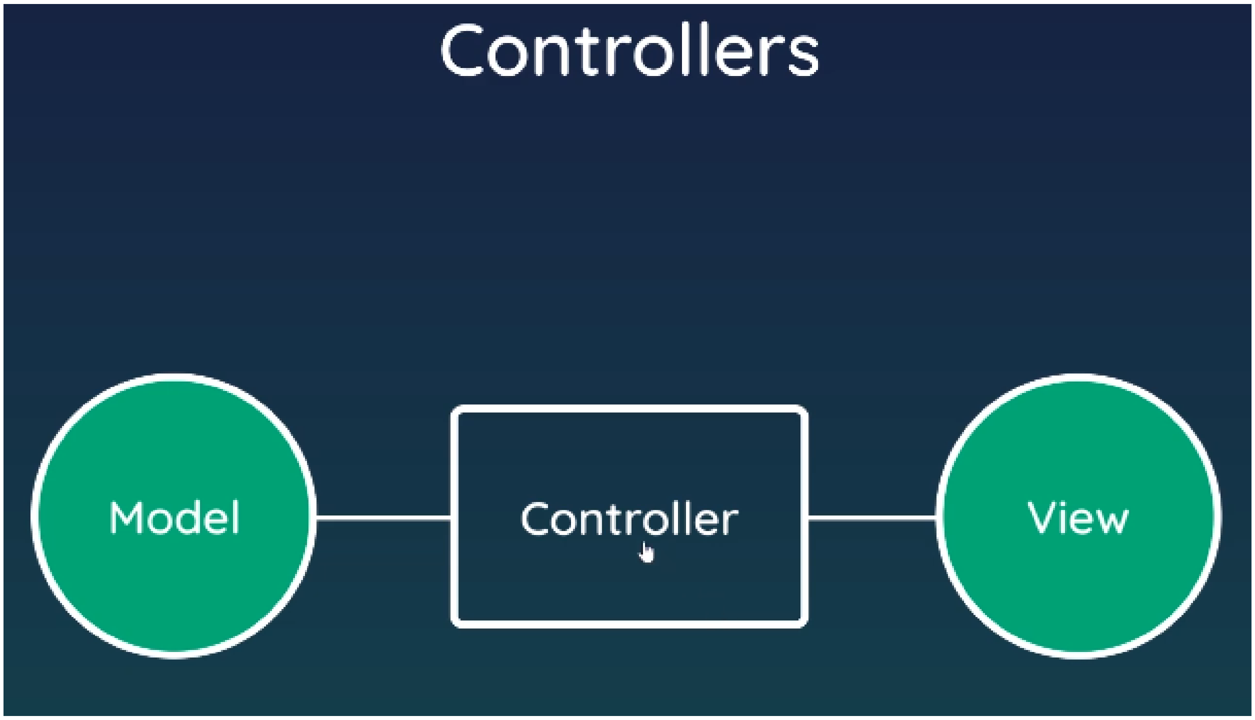
```
const express = require('express');
const Blog = require('../models/blog');

const router = express.Router();

// blog routes
router.get('/', (req, res) =>{
    Blog.find().sort({ createdAt: -1 })  //descending order
    .then((result) => {
        res.render('index', {title: 'All Blogs', blogs: result })
     })
     .catch((err) => {
        console.log(err);
    });
});

router.post('/', (req, res) =>{
    const blog = new Blog(req.body);

    blog.save()
      .then((result) => {
        res.redirect('/blogs');
      })
      .catch((err) =>{
        console.log(err);
      });
});

router.get('/create', (req, res)=>{
  res.render('create', {title: 'Create a new Blog'});
});

router.get('/:id', (req, res) =>{
    const id = req.params.id;
    Blog.findById(id)
      .then((result) =>{
        res.render('details', { blog: result, title: 'Blog Details' });
      })
      .catch((err)=>{
        console.log(err);
      });
});

router.delete('/:id', (req, res)=>{
    const id = req.params.id;
    Blog.findByIdAndDelete(id)
      .then((result) =>{
        res.json({ redirect: '/blogs'});
      })
      .catch((err) =>{
        console.log(err);
    });
});


module.exports = router;
```

MVC Basics



MVC Basics

- Stands for Model, View, Controller
- MVC is a way of structuring our code & files
- Keeps code more modular, reusable & easier to read

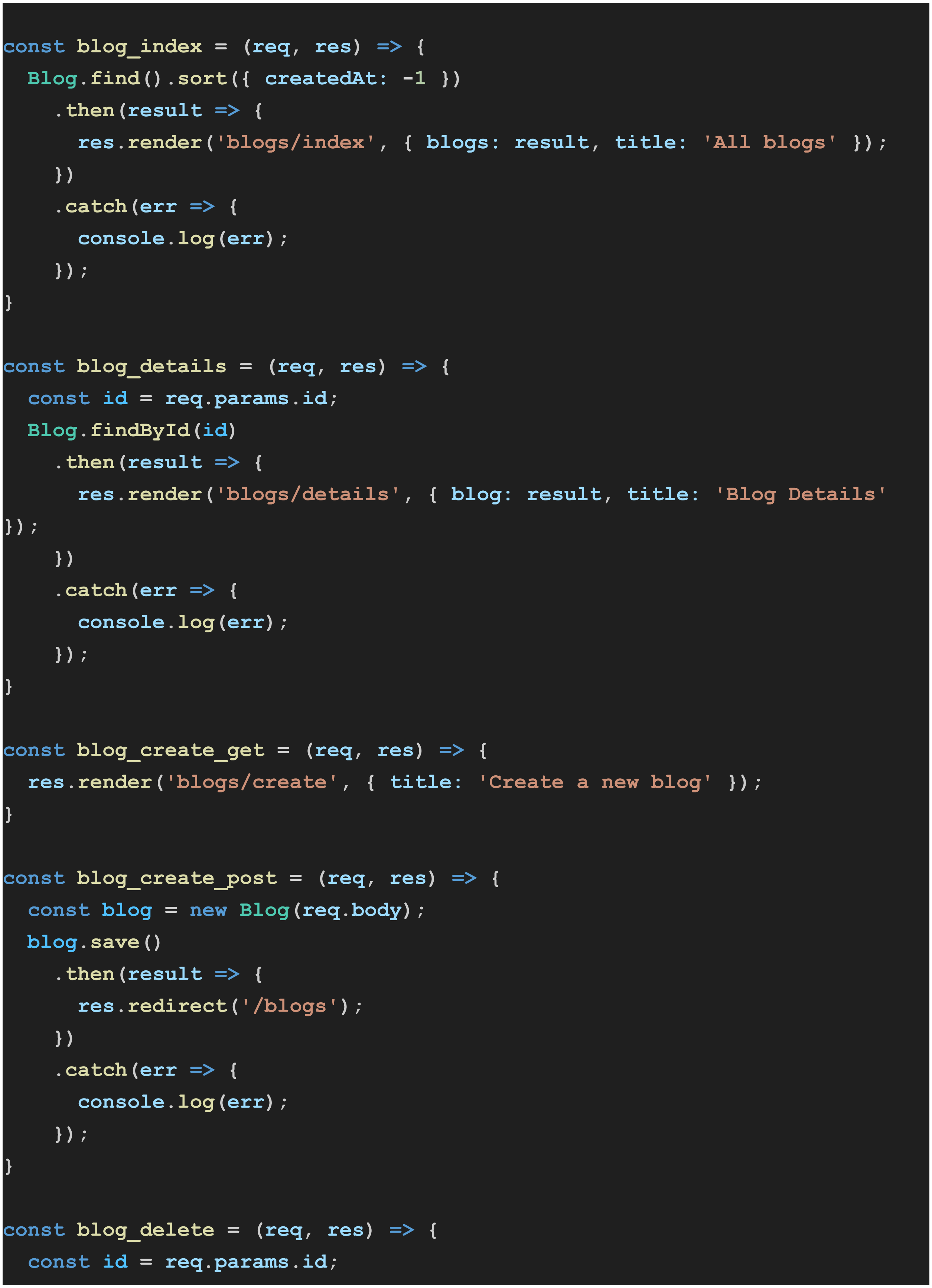


Controllers

Controllers →blogController.js

blogController.js

```
// blog_index, blog_details, blog_create_get, blog_create_post, blog_delete

const Blog = require('../models/blog');
```



```
const blog_index = (req, res) => {
  Blog.find().sort({ createdAt: -1 })
    .then(result => {
      res.render('blogs/index', { blogs: result, title: 'All blogs' });
    })
    .catch(err => {
      console.log(err);
    });
}

const blog_details = (req, res) => {
  const id = req.params.id;
  Blog.findById(id)
    .then(result => {
      res.render('blogs/details', { blog: result, title: 'Blog Details'
});
    })
    .catch(err => {
      console.log(err);
    });
}

const blog_create_get = (req, res) => {
  res.render('blogs/create', { title: 'Create a new blog' });
}

const blog_create_post = (req, res) => {
  const blog = new Blog(req.body);
  blog.save()
    .then(result => {
      res.redirect('/blogs');
    })
    .catch(err => {
      console.log(err);
    });
}

const blog_delete = (req, res) => {
  const id = req.params.id;
  Blog.findByIdAndDelete(id)
    .then(result => {
      res.json({ redirect: '/blogs' });
    })
    .catch(err => {
      console.log(err);
    });
}

module.exports = {
  blog_index,
  blog_details,
  blog_create_get,
  blog_create_post,
  blog_delete
}
```

blogRoutes.js

```
const express = require('express');
const blogController = require('../controllers/blogController');

const router = express.Router();

// blog routes
router.get('/', blogController.blog_index);
router.post('/', blogController.blog_create_post);
router.get('/create', blogController.blog_create_get);
router.get('/:id', blogController.blog_details);
router.delete('/:id', blogController.blog_delete);

module.exports = router;
```

Lesson 12

WrapUp

blogController.js

```
// blog_index, blog_details, blog_create_get, blog_create_post,
blog_delete

const Blog = require('../models/blog');

const blog_index = (req, res) => {
  Blog.find().sort({ createdAt: -1 })
    .then(result => {
      res.render('blogs/index', { blogs: result, title: 'All blogs' });
    })
    .catch(err => {
      console.log(err);
    });
}

const blog_details = (req, res) => {
  const id = req.params.id;
  Blog.findById(id)
    .then(result => {
      res.render('blogs/details', { blog: result, title: 'Blog Details'
});
    })
    .catch(err => {
      res.status(404).render('404', { title: 'Blog not found' });
    });
}

const blog_create_get = (req, res) => {
  res.render('blogs/create', { title: 'Create a new blog' });
}

const blog_create_post = (req, res) => {
  const blog = new Blog(req.body);
  blog.save()
```

```
    .then(result => {
      res.redirect('/blogs');
    })
    .catch(err => {
      console.log(err);
    });
}

const blog_delete = (req, res) => {
  const id = req.params.id;
  Blog.findByIdAndDelete(id)
    .then(result => {
      res.json({ redirect: '/blogs' });
    })
    .catch(err => {
      console.log(err);
    });
}

module.exports = {
  blog_index,
  blog_details,
  blog_create_get,
  blog_create_post,
  blog_delete
}
```

details.js

```
<html lang="en">
<%- include("../partials/head.ejs") %>

<body>
  <%- include("../partials/nav.ejs") %>

  <div class="details content">
    <h2><%= blog.title %></h2>
    <div class="content">
      <p><%= blog.body %></p>
    </div>

    <a class="delete" data-doc="<%= blog._id %>">
      <img src="/trashcan.svg" alt="delete icon">
    </a>
  </div>

  <%- include("../partials/footer.ejs") %>

  <script>
    const trashcan = document.querySelector('a.delete');

    trashcan.addEventListener('click', (e) => {
      const endpoint = `/blogs/${trashcan.dataset.doc}`;

      fetch(endpoint, {
        method: 'DELETE',
      })
      .then((response) => response.json())
      .then((data) => window.location.href = data.redirect)
      .catch((err) => console.log(err));
    });

  </script>
</body>
</html>
```

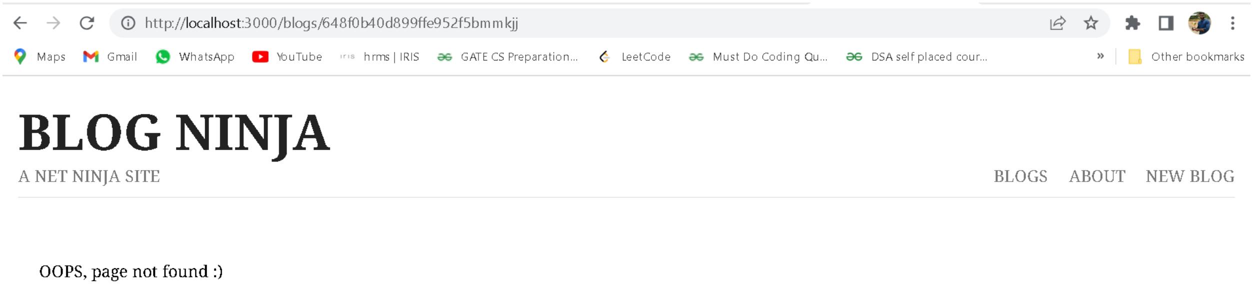
public → trashcan.svg

trashcan.svg

```
<svg xmlns="http://www.w3.org/2000/svg" height="24" viewBox="0 0 24 24"
width="24"><path d="M6 19c0 1.1.9 2 2 2h8c1.1 0 2-.9 2-2V7H6v12zM19
4h-3.5l-1-1h-5l-1 1H5v2h14V4z"/><path d="M0 0h24v24H0z"
fill="none"/></svg>
```

O/P:

→If the **“id”** is changed then page not found



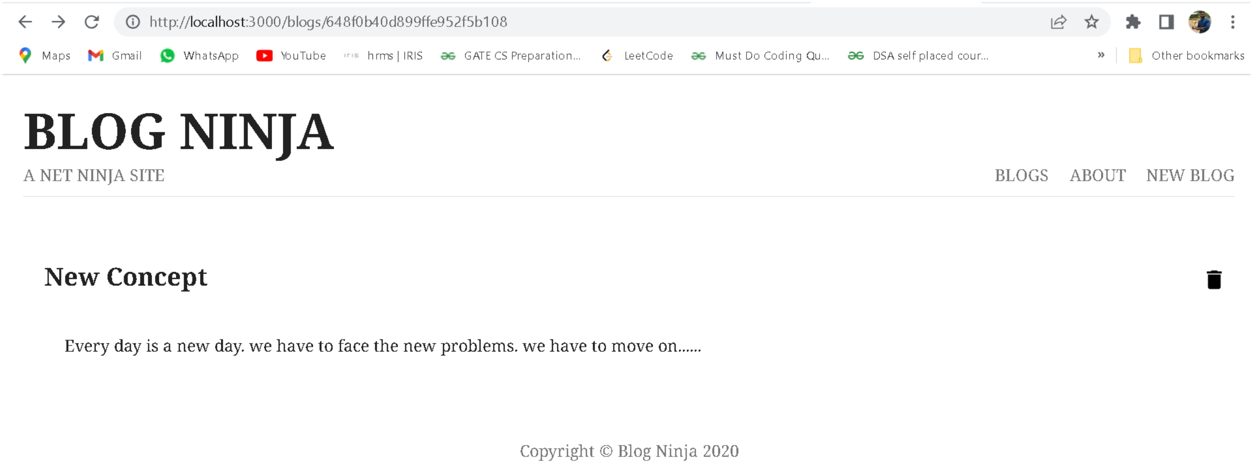
BLOG NINJA

A NET NINJA SITE

BLOGS ABOUT NEW BLOG

OOPS, page not found :)

→Delete icon is added



BLOG NINJA

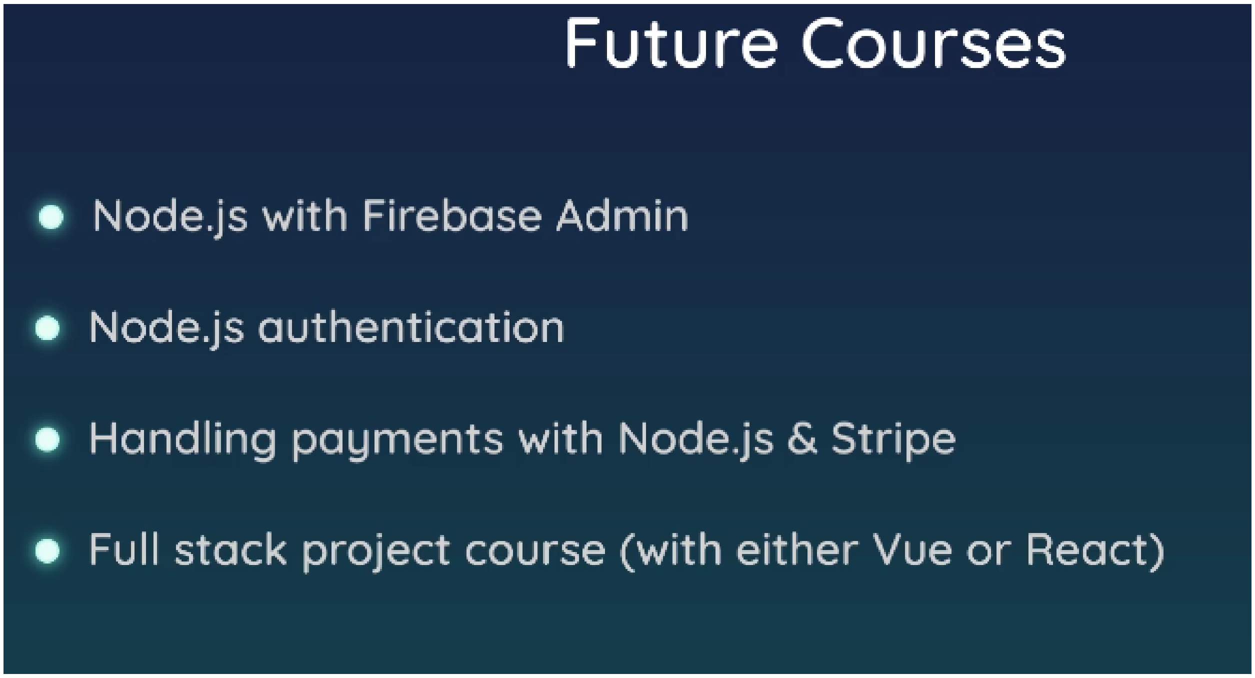
A NET NINJA SITE

BLOGS ABOUT NEW BLOG

New Concept

Every day is a new day. we have to face the new problems. we have to move on......

Copyright © Blog Ninja 2020



Future Courses

- Node.js with Firebase Admin
- Node.js authentication
- Handling payments with Node.js & Stripe
- Full stack project course (with either Vue or React)