

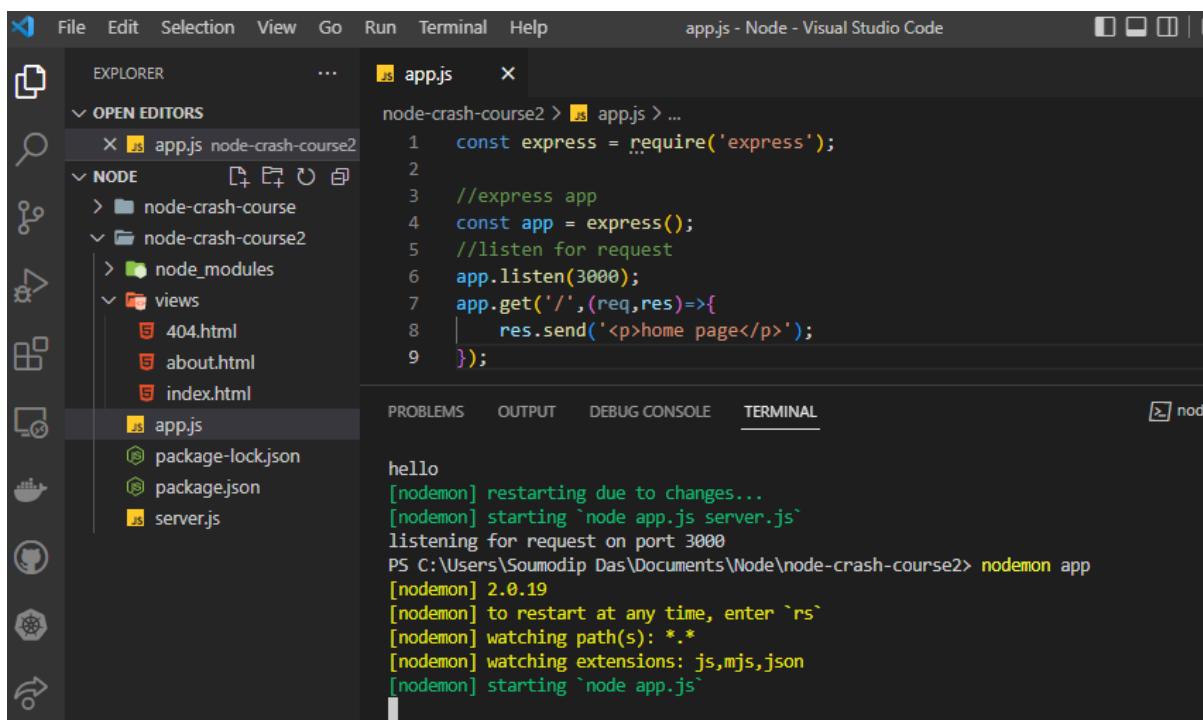
# Express

Express is a framework that helps us to easily manage our routing requests server-side logic and responses in a much more elegant way it makes our code easier to read and update and extend.

Express is really good it saves us time allows us to write clean code and it comes with some extra features that really help us out.

`npm install express` – for installing express.

## Creating an Express App

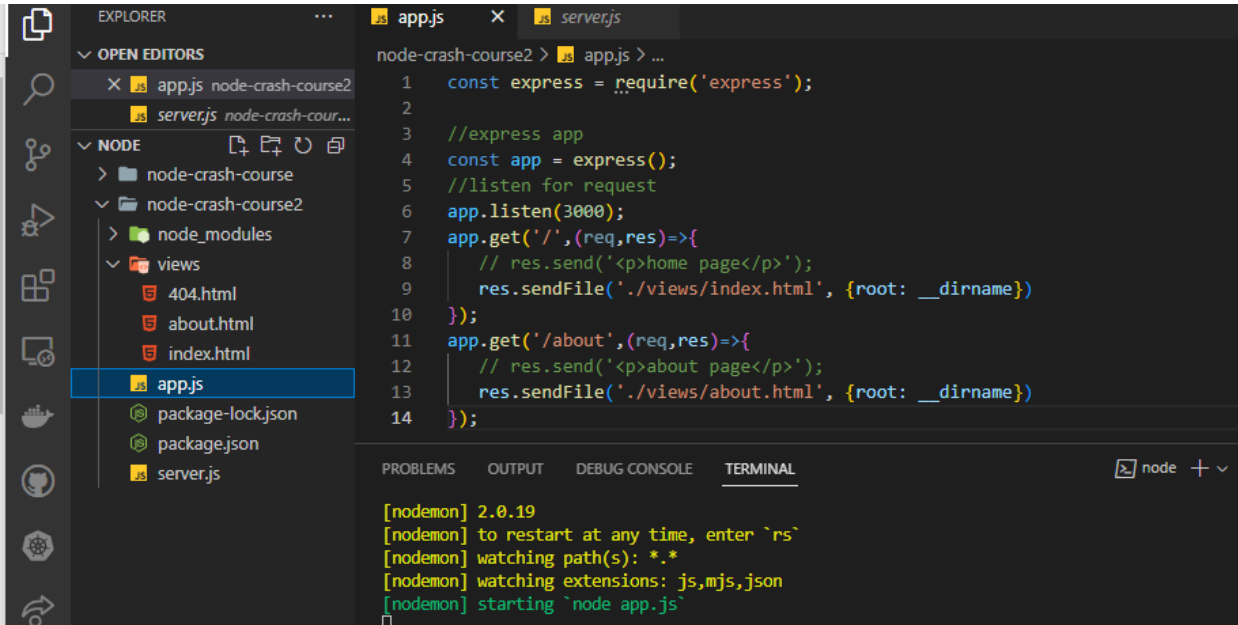


The screenshot displays the Visual Studio Code interface with a project named 'node-crash-course2'. The Explorer sidebar on the left shows the file structure, including a 'views' directory with HTML files and a main 'app.js' file. The main editor window shows the content of 'app.js', which is a simple Express.js server setup. The terminal at the bottom shows the command 'nodemon app' being executed, and the output indicates that the server is running on port 3000 and listening for requests.

```
node-crash-course2 > app.js > ...
1  const express = require('express');
2
3  //express app
4  const app = express();
5  //listen for request
6  app.listen(3000);
7  app.get('/',(req,res)=>{
8    res.send('<p>home page</p>');
9  });
```

```
hello
[nodemon] restarting due to changes...
[nodemon] starting `node app.js server.js`
listening for request on port 3000
PS C:\Users\Soumodip Das\Documents\Node\Node-crash-course2> nodemon app
[nodemon] 2.0.19
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node app.js`
```

# Routing & HTML

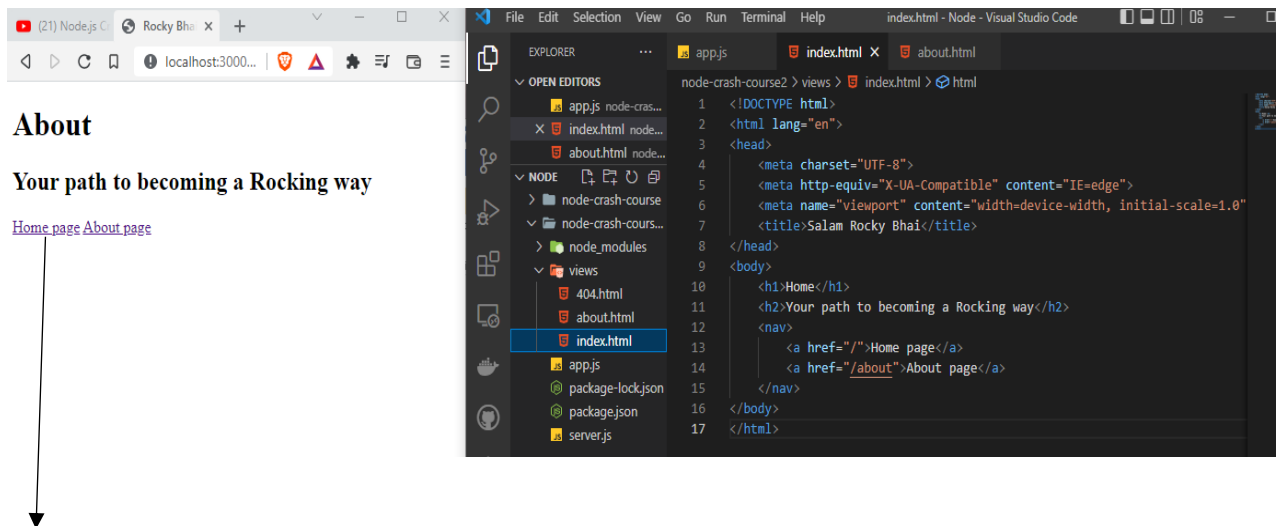


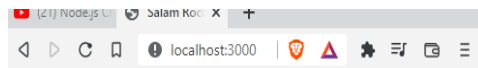
The screenshot shows the VS Code interface with the Explorer sidebar on the left, the Editor window in the center, and the Terminal at the bottom. The Explorer sidebar shows the file structure of the project, with the 'views' folder expanded and 'app.js' selected. The Editor window shows the code for 'app.js', which is a simple Express.js application. The code defines an Express app, listens on port 3000, and has two routes: a root route that sends 'home page' and an '/about' route that sends 'about page'. The Terminal shows the output of the application running with nodemon, indicating it is watching for changes and starting the application.

```
node-crash-course2 > app.js > ...
1  const express = require('express');
2
3  //express app
4  const app = express();
5  //listen for request
6  app.listen(3000);
7  app.get('/',(req,res)=>{
8    // res.send('<p>home page</p>');
9    res.sendFile('./views/index.html', {root: __dirname})
10 });
11 app.get('/about',(req,res)=>{
12   // res.send('<p>about page</p>');
13   res.sendFile('./views/about.html', {root: __dirname})
14 });
```

```
[nodemon] 2.0.19
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node app.js`
```

Linking a page with another page.





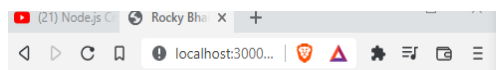
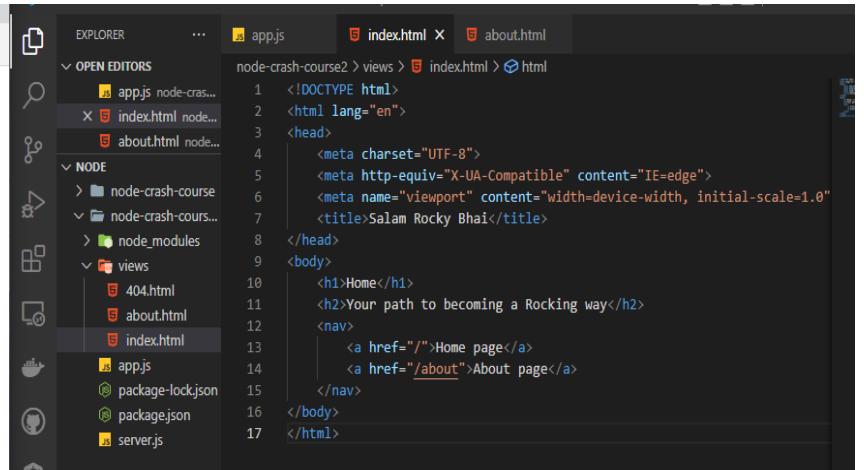
# Home

Your path to becoming a Rocking way

[Home page](#) [About page](#)



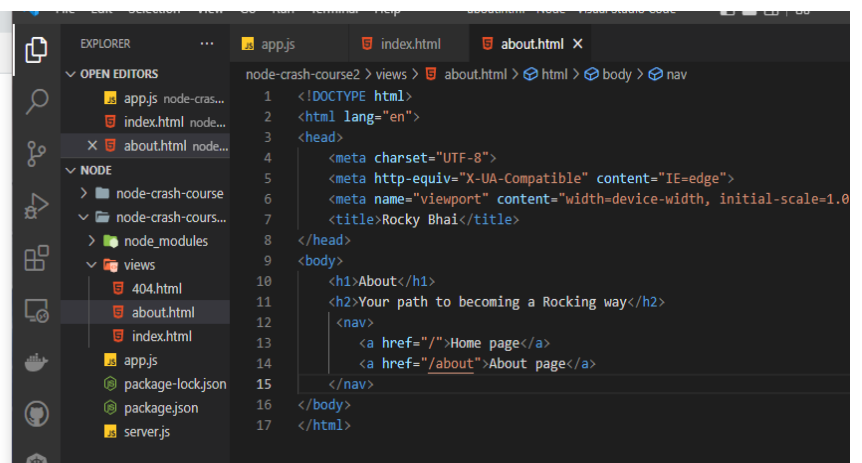
Click



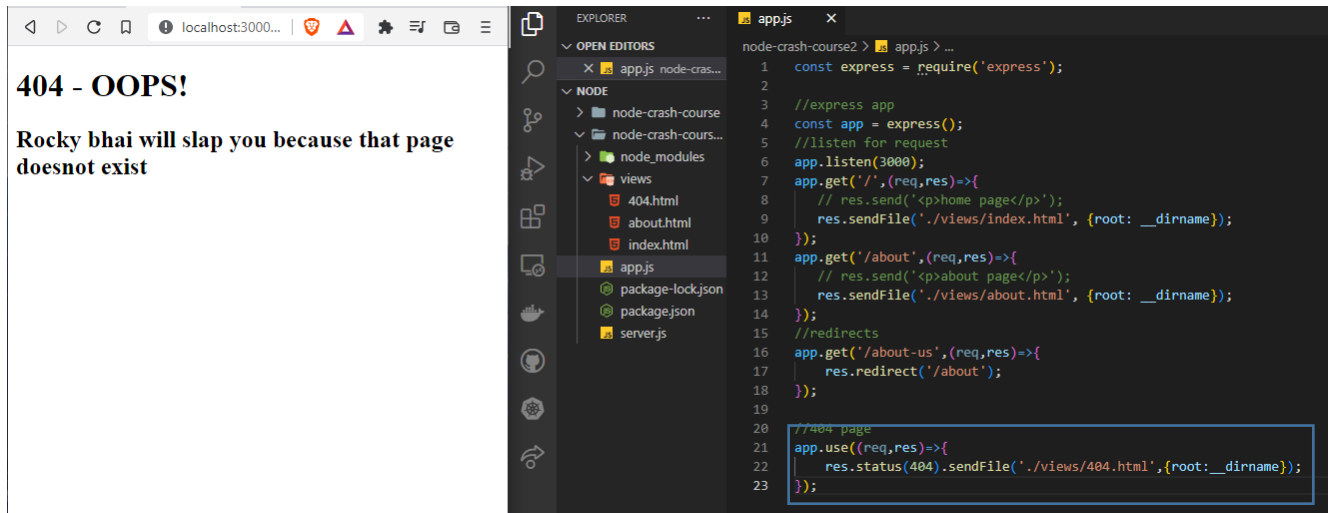
# About

Your path to becoming a Rocking way

[Home page](#) [About page](#)

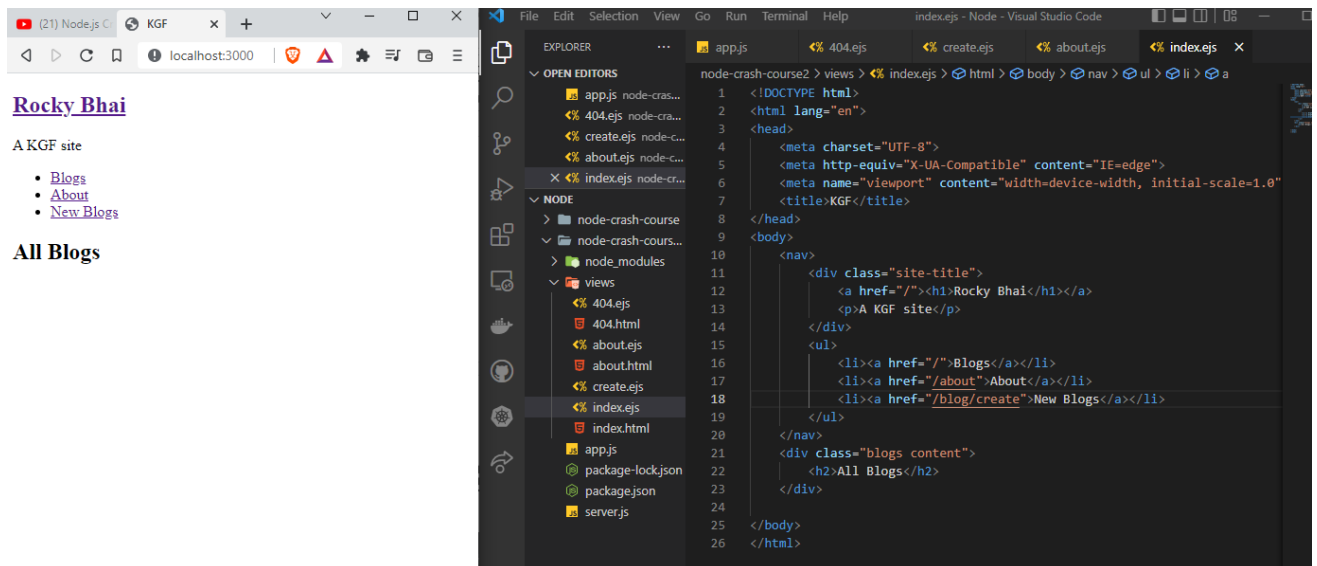


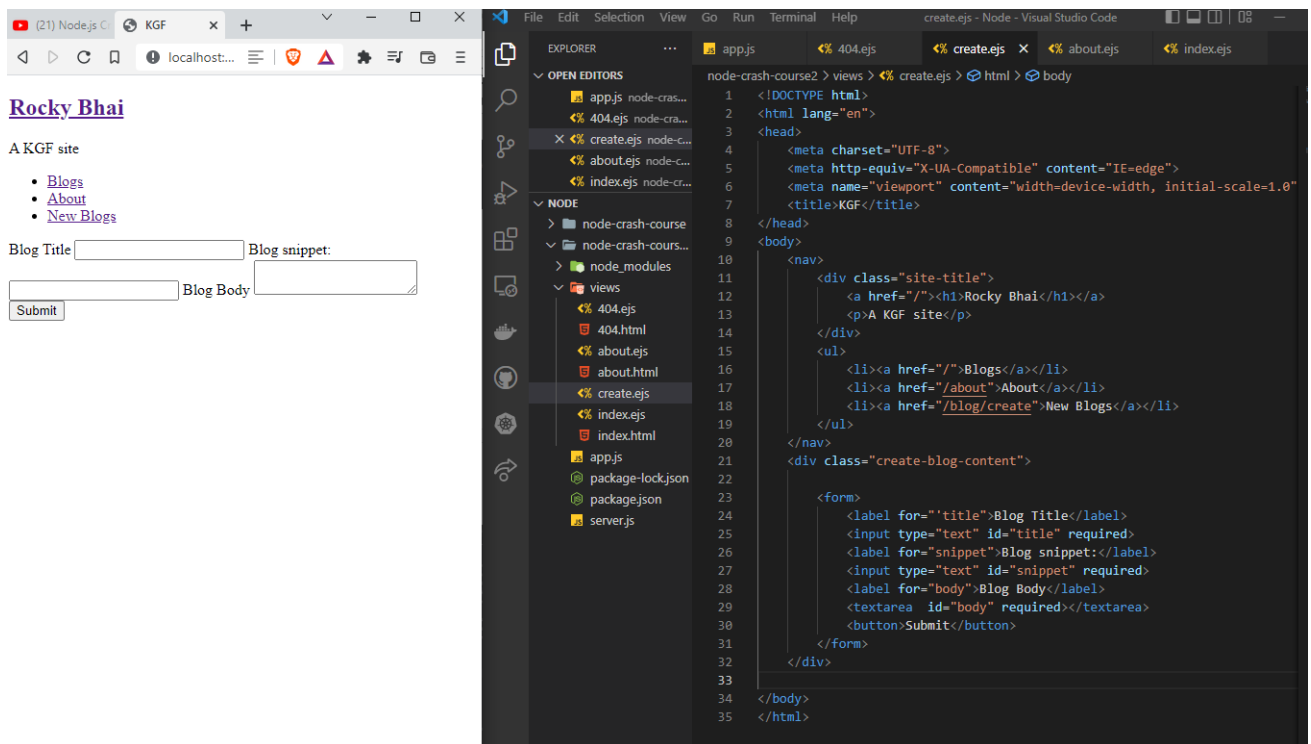
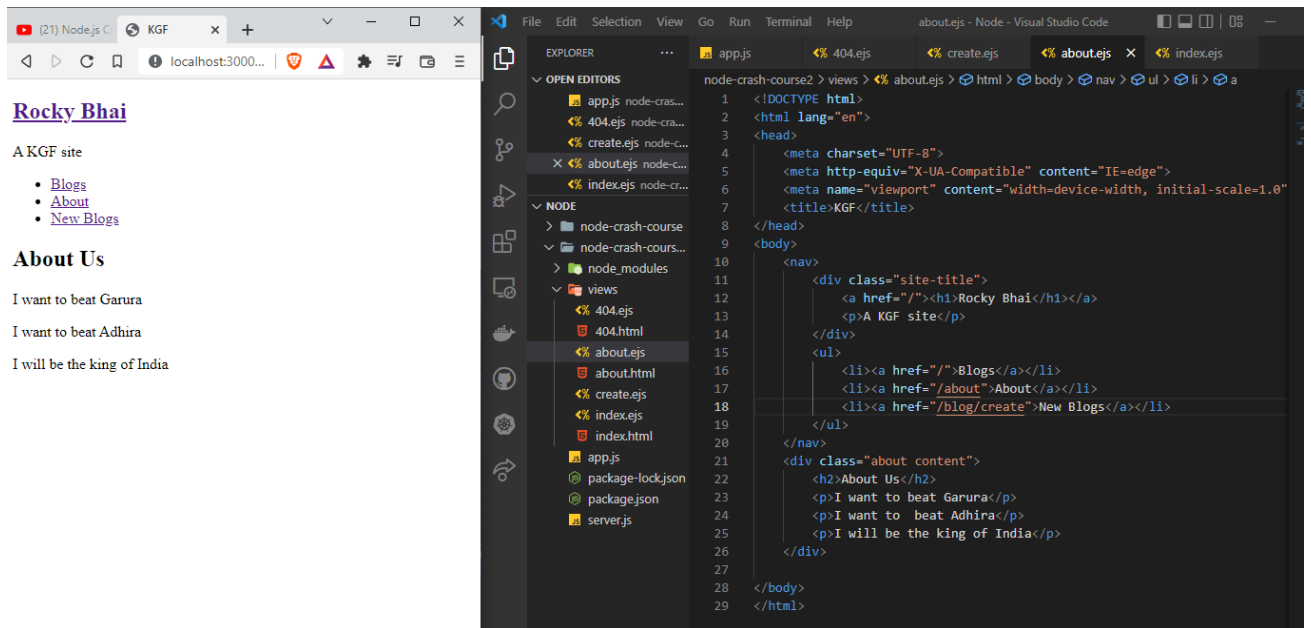
# Redirect & 404's



## EJS(Embedded Javascript Templating)

Installation: npm install ejs

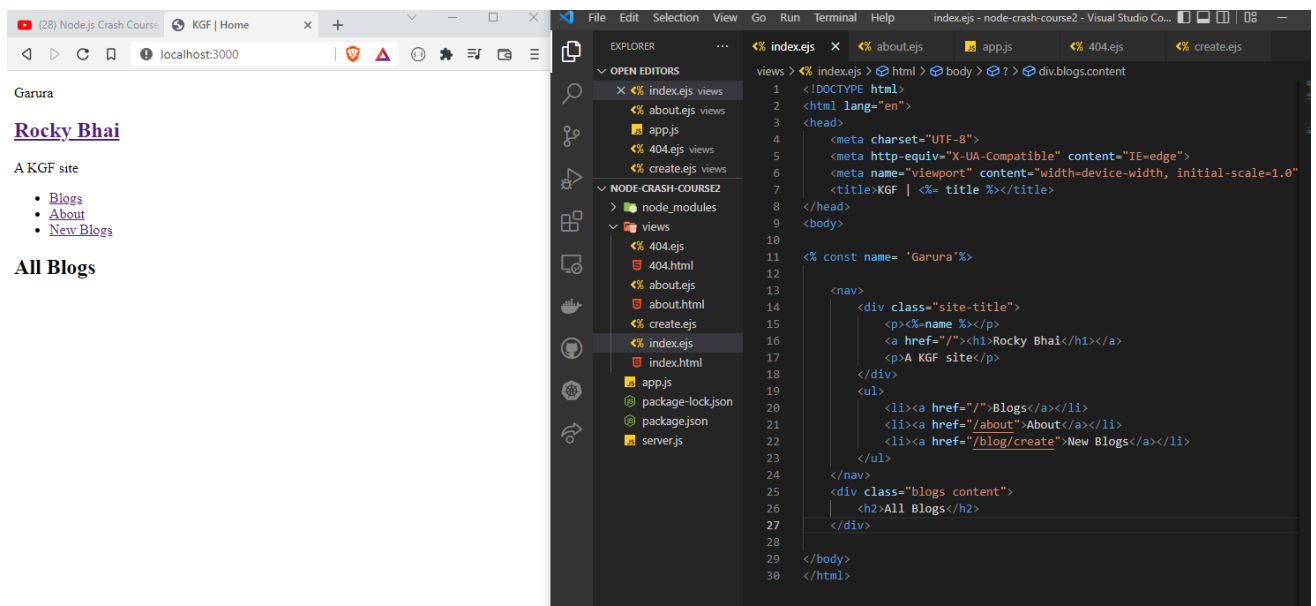
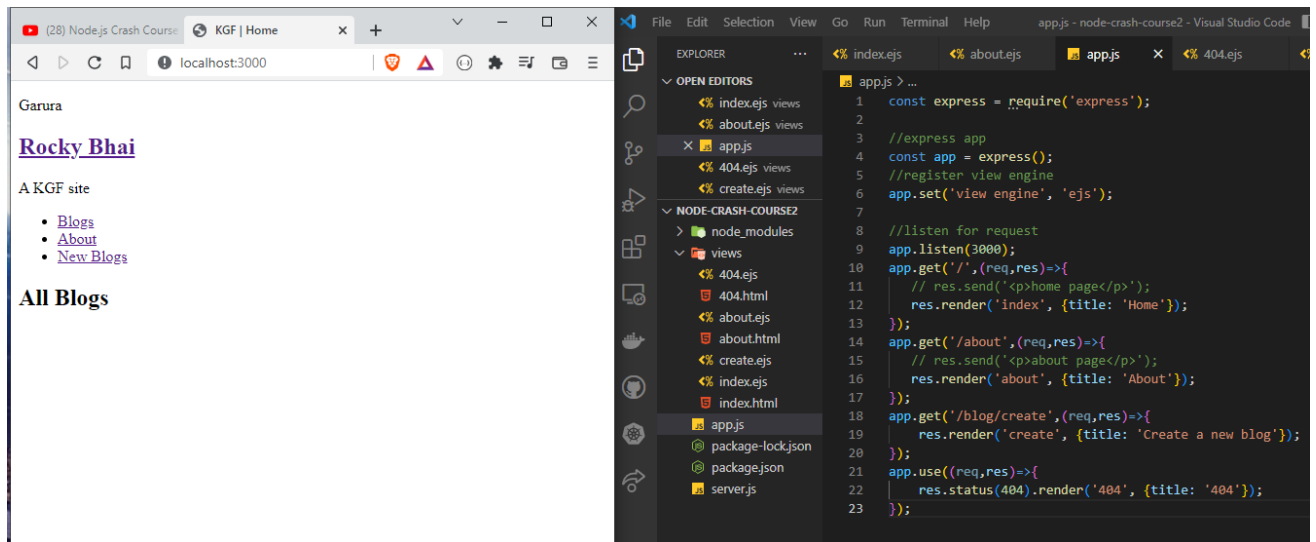


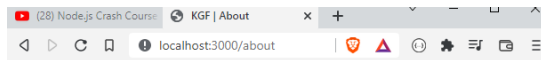


```
1 const express = require('express');
2
3 //express app
4 const app = express();
5 //register view engine
6 app.set('view engine', 'ejs');
7
8 //listen for request
9 app.listen(3000);
10 app.get('/', (req, res) => {
11   // res.send('<p>home page</p>');
12   res.render('index');
13 });
14 app.get('/about', (req, res) => {
15   // res.send('<p>about page</p>');
16   res.render('about');
17 });
18 app.get('/blog/create', (req, res) => {
19   res.render('create');
20 });
21 app.use((req, res) => {
22   res.status(404).render('404');
23 });
```

## Passing data into views

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>KGF</title>
8 </head>
9 <body>
10
11   <% const name= 'Garura'%>
12
13   <nav>
14     <div class="site-title">
15       <p><%=name %></p>
16       <a href="/"><h1>Rocky Bhai</h1></a>
17       <p>A KGF site</p>
18     </div>
19     <ul>
20       <li><a href="/">Blogs</a></li>
21       <li><a href="/about">About</a></li>
22       <li><a href="/blog/create">New Blogs</a></li>
23     </ul>
24   </nav>
25   <div class="blogs content">
26     <h2>All Blogs</h2>
27   </div>
28
29 </body>
30 </html>
```





## Rocky Bhai

A KGF site

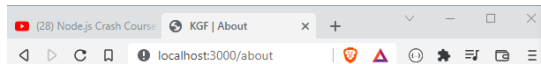
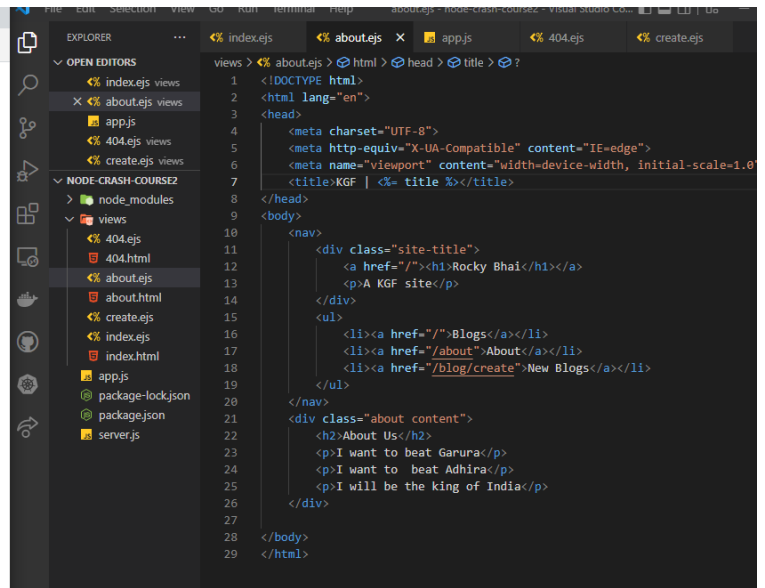
- [Blogs](#)
- [About](#)
- [New Blogs](#)

### About Us

I want to beat Garura

I want to beat Adhira

I will be the king of India



## Rocky Bhai

A KGF site

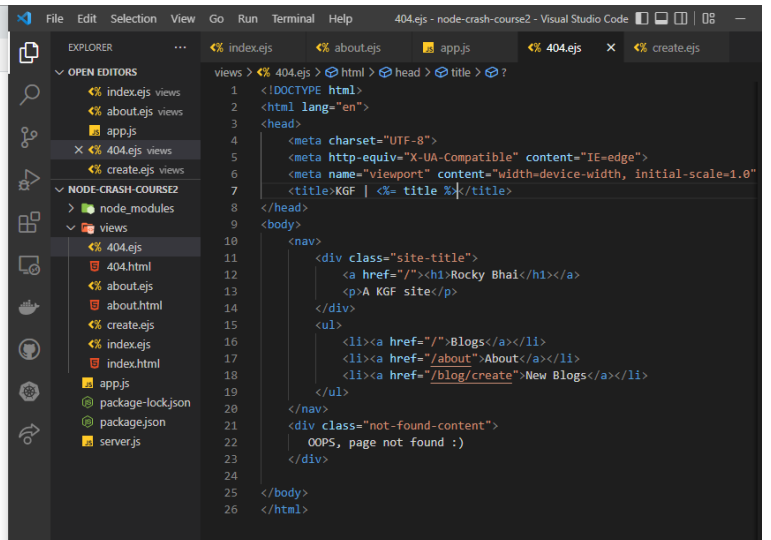
- [Blogs](#)
- [About](#)
- [New Blogs](#)

### About Us

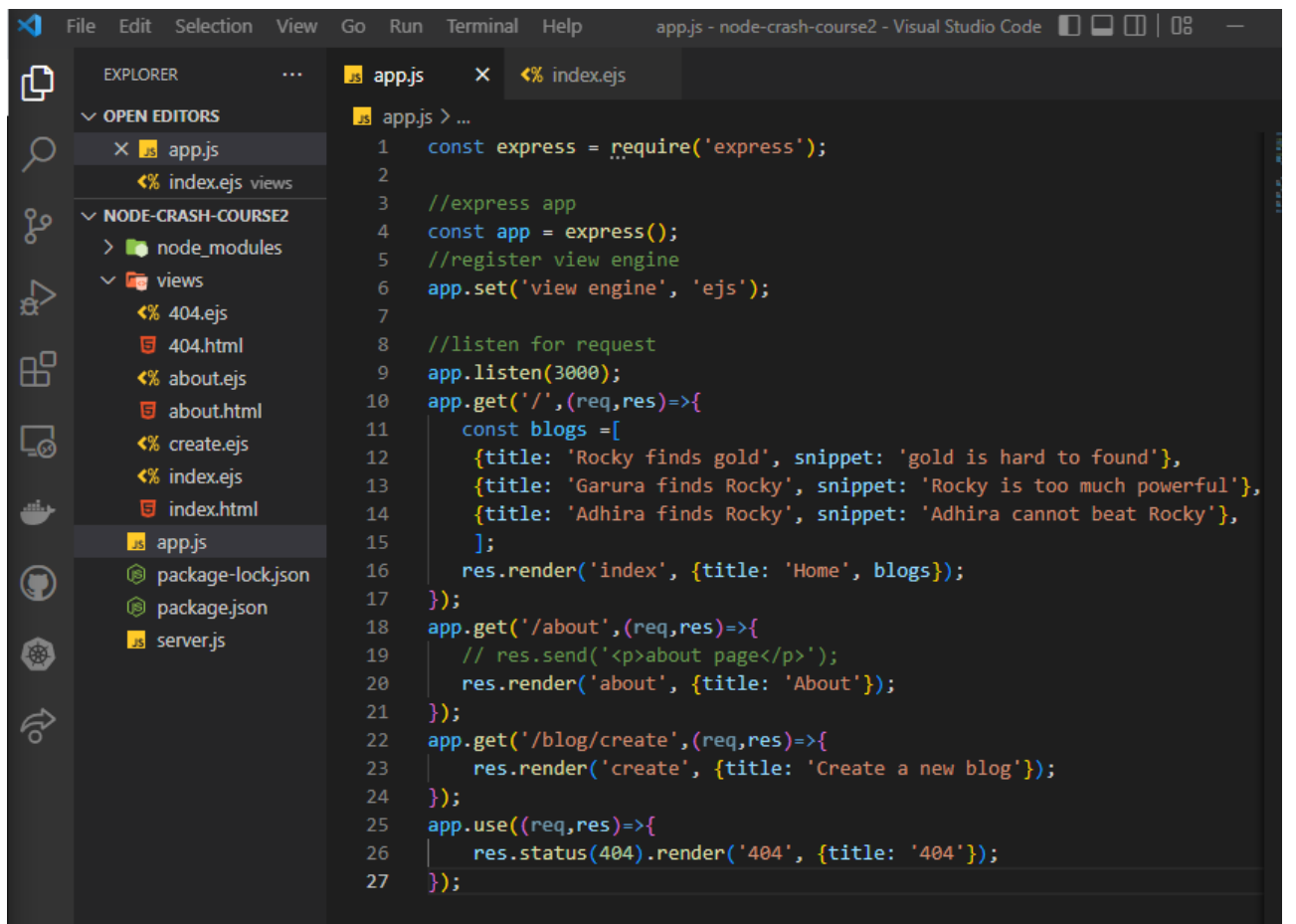
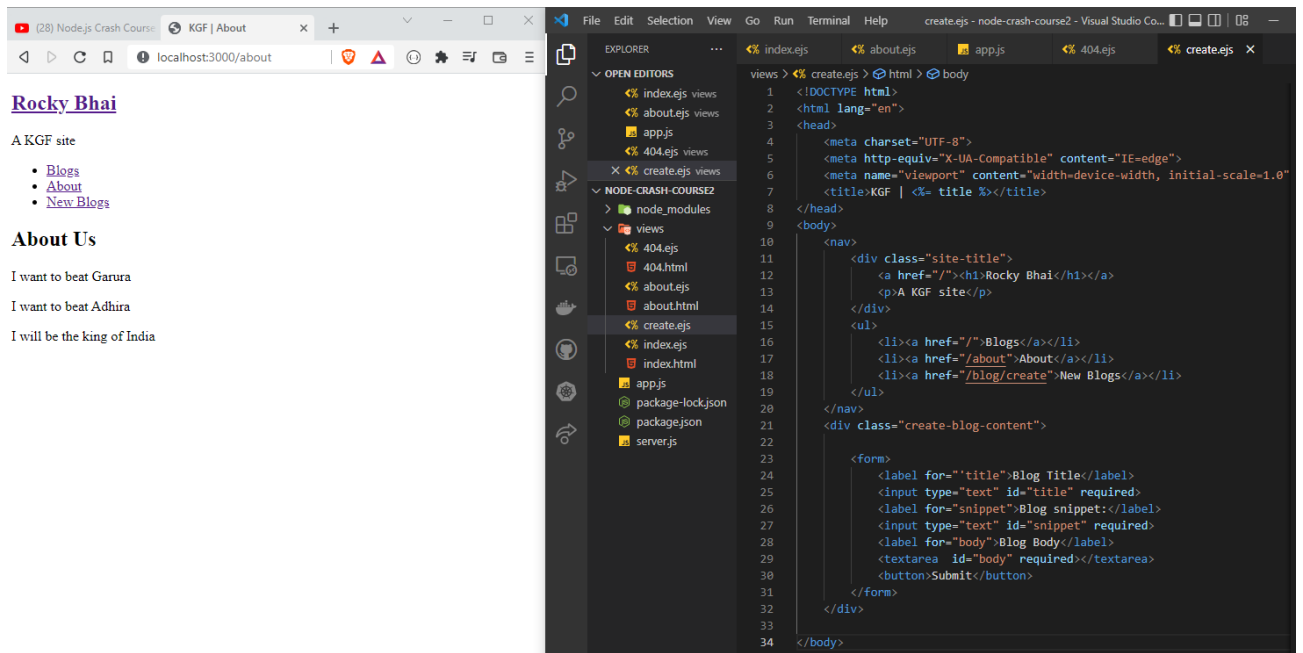
I want to beat Garura

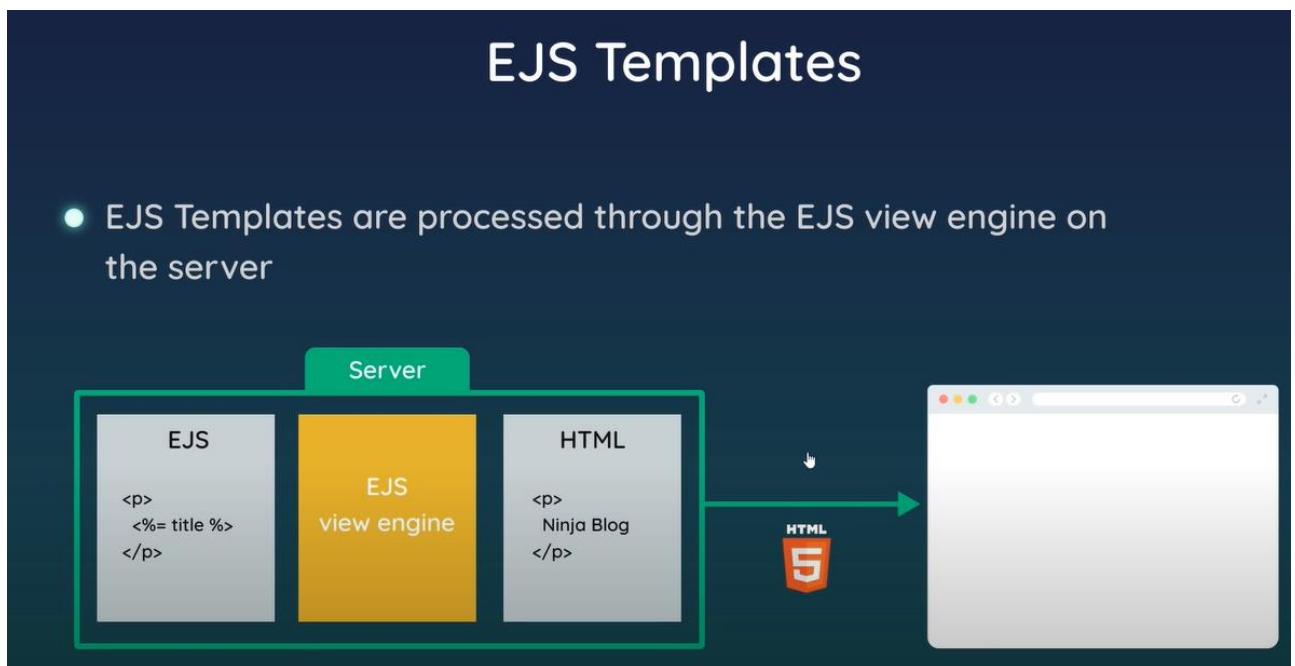
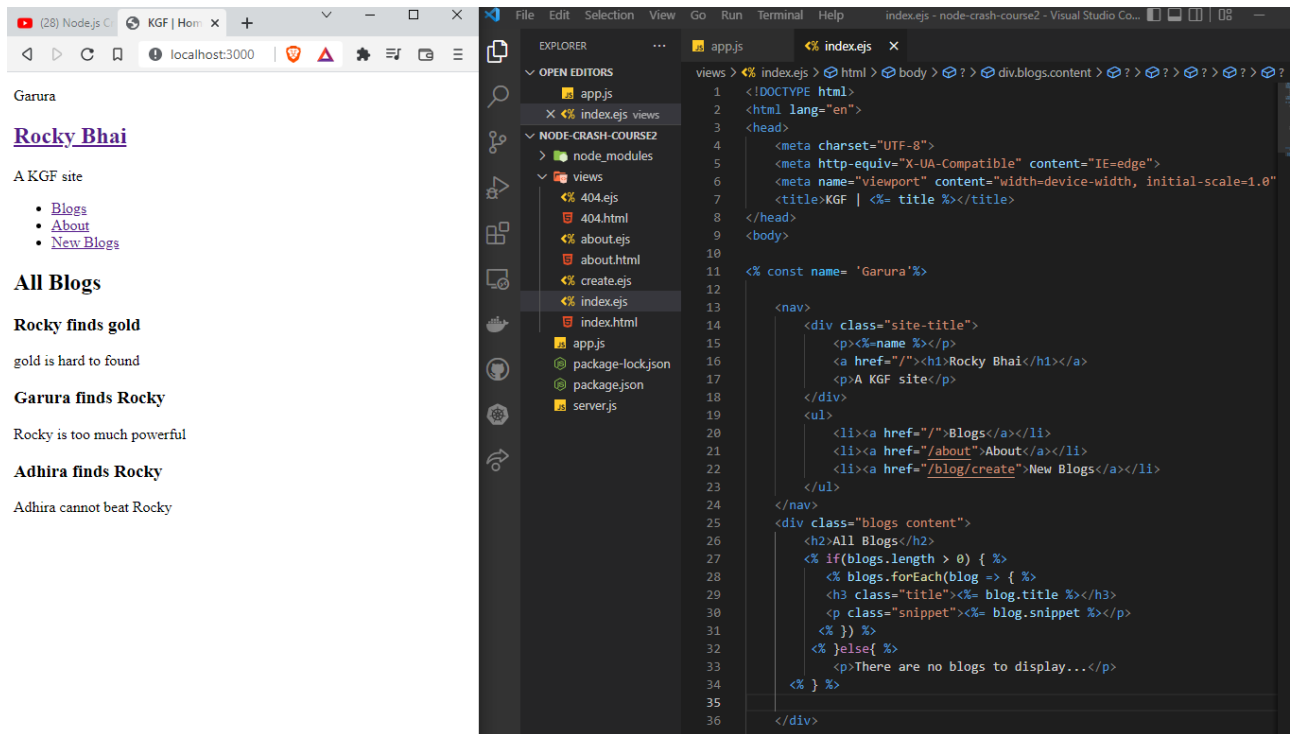
I want to beat Adhira

I will be the king of India

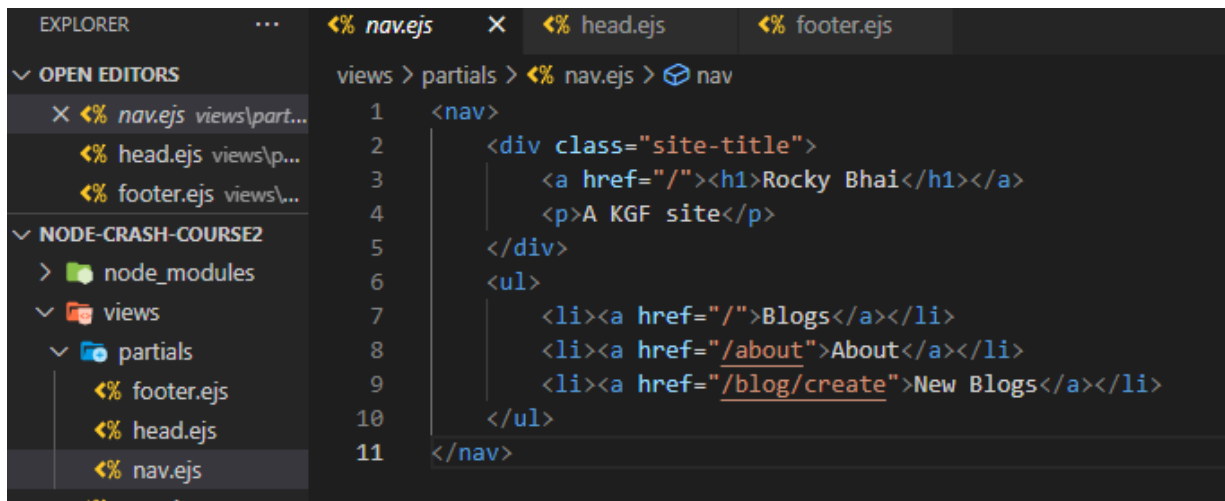




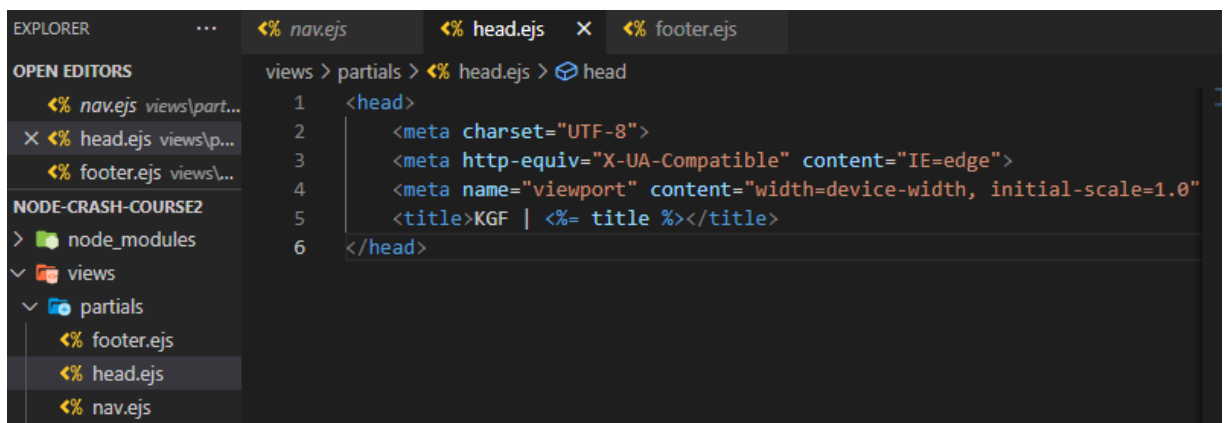




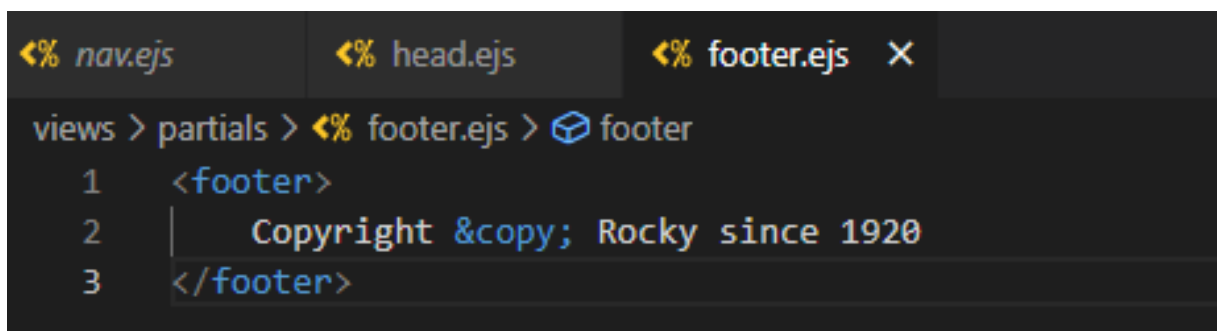
## Partials



```
1 <nav>
2   <div class="site-title">
3     <a href="/"><h1>Rocky Bhai</h1></a>
4     <p>A KGF site</p>
5   </div>
6   <ul>
7     <li><a href="/">Blogs</a></li>
8     <li><a href="/about">About</a></li>
9     <li><a href="/blog/create">New Blogs</a></li>
10  </ul>
11 </nav>
```



```
1 <head>
2   <meta charset="UTF-8">
3   <meta http-equiv="X-UA-Compatible" content="IE=edge">
4   <meta name="viewport" content="width=device-width, initial-scale=1.0">
5   <title>KGF | <%= title %></title>
6 </head>
```



```
1 <footer>
2   Copyright &copy; Rocky since 1920
3 </footer>
```

- For including those in ejs we need to write some syntaxes

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

# Adding CSS

```
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>KGF | <%= title %></title>
  <style>
    @import
url('https://fonts.googleapis.com/css2?family=Lobster&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-content form{
      max-width: 400px;
      margin: 0 auto;
    }
    .create-blog-content input,
    .create-blog-content textarea{
      display: block;
      width: 100%;
      margin: 10px 0;
      padding: 8px;
    }
    .create-blog-content label{
      display: block;
      margin-top: 24px;
    }
    textarea{
      height: 120px;
    }
    .create-blog-content button{
      margin-top: 20px;
      background: crimson;
      color: white;
      padding: 6px;
      border: 0;
      font-size: 1.2em;
      cursor: pointer;
    }
  </style>
</head>

```

//ON head.ejs