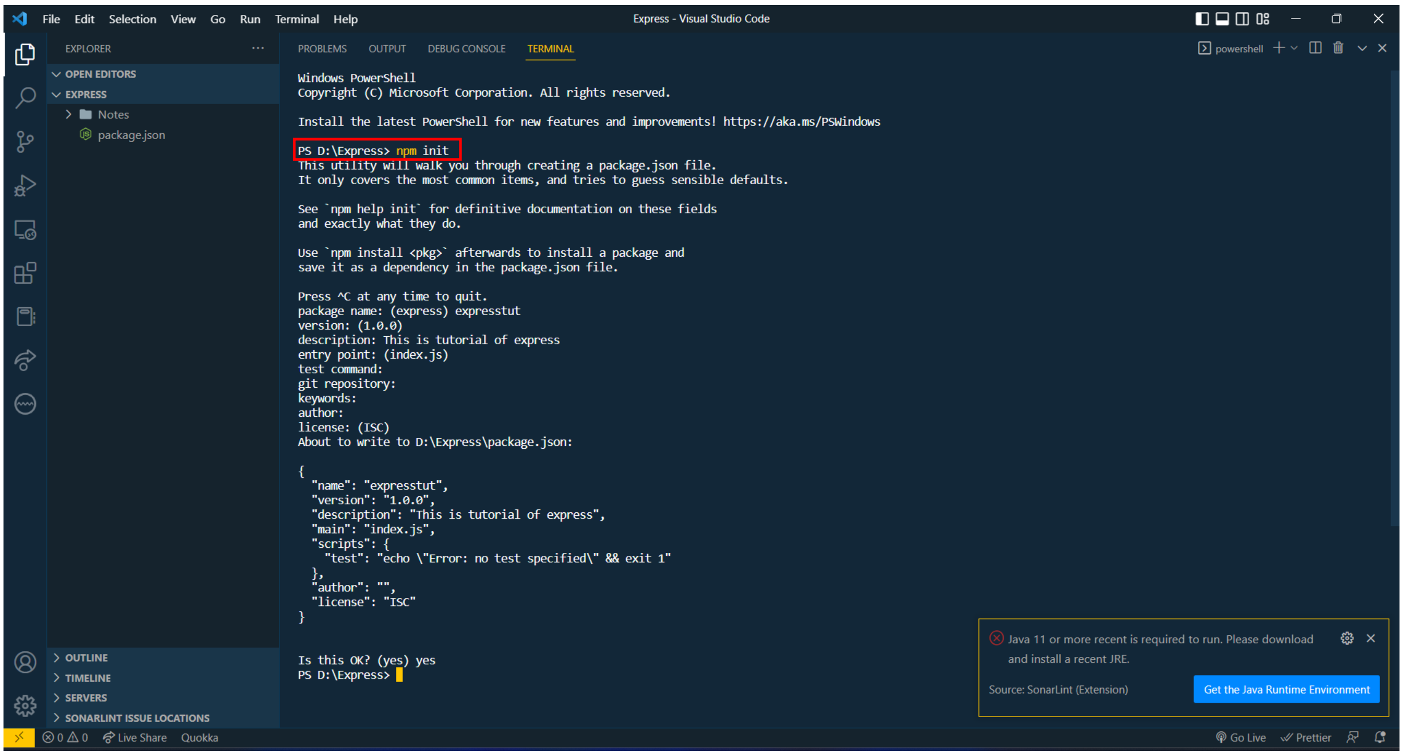
Express Js

- Is server side framework.
- If you want to use Node Js on server side then you will use Express Js.
- We can create our API.
- Unopinionated (You can write code on own opnion)

- Open vs code in particular folder.
- Open terminal and type: npm init

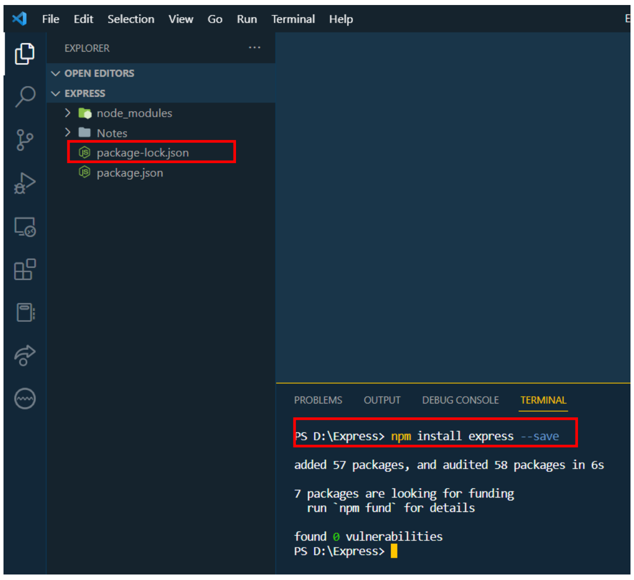

File Edit Selection View Go Run Terminal Help
Express - Visual Studio Code
EXPLORER
OPEN EDITORS
EXPRESS
Notes
package.json
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
OUTLINE
TIMELINE
SERVERS
SONARLINT ISSUE LOCATIONS
Java 11 or more recent is required to run. Please download and install a recent JRE.
Source: SonarLint (Extension)
Get the Java Runtime Environment
Live Share Quokka Go Live Prettier

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS D:\Express> npm init
This utility will walk you through creating a package.json file.
It only covers the most common items, and tries to guess sensible defaults.

See `npm help init` for definitive documentation on these fields
and exactly what they do.

Use `npm install <pkg>` afterwards to install a package and
save it as a dependency in the package.json file.

Press ^C at any time to quit.
package name: (express) expresstut
version: (1.0.0)
description: This is tutorial of express
entry point: (index.js)
test command:
git repository:
keywords:
author:
license: (ISC)
About to write to D:\Express\package.json:

{
  "name": "expresstut",
  "version": "1.0.0",
  "description": "This is tutorial of express",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "",
  "license": "ISC"
}


Is this OK? (yes) yes
PS D:\Express>
```

- Create index.js (because it is entry point: we define in above image.)
- We write code in index.js

Node js: browser per je javaScript run thay che ene server per run karva mate no ek tariko che.

- Run this command in terminal : `$ npm install express --save`

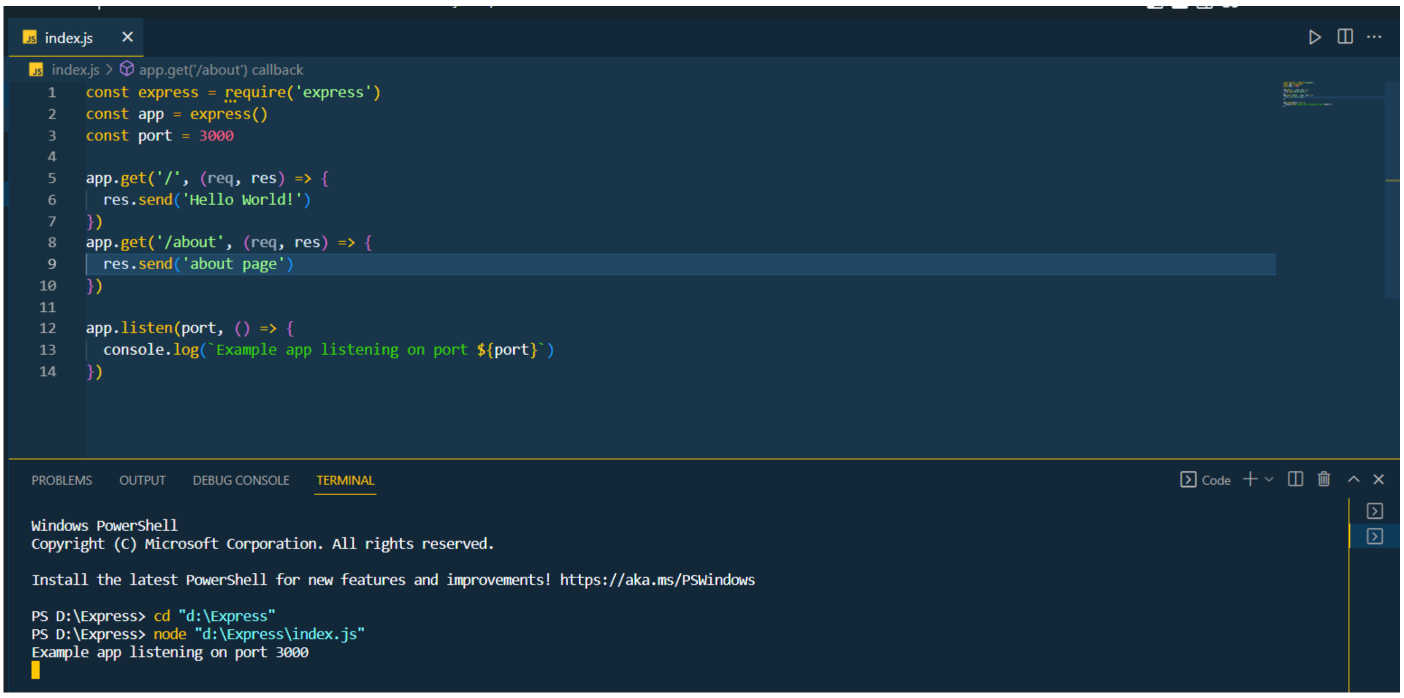

File Edit Selection View Go Run Terminal Help
EXPLORER
OPEN EDITORS
EXPRESS
node_modules
Notes
package-lock.json
package.json
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
PS D:\Express> npm install express --save

added 57 packages, and audited 58 packages in 6s

7 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
PS D:\Express>
```

npm → is a package manager of node. With help of we can download other packages in node_modules. Also we can use other package using npm.

node_modules: is a very big folder. When we run ***npm i*** or ***npm init*** that time this folder is created

Docs: of Hello world



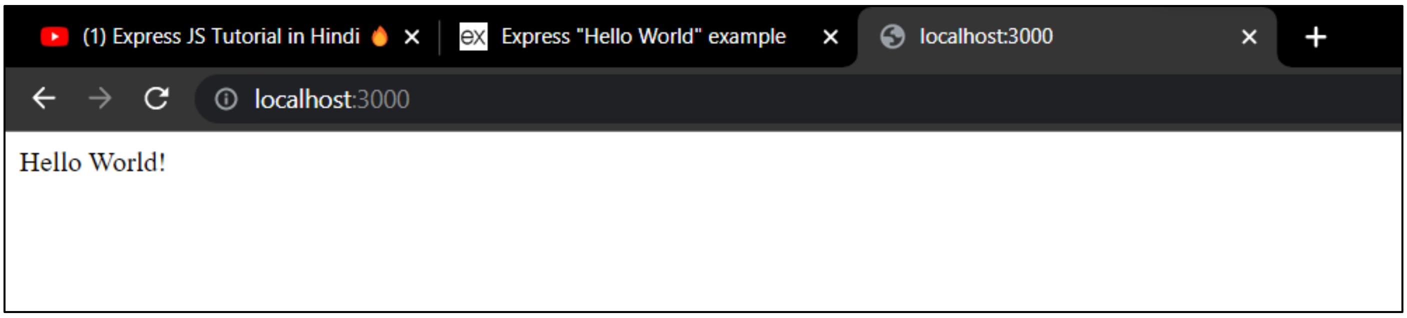
```
const express = require('express')
const app = express()
const port = 3000

app.get('/', (req, res) => {
  res.send('Hello World!')
})
app.get('/about', (req, res) => {
  res.send('about page')
})

app.listen(port, () => {
  console.log(`Example app listening on port ${port}`)
})
```

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS D:\Express> cd "d:\Express"
PS D:\Express> node "d:\Express\index.js"
Example app listening on port 3000
```

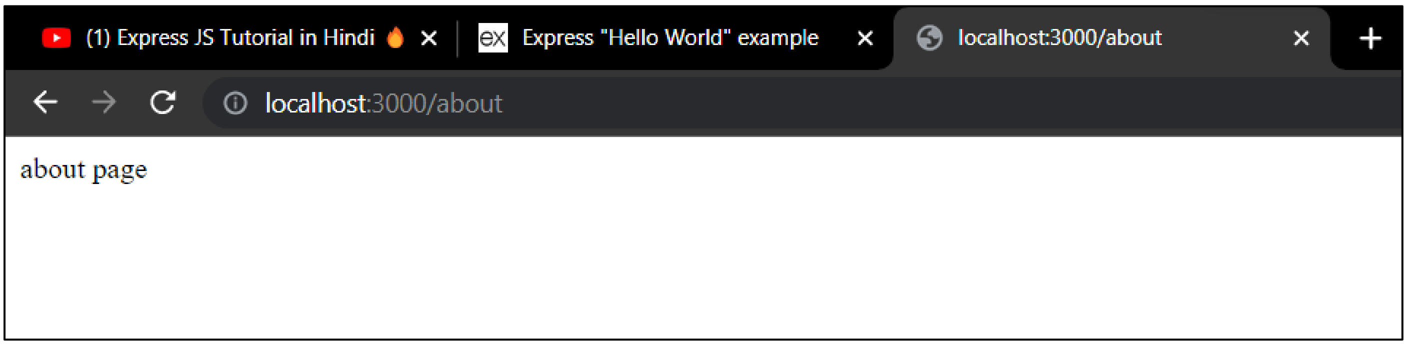
Open any browser and type:

localhost:3000 → This page show message "Hello World!"

here 3000 is our port number which is we define in code.
localhost:3000/about → This page show message "about page"


(1) Express JS Tutorial in Hindi
Express "Hello World" example
localhost:3000
localhost:3000

Hello World!


(1) Express JS Tutorial in Hindi
Express "Hello World" example
localhost:3000/about
localhost:3000/about

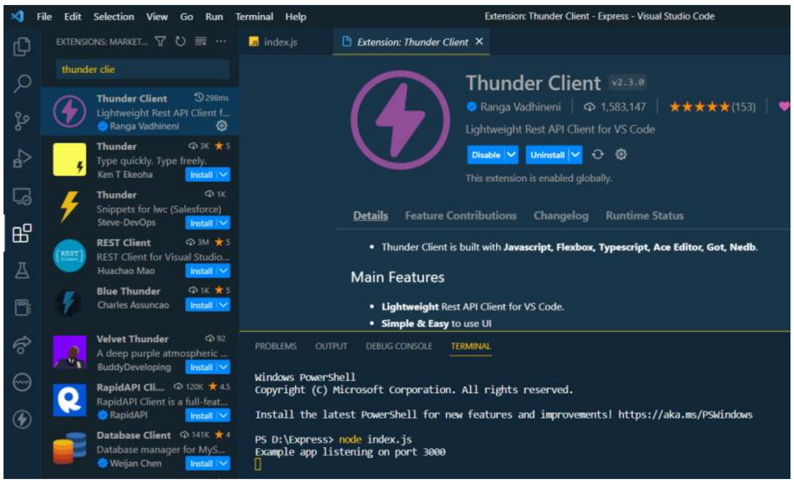
about page

Now, install extension thunder client.

It is use to run code inner vs code.

Which is use for pass get request. Like to run above program in vs code. Open thunder client and type:

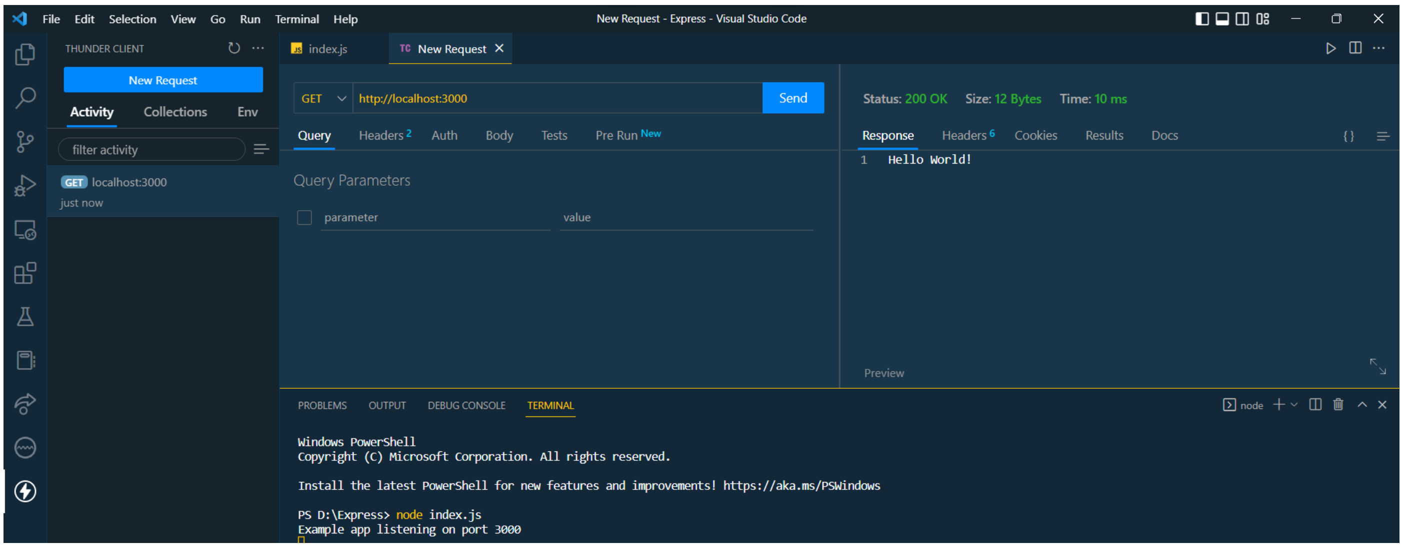
http://localhost:3000


Extension: Thunder Client - Express - Visual Studio Code
Thunder Client v2.3.0
Ranga Vadhineni
Lightweight Rest API Client for VS Code
Disable
Uninstall
This extension is enabled globally.
Details
Feature Contributions
Changelog
Runtime Status
Main Features

• Thunder Client is built with **Javascript, Flexbox, Typescript, Ace Editor, Got, Nedb.**

• **Lightweight** Rest API Client for VS Code.
• **Simple & Easy** to use UI

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS D:\Express> node index.js
Example app listening on port 3000
```

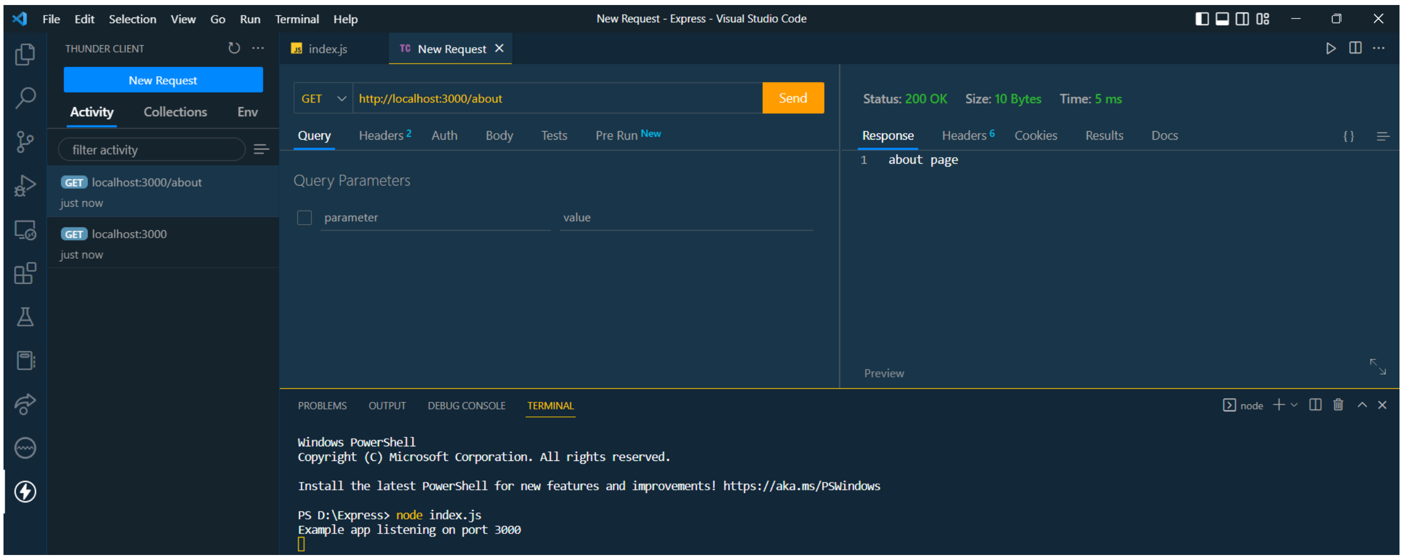


GET http://localhost:3000

Status: 200 OK Size: 12 Bytes Time: 10 ms

```
Hello World!
```

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS D:\Express> node index.js
Example app listening on port 3000
```



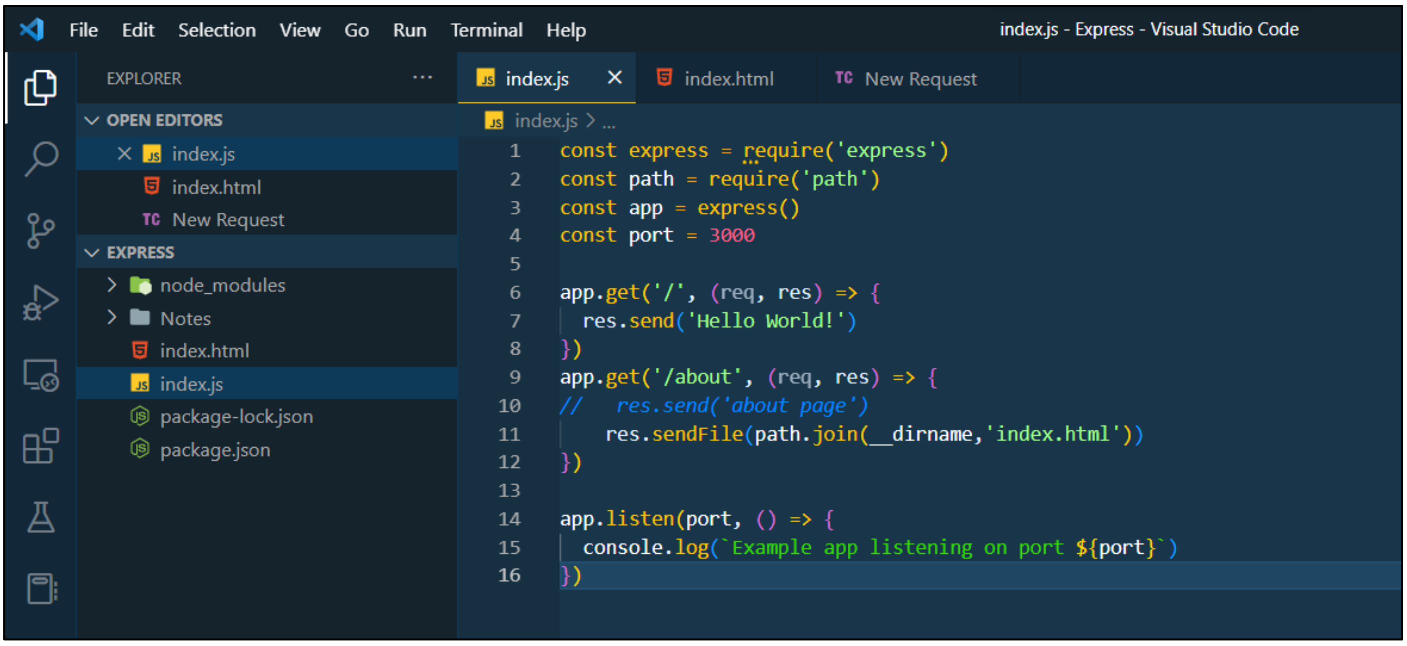
GET http://localhost:3000/about

Status: 200 OK Size: 10 Bytes Time: 5 ms

```
about page
```

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS D:\Express> node index.js
Example app listening on port 3000
```

We can also pass the file. Like inde.html , etc. use below code like that

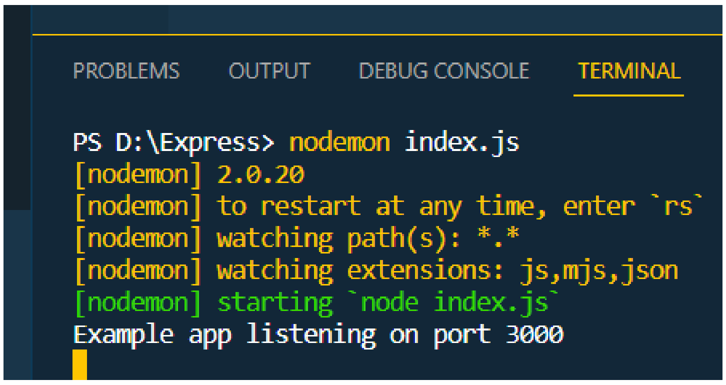


```
const express = require('express')
const path = require('path')
const app = express()
const port = 3000

app.get('/', (req, res) => {
  res.send('Hello World!')
})
app.get('/about', (req, res) => {
//   res.send('about page')
    res.sendFile(path.join(__dirname,'index.html'))
})

app.listen(port, () => {
  console.log(`Example app listening on port ${port}`)
})
```

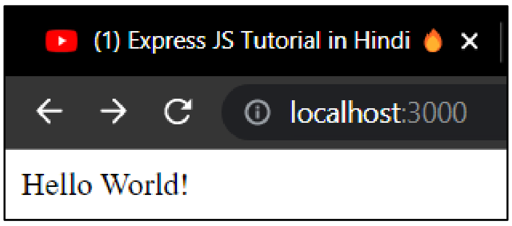
Open terminal: npm install -g nodemon

It is use to automatic execute code

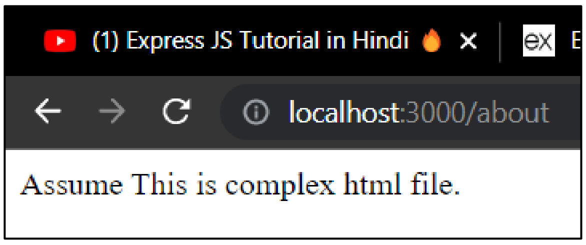
Now type: nodemon index.js



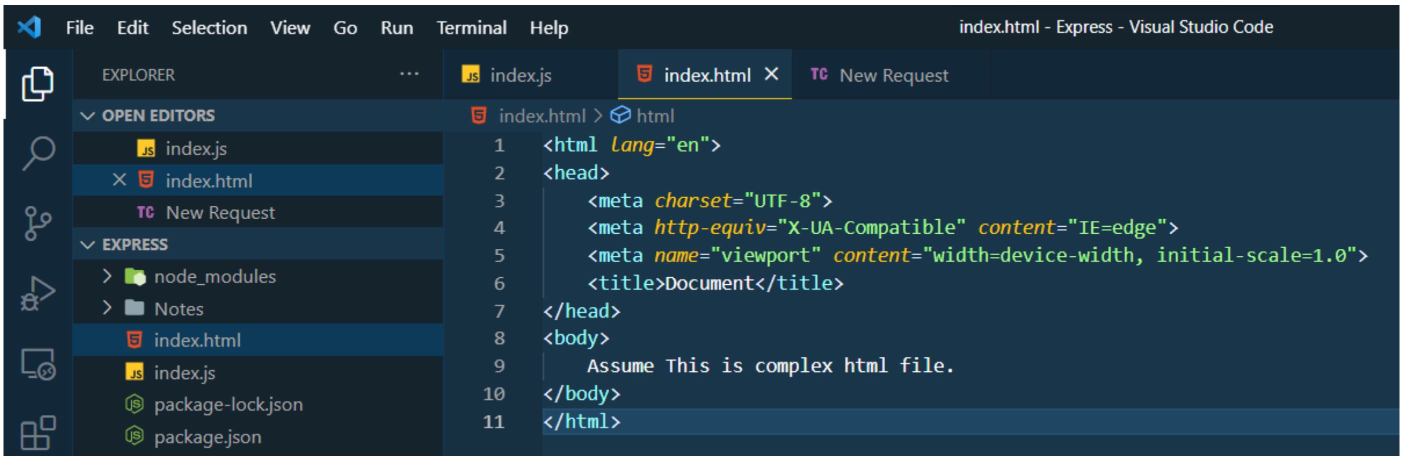
```
PS D:\Express> nodemon index.js
[nodemon] 2.0.20
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node index.js`
Example app listening on port 3000
```



Hello World!



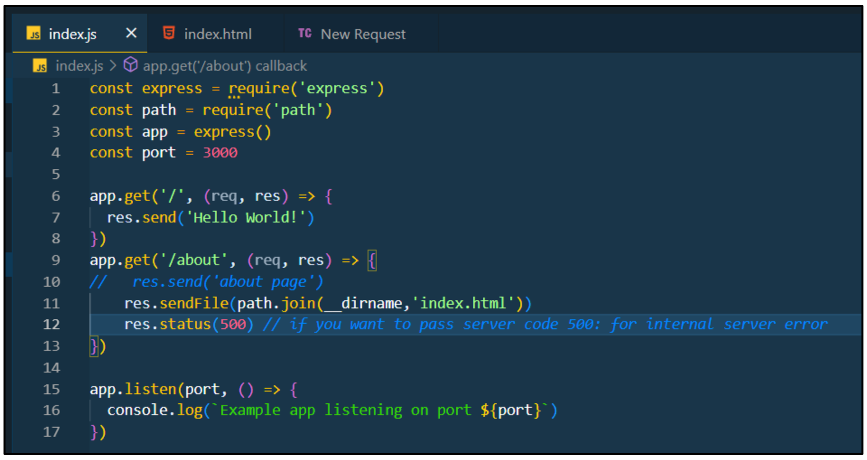
Assume This is complex html file.



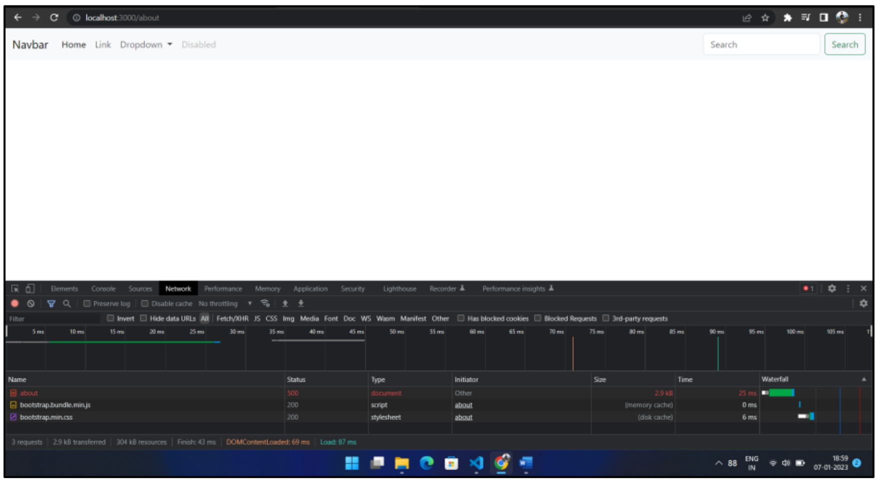
```
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    Assume This is complex html file.
</body>
</html>
```

We add bootstrap in index.html

With help of express we can also create static website.

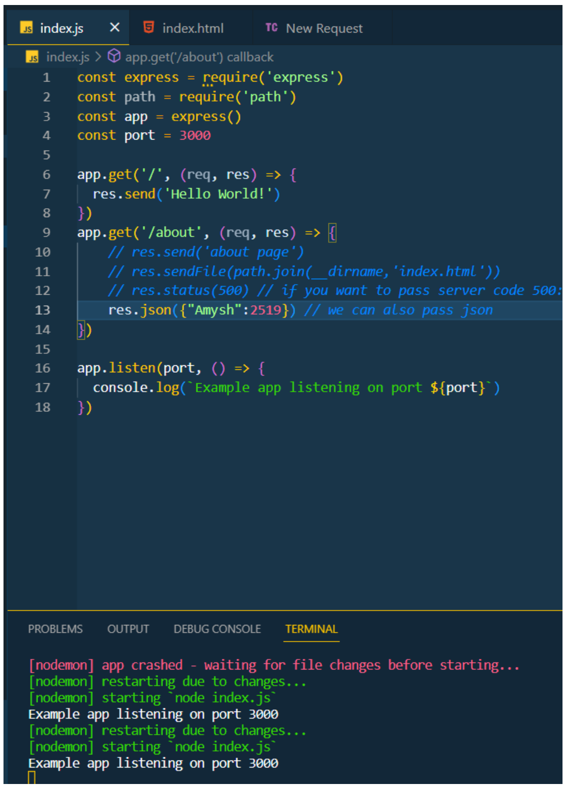


```
const express = require('express')
const path = require('path')
const app = express()
const port = 3000

app.get('/', (req, res) => {
  res.send('Hello World!')
})
app.get('/about', (req, res) => {
//    res.send('about page')
    res.sendFile(path.join(__dirname,'index.html'))
    res.status(500) // if you want to pass server code 500: for internal server error
})

app.listen(port, () => {
  console.log(`Example app listening on port ${port}`)
})
```



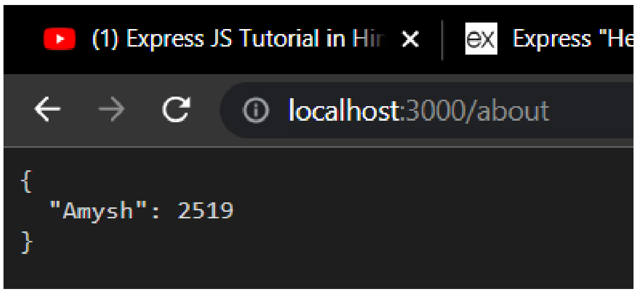
```
Name                    Status  Type        Initiator  Size            Time
about                   500     document    Other      2.9 kB          25 ms
bootstrap.bundle.min.js 200     script      about      (memory cache)  0 ms
bootstrap.min.css       200     stylesheet  about      (disk cache)    6 ms

3 requests | 2.9 kB transferred | 304 kB resources | Finish: 43 ms | DOMContentLoaded: 69 ms | Load: 87 ms
```

Pass json:



```
const express = require('express')
const path = require('path')
const app = express()
const port = 3000

app.get('/', (req, res) => {
  res.send('Hello World!')
})
app.get('/about', (req, res) => {
    // res.send('about page')
    // res.sendFile(path.join(__dirname,'index.html'))
    // res.status(500) // if you want to pass server code 500:
    res.json({"Amysh":2519}) // we can also pass json
})

app.listen(port, () => {
  console.log(`Example app listening on port ${port}`)
})
```

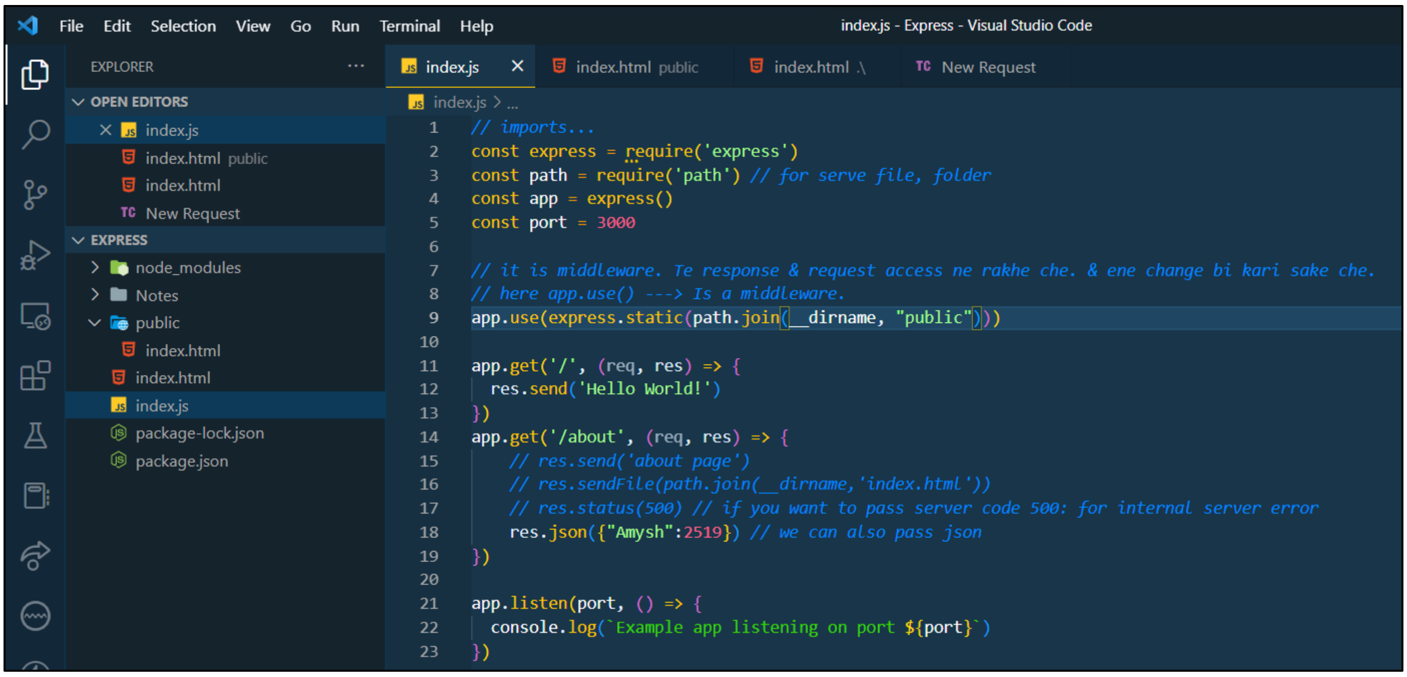
```
[nodemon] app crashed - waiting for file changes before starting...
[nodemon] restarting due to changes...
[nodemon] starting `node index.js`
Example app listening on port 3000
[nodemon] restarting due to changes...
[nodemon] starting `node index.js`
Example app listening on port 3000
```



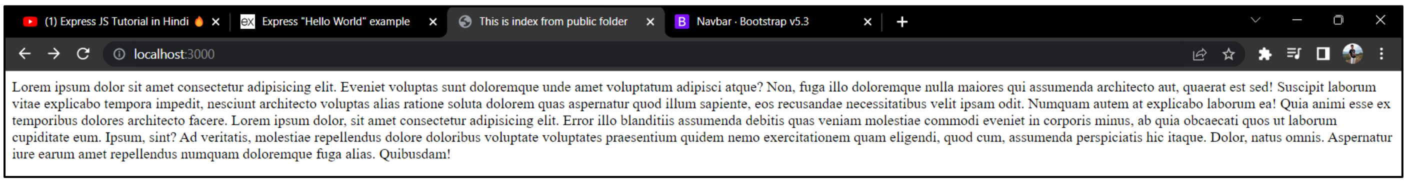
```
{
  "Amysh": 2519
}
```

Now we will serve static folder.

- We use middle ware: app.use()

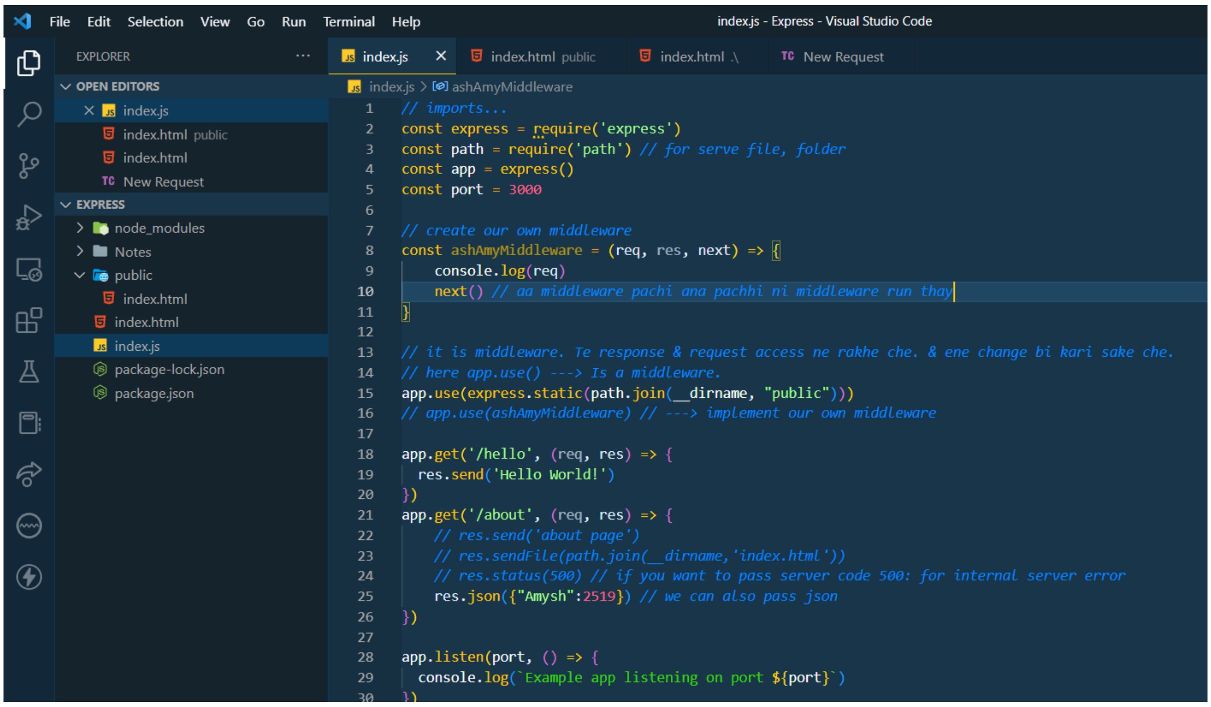


```
// imports...
const express = require('express')
const path = require('path') // for serve file, folder
const app = express()
const port = 3000

// it is middleware. Te response & request access ne rakhe che. & ene change bi kari sake che.
// here app.use() ---> Is a middleware.
app.use(express.static(path.join(__dirname, "public")))

app.get('/', (req, res) => {
  res.send('Hello World!')
})
app.get('/about', (req, res) => {
    // res.send('about page')
    // res.sendFile(path.join(__dirname,'index.html'))
    // res.status(500) // if you want to pass server code 500: for internal server error
    res.json({"Amysh":2519}) // we can also pass json
})

app.listen(port, () => {
  console.log(`Example app listening on port ${port}`)
})
```

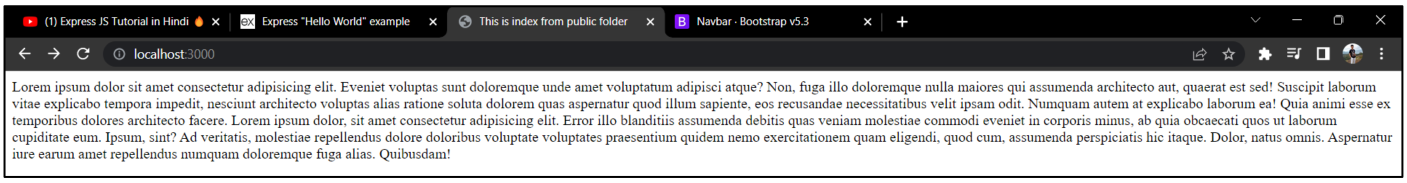


Lorem ipsum dolor sit amet consectetur adipisicing elit. Eveniet voluptas sunt doloremque unde amet voluptatum adipisci atque? Non, fuga illo doloremque nulla maiores qui assumenda architecto aut, quaerat est sed! Suscipit laborum vitae explicabo tempora impedit, nesciunt architecto voluptas alias ratione soluta dolorem quas aspernatur quod illum sapiente, eos recusandae necessitatibus velit ipsam odit. Numquam autem at explicabo laborum ea! Quia animi esse ex temporibus dolores architecto facere. Lorem ipsum dolor, sit amet consectetur adipisicing elit. Error illo blanditiis assumenda debitis quas veniam molestiae commodi eveniet in corporis minus, ab quia obcaecati quos ut laborum cupiditate eum. Ipsum, sint? Ad veritatis, molestiae repellendus dolore doloribus voluptate voluptates praesentium quidem nemo exercitationem quam eligendi, quod cum, assumenda perspiciatis hic itaque. Dolor, natus omnis. Aspernatur iure earum amet repellendus numquam doloremque fuga alias. Quibusdam!

- We can also write our own middle ware.

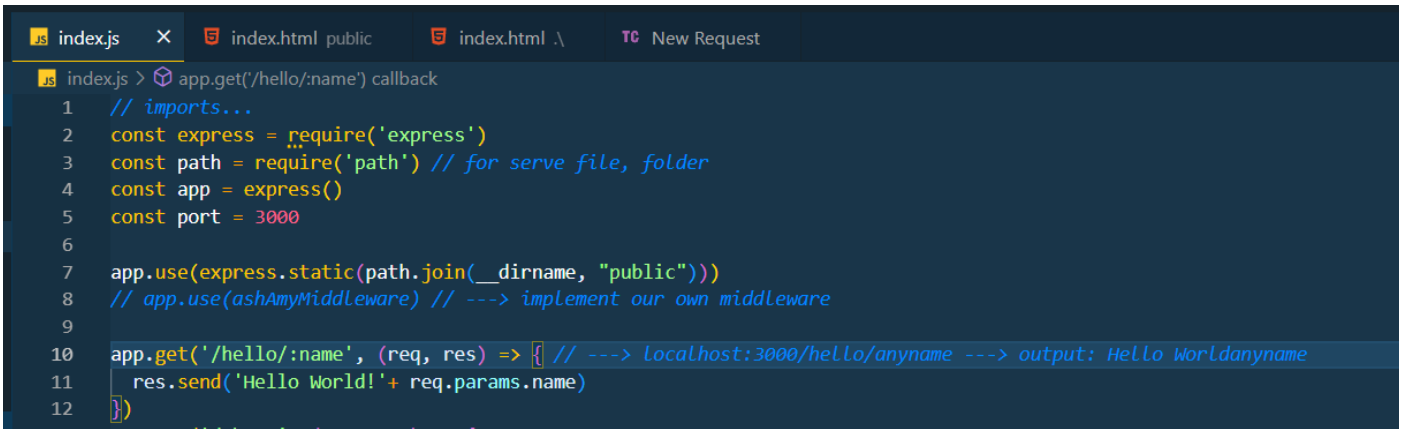


```
// imports...
const express = require('express')
const path = require('path') // for serve file, folder
const app = express()
const port = 3000

// create our own middleware
const ashAmyMiddleware = (req, res, next) => {
    console.log(req)
    next() // aa middleware pachi ana pachhi ni middleware run thay
}

// it is middleware. Te response & request access ne rakhe che. & ene change bi kari sake che.
// here app.use() ---> Is a middleware.
app.use(express.static(path.join(__dirname, "public")))
// app.use(ashAmyMiddleware) // ---> implement our own middleware

app.get('/hello', (req, res) => {
  res.send('Hello World!')
})
app.get('/about', (req, res) => {
    // res.send('about page')
    // res.sendFile(path.join(__dirname,'index.html'))
    // res.status(500) // if you want to pass server code 500: for internal server error
    res.json({"Amysh":2519}) // we can also pass json
})

app.listen(port, () => {
  console.log(`Example app listening on port ${port}`)
})
```



Lorem ipsum dolor sit amet consectetur adipisicing elit. Eveniet voluptas sunt doloremque unde amet voluptatum adipisci atque? Non, fuga illo doloremque nulla maiores qui assumenda architecto aut, quaerat est sed! Suscipit laborum vitae explicabo tempora impedit, nesciunt architecto voluptas alias ratione soluta dolorem quas aspernatur quod illum sapiente, eos recusandae necessitatibus velit ipsam odit. Numquam autem at explicabo laborum ea! Quia animi esse ex temporibus dolores architecto facere. Lorem ipsum dolor, sit amet consectetur adipisicing elit. Error illo blanditiis assumenda debitis quas veniam molestiae commodi eveniet in corporis minus, ab quia obcaecati quos ut laborum cupiditate eum. Ipsum, sint? Ad veritatis, molestiae repellendus dolore doloribus voluptate voluptates praesentium quidem nemo exercitationem quam eligendi, quod cum, assumenda perspiciatis hic itaque. Dolor, natus omnis. Aspernatur iure earum amet repellendus numquam doloremque fuga alias. Quibusdam!

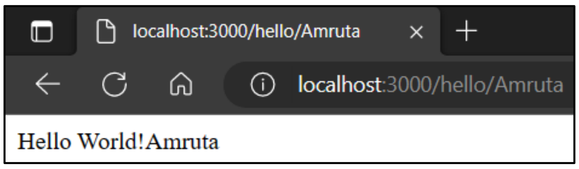
Pass any parameter.



index.js × | index.html public | index.html .\ | New Request

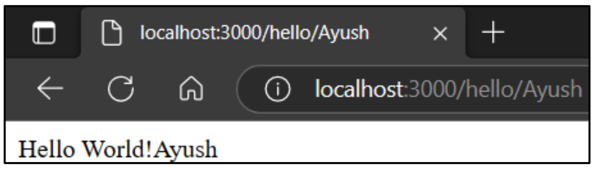
index.js > app.get('/hello/:name') callback

```
// imports...
const express = require('express')
const path = require('path') // for serve file, folder
const app = express()
const port = 3000

app.use(express.static(path.join(__dirname, "public")))
// app.use(ashAmyMiddleware) // ---> implement our own middleware

app.get('/hello/:name', (req, res) => { // ---> localhost:3000/hello/anyname ---> output: Hello Worldanyname
  res.send('Hello World!'+ req.params.name)
})
```



localhost:3000/hello/Amruta

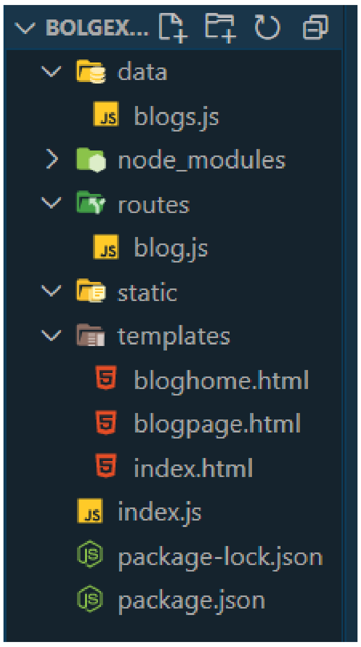
```
Hello World!Amruta
```



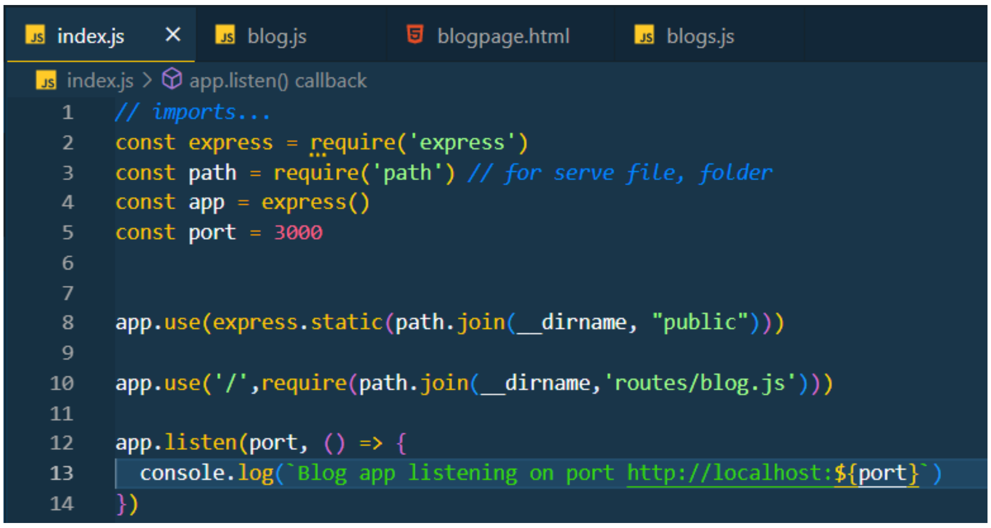
localhost:3000/hello/Ayush

```
Hello World!Ayush
```

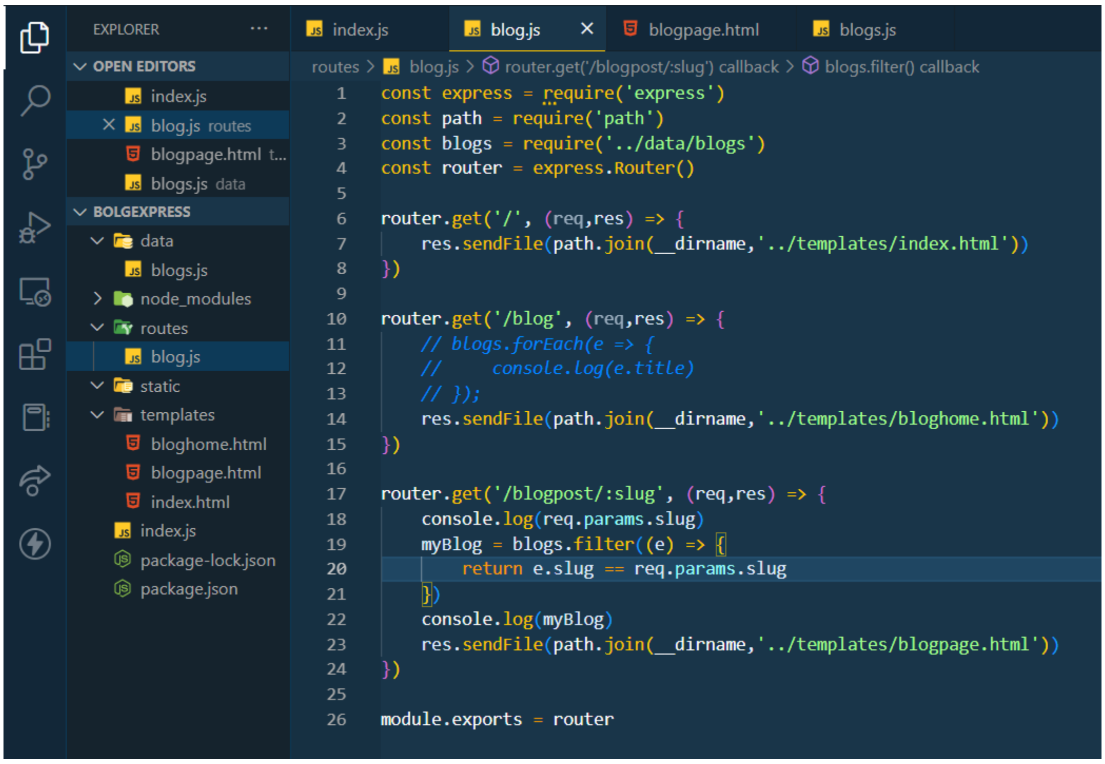
File structure:



```
BOLGEX...
  data
    blogs.js
  node_modules
  routes
    blog.js
  static
  templates
    bloghome.html
    blogpage.html
    index.html
  index.js
  package-lock.json
  package.json
```

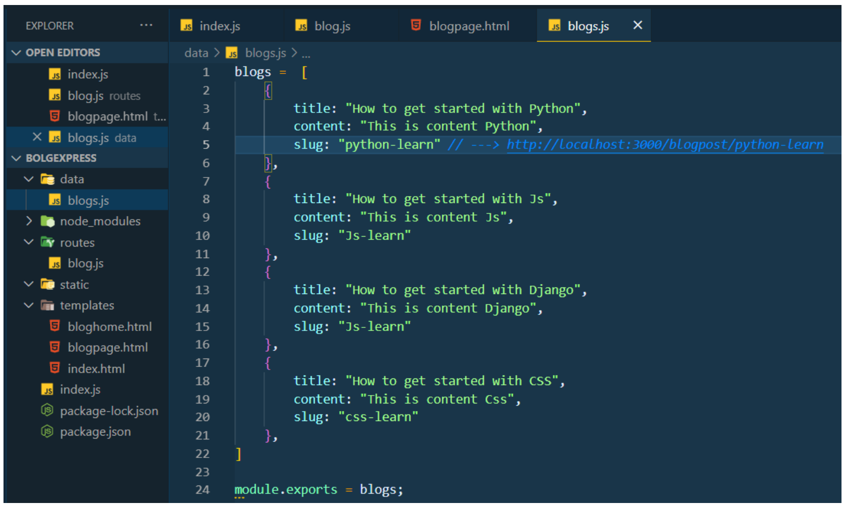


```javascript
// imports...
const express = require('express')
const path = require('path') // for serve file, folder
const app = express()
const port = 3000


app.use(express.static(path.join(__dirname, "public")))

app.use('/',require(path.join(__dirname,'routes/blog.js')))

app.listen(port, () => {
    console.log(`Blog app listening on port http://localhost:${port}`)
})
```

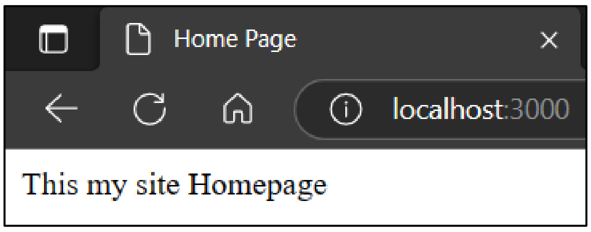


```javascript
const express = require('express')
const path = require('path')
const blogs = require('../data/blogs')
const router = express.Router()

router.get('/', (req,res) => {
    res.sendFile(path.join(__dirname,'../templates/index.html'))
})

router.get('/blog', (req,res) => {
    // blogs.forEach(e => {
    //     console.log(e.title)
    // });
    res.sendFile(path.join(__dirname,'../templates/bloghome.html'))
})

router.get('/blogpost/:slug', (req,res) => {
    console.log(req.params.slug)
    myBlog = blogs.filter((e) => {
        return e.slug == req.params.slug
    })
    console.log(myBlog)
    res.sendFile(path.join(__dirname,'../templates/blogpage.html'))
})

module.exports = router
```

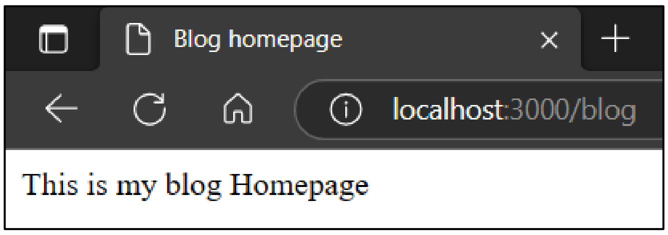
s



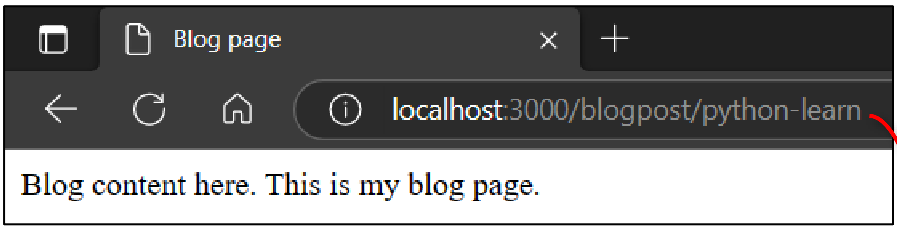
```
blogs = [
    {
        title: "How to get started with Python",
        content: "This is content Python",
        slug: "python-learn" // ---> http://localhost:3000/blogpost/python-learn
    },
    {
        title: "How to get started with Js",
        content: "This is content Js",
        slug: "Js-learn"
    },
    {
        title: "How to get started with Django",
        content: "This is content Django",
        slug: "Js-learn"
    },
    {
        title: "How to get started with CSS",
        content: "This is content Css",
        slug: "css-learn"
    },
]

module.exports = blogs;
```



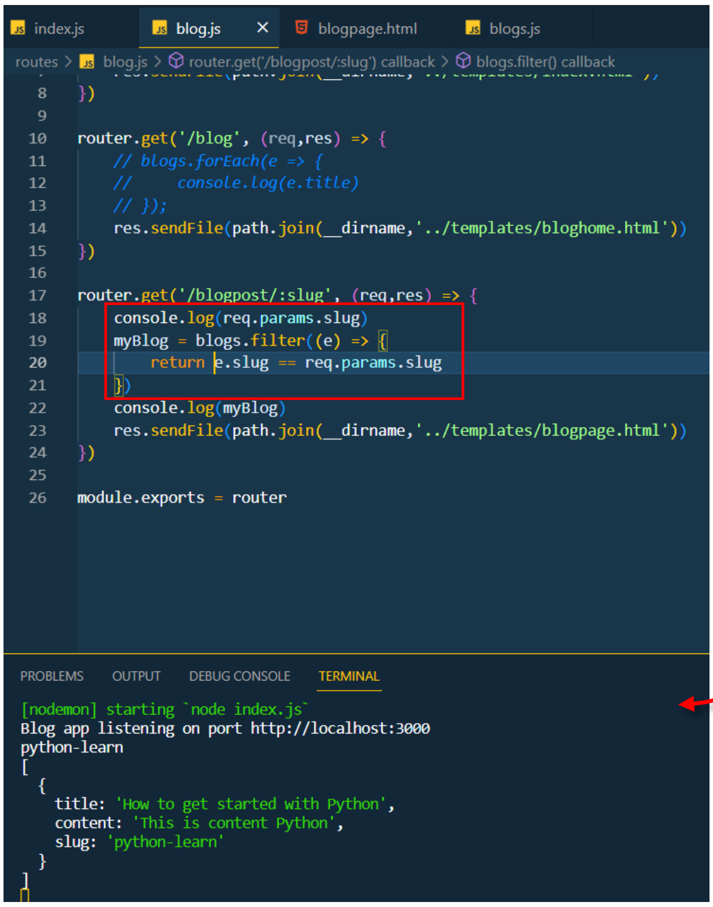
This my site Homepage



This is my blog Homepage



Blog content here. This is my blog page.


index.js
blog.js
blogpage.html
blogs.js
routes > blog.js > router.get('/blogpost/:slug') callback > blogs.filter() callback
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL

```
})

router.get('/blog', (req,res) => {
    // blogs.forEach(e => {
    //     console.log(e.title)
    // });
    res.sendFile(path.join(__dirname,'../templates/bloghome.html'))
})

router.get('/blogpost/:slug', (req,res) => {
    console.log(req.params.slug)
    myBlog = blogs.filter((e) => {
        return e.slug == req.params.slug
    })
    console.log(myBlog)
    res.sendFile(path.join(__dirname,'../templates/blogpage.html'))
})

module.exports = router
```

```
[nodemon] starting `node index.js`
Blog app listening on port http://localhost:3000
python-learn
[
  {
    title: 'How to get started with Python',
    content: 'This is content Python',
    slug: 'python-learn'
  }
]
```

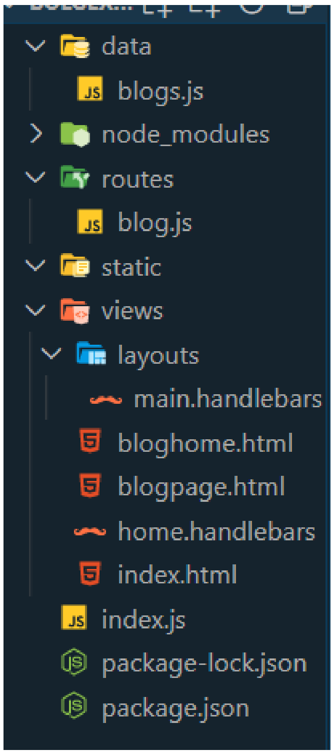
- Now use with database we use: mongoose.
- Which is use to connect with database.

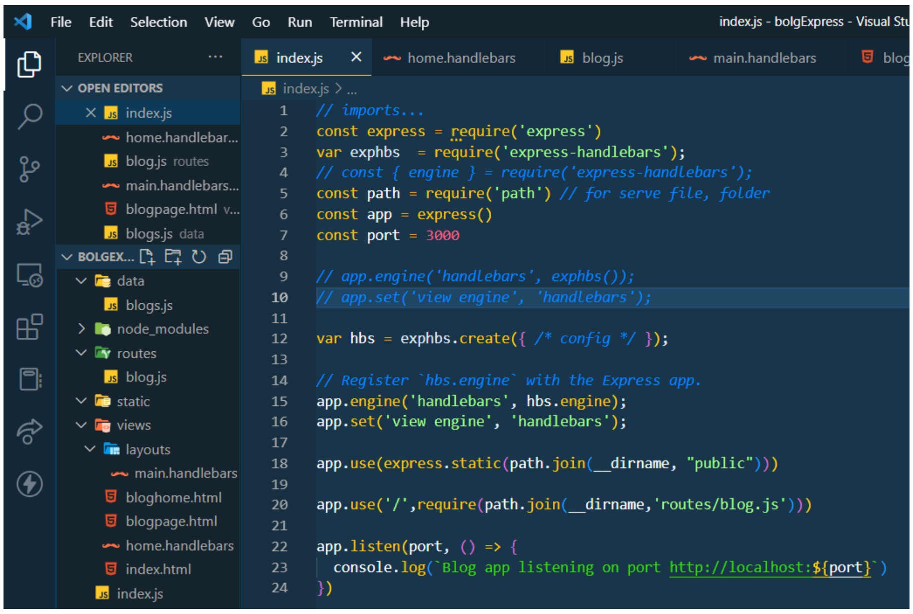
Now, we use handlebars.

- Express handlebars & (Is a npm package)
- Handlebars (Is templating engine which is very helpful for JavaScrip templating)

```
$ npm install express-handlebars
```

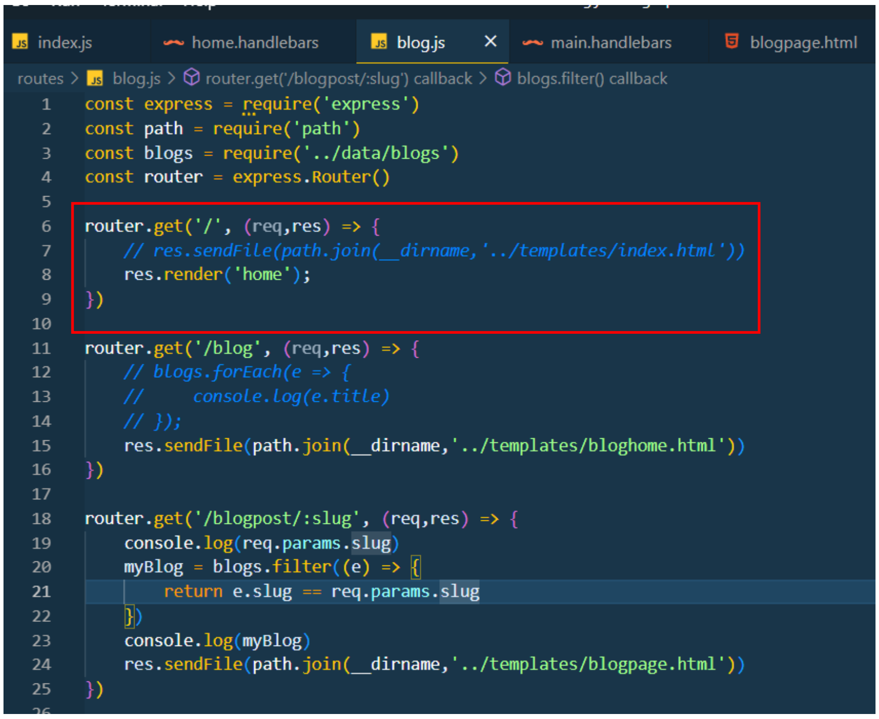
Follow this docs in above project



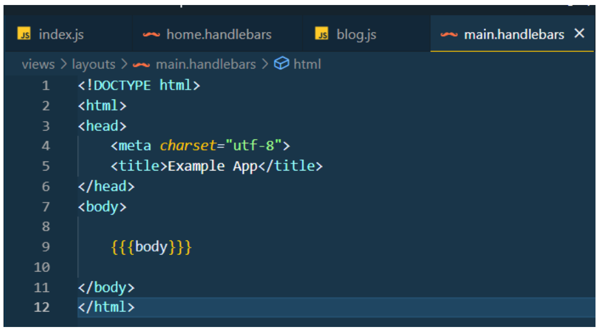

File Edit Selection View Go Run Terminal Help
index.js - bolgExpress - Visual Stu
EXPLORER
OPEN EDITORS
index.js
home.handlebars
blog.js
main.handlebars
blog

```
// imports...
const express = require('express')
var exphbs  = require('express-handlebars');
// const { engine } = require('express-handlebars');
const path = require('path') // for serve file, folder
const app = express()
const port = 3000

// app.engine('handlebars', exphbs());
// app.set('view engine', 'handlebars');

var hbs = exphbs.create({ /* config */ });

// Register `hbs.engine` with the Express app.
app.engine('handlebars', hbs.engine);
app.set('view engine', 'handlebars');

app.use(express.static(path.join(__dirname, "public")))

app.use('/',require(path.join(__dirname,'routes/blog.js')))

app.listen(port, () => {
  console.log(`Blog app listening on port http://localhost:${port}`)
})
```

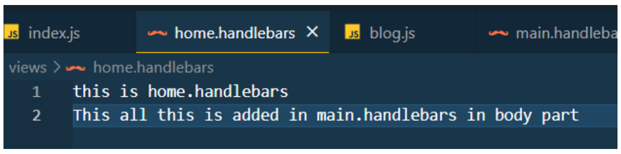
s


index.js
home.handlebars
blog.js
main.handlebars
blogpage.html
routes > blog.js > router.get('/blogpost/:slug') callback > blogs.filter() callback

```
const express = require('express')
const path = require('path')
const blogs = require('../data/blogs')
const router = express.Router()

router.get('/', (req,res) => {
    // res.sendFile(path.join(__dirname,'../templates/index.html'))
    res.render('home');
})

router.get('/blog', (req,res) => {
    // blogs.forEach(e => {
    //     console.log(e.title)
    // });
    res.sendFile(path.join(__dirname,'../templates/bloghome.html'))
})

router.get('/blogpost/:slug', (req,res) => {
    console.log(req.params.slug)
    myBlog = blogs.filter((e) => {
        return e.slug == req.params.slug
    })
    console.log(myBlog)
    res.sendFile(path.join(__dirname,'../templates/blogpage.html'))
})
```

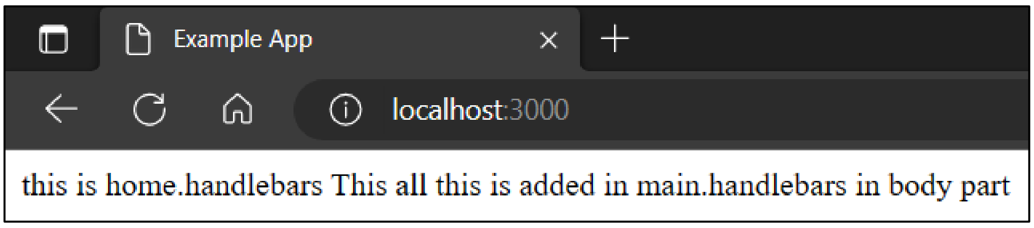


```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title>Example App</title>
</head>
<body>

    {{{body}}}

</body>
</html>
```



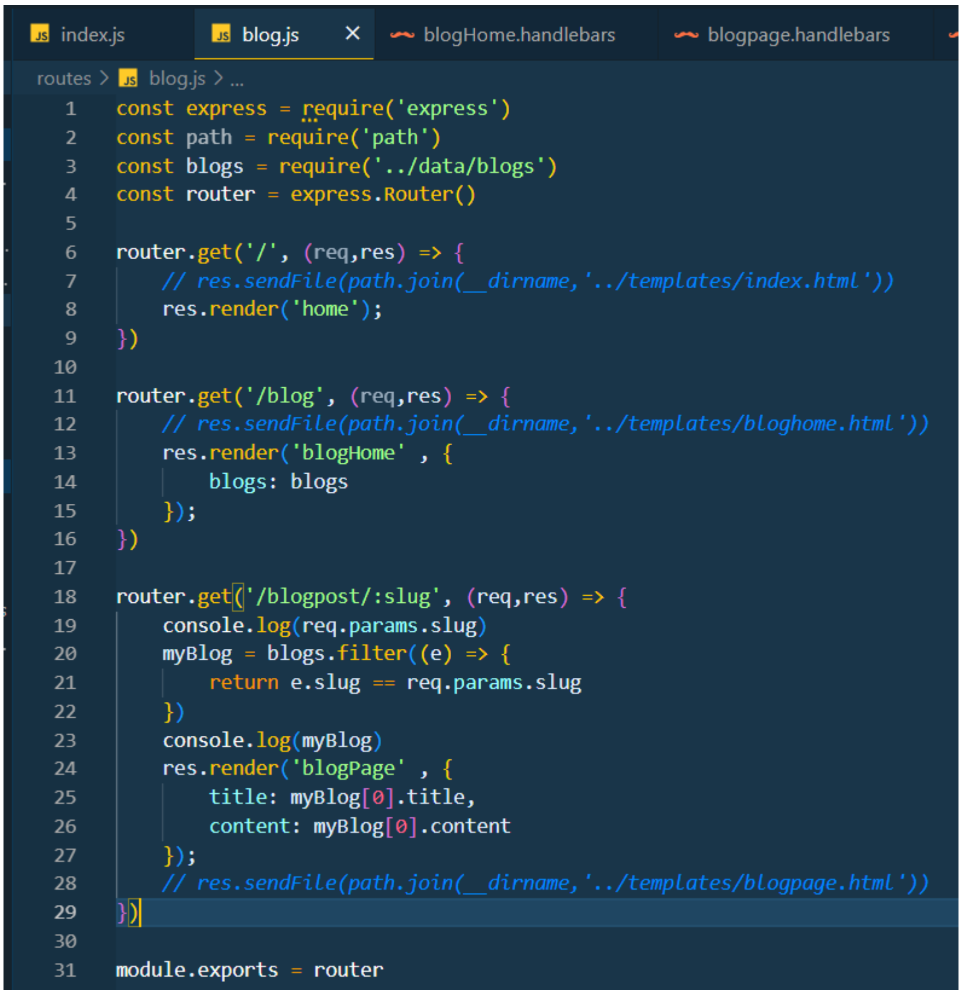
```
this is home.handlebars
This all this is added in main.handlebars in body part
```

Output:



```
this is home.handlebars This all this is added in main.handlebars in body part
```

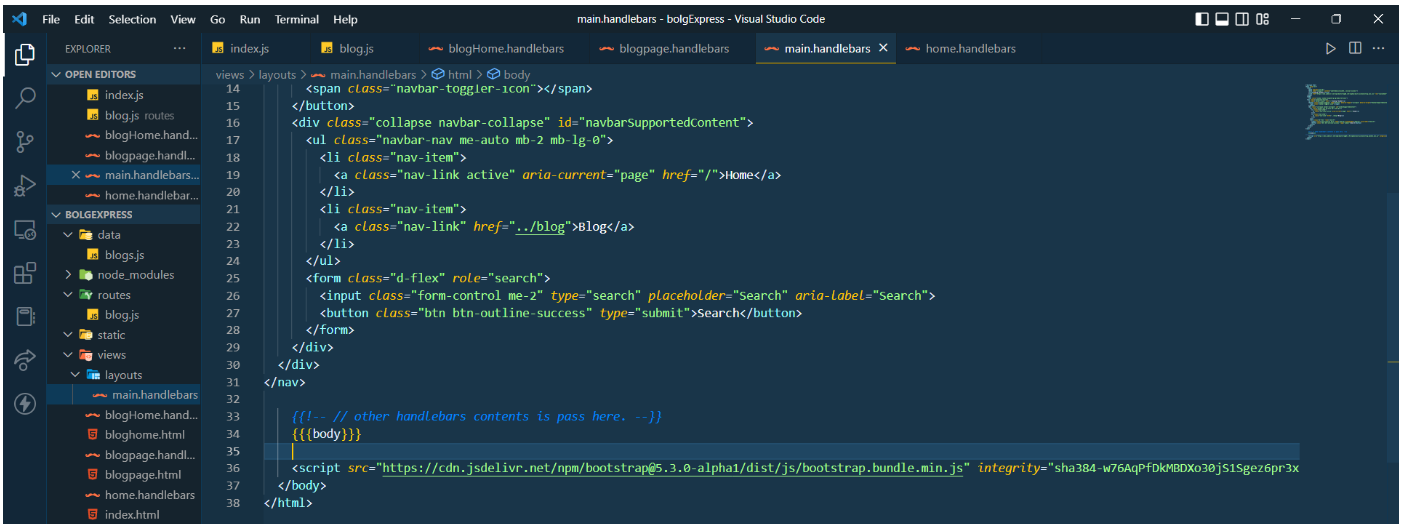
- Now create other files in view folder.
  - blogHome.handlebars
  - blogPage.handlebars



index.js | blog.js | blogHome.handlebars | blogpage.handlebars

routes > blog.js > ...

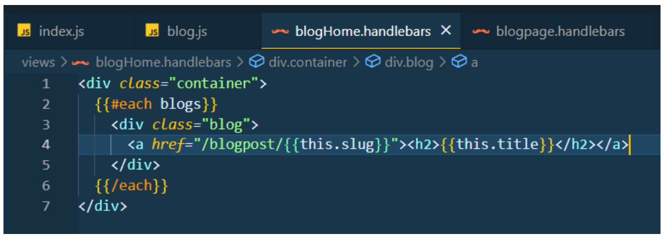
```javascript
const express = require('express')
const path = require('path')
const blogs = require('../data/blogs')
const router = express.Router()

router.get('/', (req,res) => {
    // res.sendFile(path.join(__dirname,'../templates/index.html'))
    res.render('home');
})

router.get('/blog', (req,res) => {
    // res.sendFile(path.join(__dirname,'../templates/bloghome.html'))
    res.render('blogHome' , {
        blogs: blogs
    });
})

router.get('/blogpost/:slug', (req,res) => {
    console.log(req.params.slug)
    myBlog = blogs.filter((e) => {
        return e.slug == req.params.slug
    })
    console.log(myBlog)
    res.render('blogPage' , {
        title: myBlog[0].title,
        content: myBlog[0].content
    });
    // res.sendFile(path.join(__dirname,'../templates/blogpage.html'))
})

module.exports = router
```



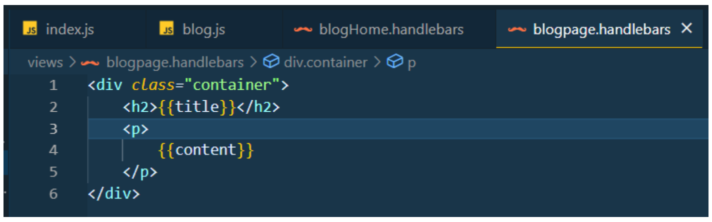
main.handlebars - bolgExpress - Visual Studio Code

views > layouts > main.handlebars > html > body

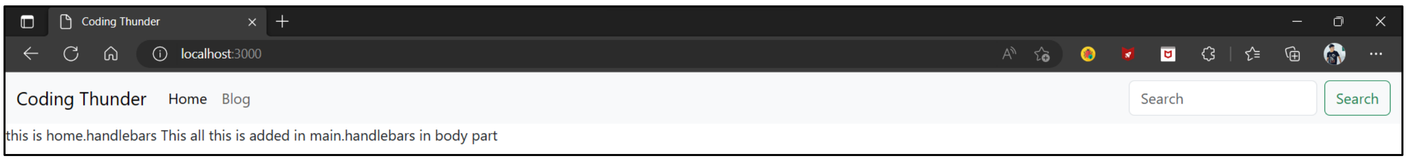
```html
        <span class="navbar-toggler-icon"></span>
      </button>
      <div class="collapse navbar-collapse" id="navbarSupportedContent">
        <ul class="navbar-nav me-auto mb-2 mb-lg-0">
          <li class="nav-item">
            <a class="nav-link active" aria-current="page" href="/">Home</a>
          </li>
          <li class="nav-item">
            <a class="nav-link" href="../blog">Blog</a>
          </li>
        </ul>
        <form class="d-flex" role="search">
          <input class="form-control me-2" type="search" placeholder="Search" aria-label="Search">
          <button class="btn btn-outline-success" type="submit">Search</button>
        </form>
      </div>
    </div>
  </nav>

    {{!-- // other handlebars contents is pass here. --}}
    {{{body}}}

    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/js/bootstrap.bundle.min.js" integrity="sha384-w76AqPfDkMBDXo30jS1Sgez6pr3x
  </body>
</html>
```



```
<div class="container">
  {{#each blogs}}
    <div class="blog">
      <a href="/blogpost/{{this.slug}}"><h2>{{this.title}}</h2></a>
    </div>
  {{/each}}
</div>
```



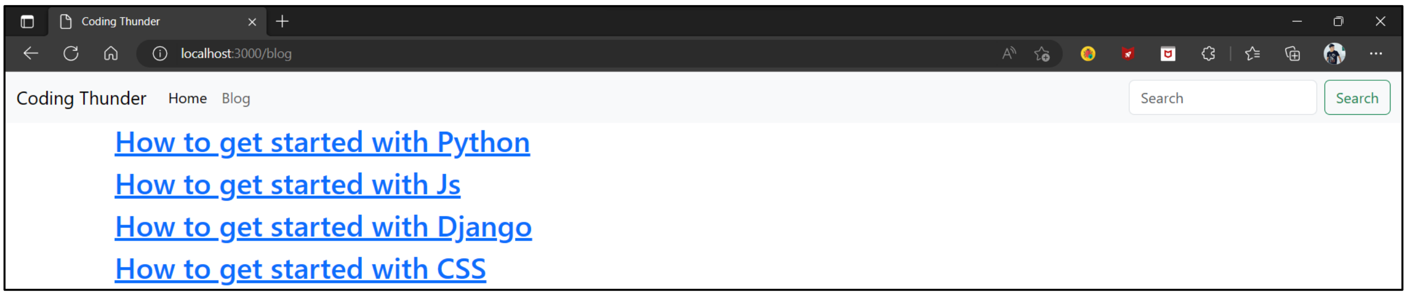
```
<div class="container">
    <h2>{{title}}</h2>
    <p>
        {{content}}
    </p>
</div>
```



localhost:3000

Coding Thunder Home Blog

this is home.handlebars This all this is added in main.handlebars in body part

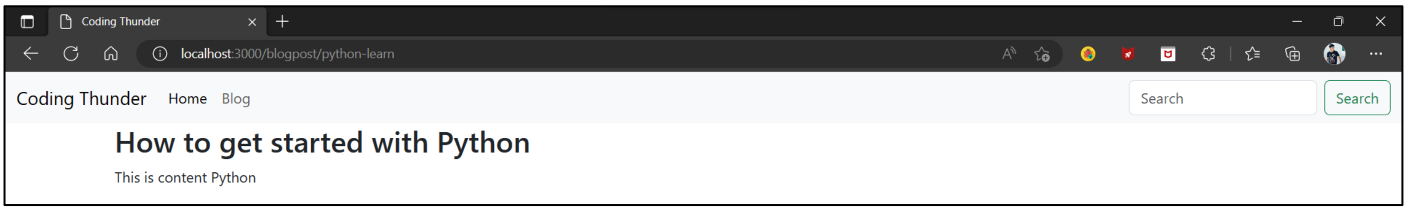


localhost:3000/blog

Coding Thunder Home Blog

How to get started with Python
How to get started with Js
How to get started with Django
How to get started with CSS

Click any one of link.



localhost:3000/blogpost/python-learn

Coding Thunder Home Blog

How to get started with Python

This is content Python