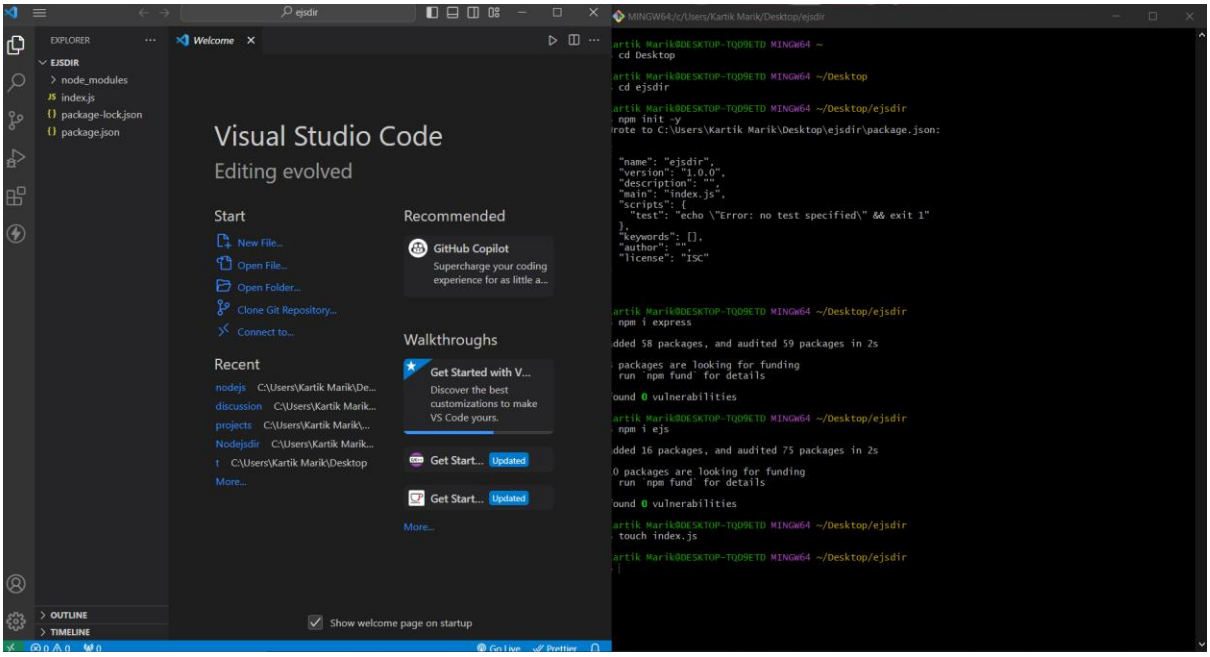
Contents

EJS

- Ejs is simple templating language that lets you generate html markup with plain javascript
- We can call it as a templating tool

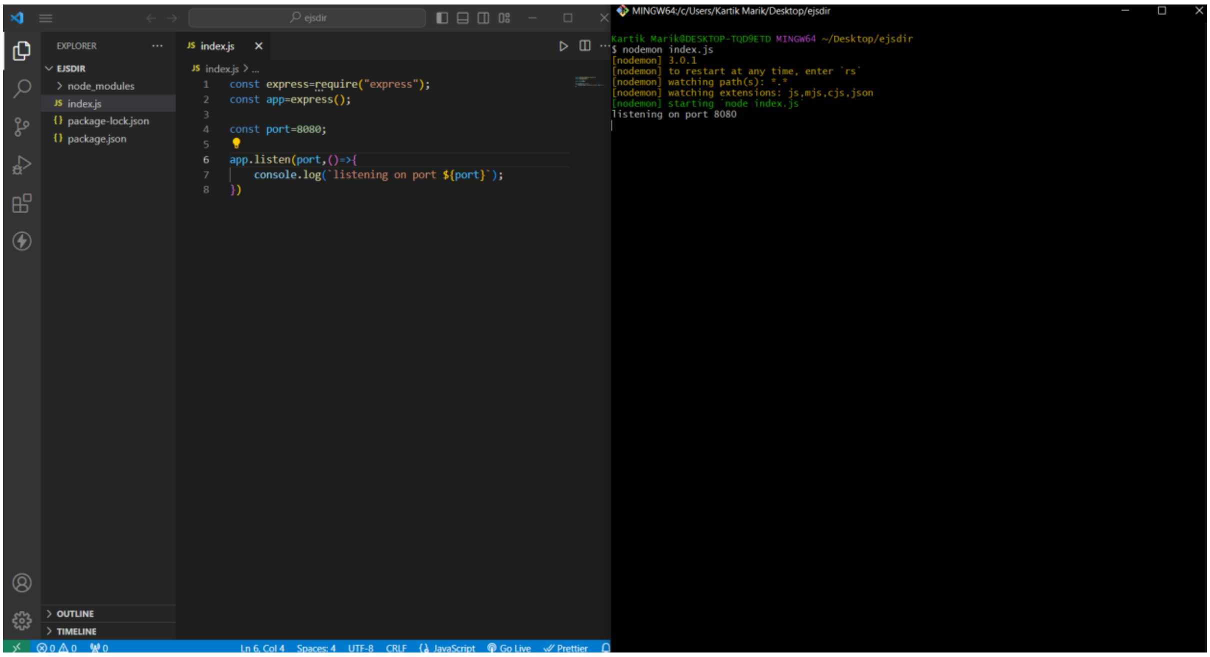
Examples

- Do these basic set up
- Make a folder named ejsdir and inside that do these basic setup



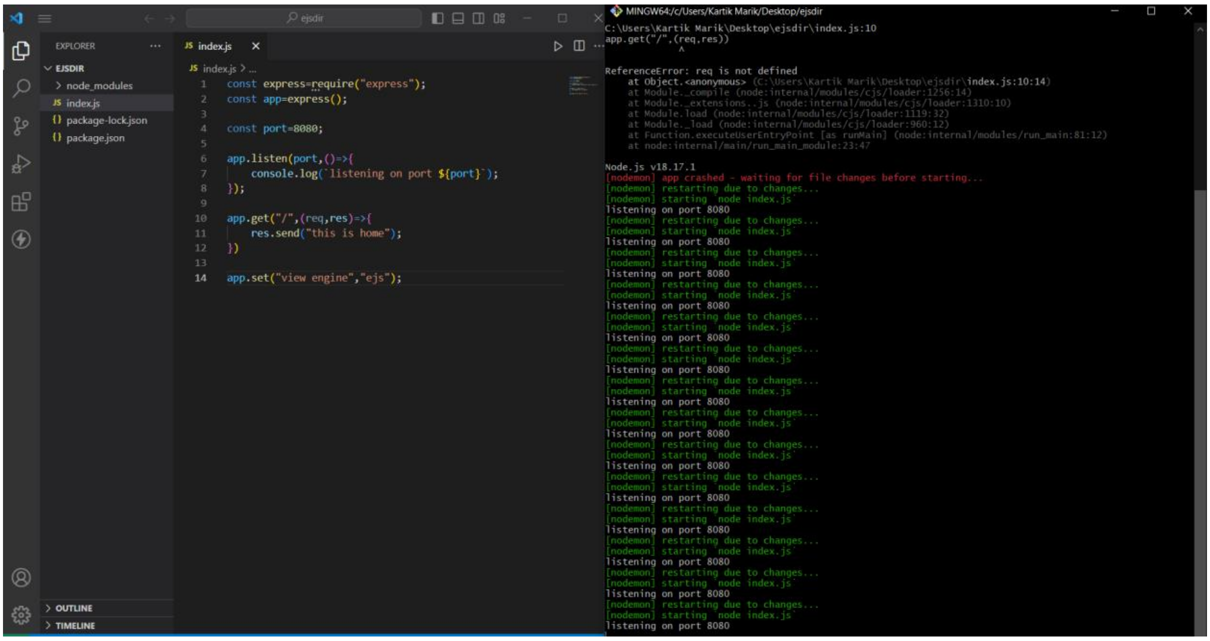
```
artik Marik@DESKTOP-TQD9ETD MINGW64 ~
 cd Desktop

artik Marik@DESKTOP-TQD9ETD MINGW64 ~/Desktop
 cd ejsdir

artik Marik@DESKTOP-TQD9ETD MINGW64 ~/Desktop/ejsdir
 npm init -y
rote to C:\Users\Kartik Marik\Desktop\ejsdir\package.json:


 "name": "ejsdir",
 "version": "1.0.0",
 "description": "",
 "main": "index.js",
 "scripts": {
   "test": "echo \"Error: no test specified\" && exit 1"
 },
 "keywords": [],
 "author": "",
 "license": "ISC"



artik Marik@DESKTOP-TQD9ETD MINGW64 ~/Desktop/ejsdir
 npm i express

dded 58 packages, and audited 59 packages in 2s

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

ound 0 vulnerabilities

artik Marik@DESKTOP-TQD9ETD MINGW64 ~/Desktop/ejsdir
 npm i ejs

dded 16 packages, and audited 75 packages in 2s

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

ound 0 vulnerabilities

artik Marik@DESKTOP-TQD9ETD MINGW64 ~/Desktop/ejsdir
 touch index.js

artik Marik@DESKTOP-TQD9ETD MINGW64 ~/Desktop/ejsdir
```

- Set the server


ejsdir
EXPLORER
EJSDIR
node_modules
index.js
package-lock.json
package.json
OUTLINE
TIMELINE
Ln 6, Col 4 Spaces: 4 UTF-8 CRLF JavaScript Go Live Prettier
MINGW64:/c/Users/Kartik Marik/Desktop/ejsdir

```javascript
const express=require("express");
const app=express();

const port=8080;

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

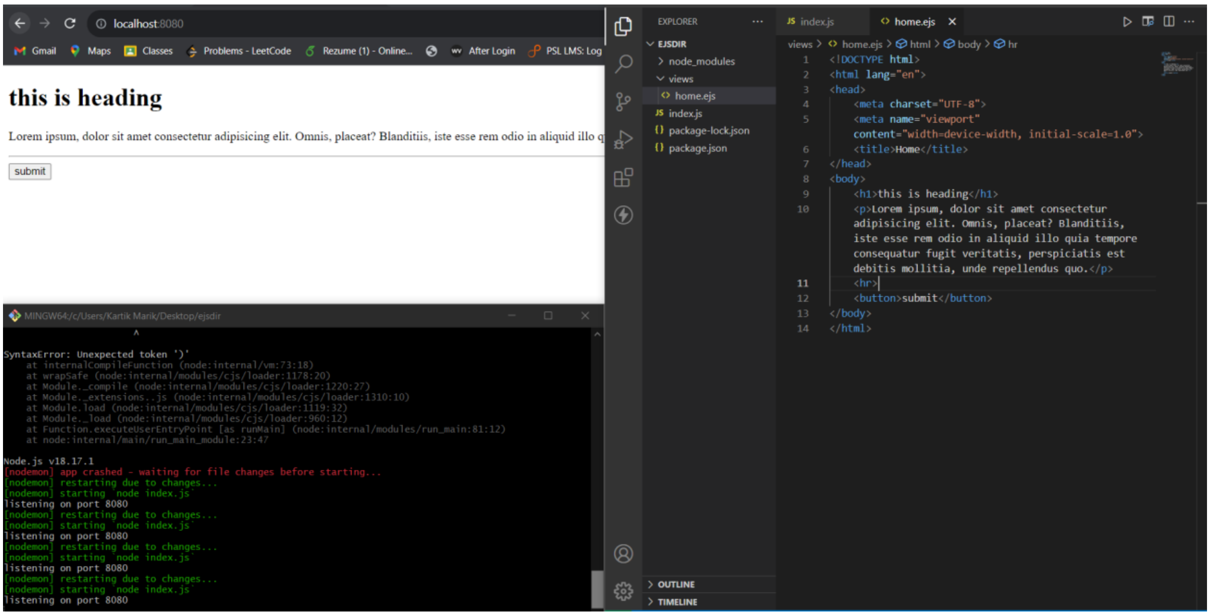
```
Kartik Marik@DESKTOP-TQD9ETD MINGW64 ~/Desktop/ejsdir
$ nodemon index.js
[nodemon] 3.0.1
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting `node index.js`
listening on port 8080
```



```js
const express=require("express");
const app=express();

const port=8080;

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

app.get("/",(req,res)=>{
    res.send("this is home");
})

app.set("view engine","ejs");
```

```
MINGW64:/c/Users/Kartik Marik/Desktop/ejsdir
C:\Users\Kartik Marik\Desktop\ejsdir\index.js:10
app.get("/",(req,res))
             ^

ReferenceError: req is not defined
    at Object.<anonymous> (C:\Users\Kartik Marik\Desktop\ejsdir\index.js:10:14)
    at Module._compile (node:internal/modules/cjs/loader:1256:14)
    at Module._extensions..js (node:internal/modules/cjs/loader:1310:10)
    at Module.load (node:internal/modules/cjs/loader:1119:32)
    at Module._load (node:internal/modules/cjs/loader:960:12)
    at Function.executeUserEntryPoint [as runMain] (node:internal/modules/run_main:81:12)
    at node:internal/main/run_main_module:23:47

Node.js v18.17.1
[nodemon] app crashed - waiting for file changes before starting...
[nodemon] restarting due to changes...
[nodemon] starting `node index.js`
listening on port 8080
[nodemon] restarting due to changes...
[nodemon] starting `node index.js`
listening on port 8080
[nodemon] restarting due to changes...
[nodemon] starting `node index.js`
listening on port 8080
[nodemon] restarting due to changes...
[nodemon] starting `node index.js`
listening on port 8080
[nodemon] restarting due to changes...
[nodemon] starting `node index.js`
listening on port 8080
[nodemon] restarting due to changes...
[nodemon] starting `node index.js`
listening on port 8080
[nodemon] restarting due to changes...
[nodemon] starting `node index.js`
listening on port 8080
[nodemon] restarting due to changes...
[nodemon] starting `node index.js`
listening on port 8080
[nodemon] restarting due to changes...
[nodemon] starting `node index.js`
listening on port 8080
[nodemon] restarting due to changes...
[nodemon] starting `node index.js`
listening on port 8080
[nodemon] restarting due to changes...
[nodemon] starting `node index.js`
listening on port 8080
[nodemon] restarting due to changes...
[nodemon] starting `node index.js`
listening on port 8080
[nodemon] restarting due to changes...
[nodemon] starting `node index.js`
listening on port 8080
[nodemon] restarting due to changes...
[nodemon] starting `node index.js`
listening on port 8080
```

- Previously we used to send string, array,object html via res.send
- What about if I have to send files
- For that we have to set view engine to ejs
- We have to make a folder named views inside ejsdir folder and whatever we want file to be sent we will make here only



this is heading

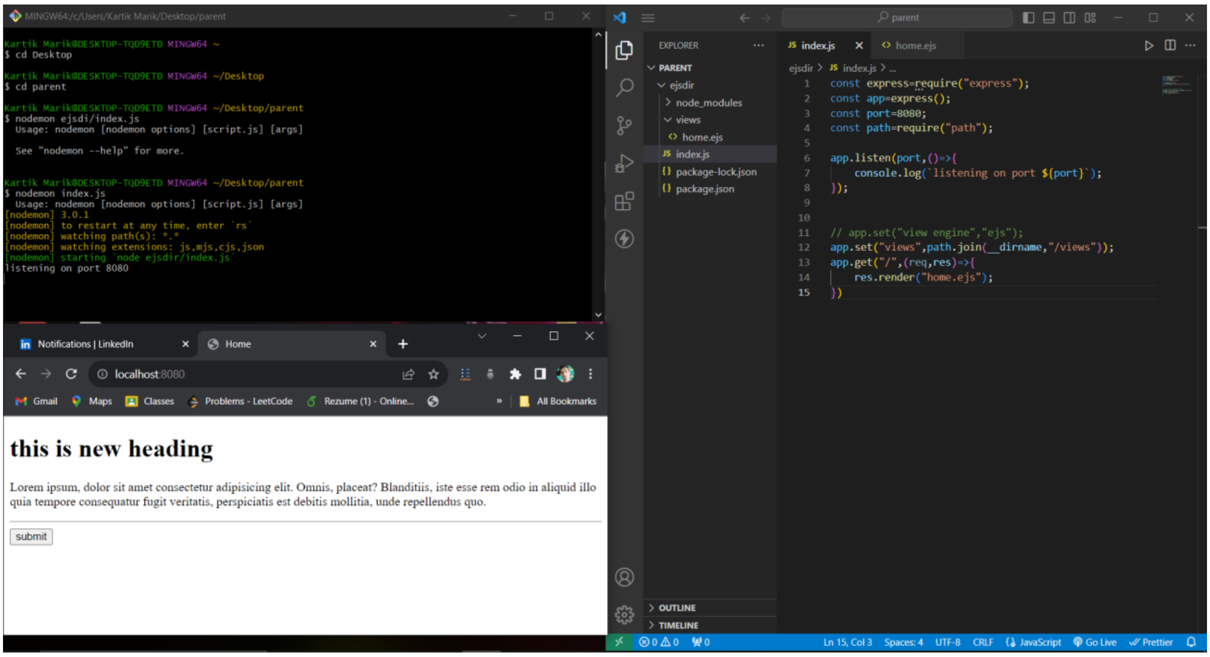
Lorem ipsum, dolor sit amet consectetur adipisicing elit. Omnis, placeat? Blanditiis, iste esse rem odio in aliquid illo q

submit

```
MINGW64:/c/Users/Kartik Marik/Desktop/ejsdir
                    ^

SyntaxError: Unexpected token ')'
    at internalCompileFunction (node:internal/vm:73:18)
    at wrapSafe (node:internal/modules/cjs/loader:1178:20)
    at Module._compile (node:internal/modules/cjs/loader:1220:27)
    at Module._extensions..js (node:internal/modules/cjs/loader:1310:10)
    at Module.load (node:internal/modules/cjs/loader:1119:32)
    at Module._load (node:internal/modules/cjs/loader:960:12)
    at Function.executeUserEntryPoint [as runMain] (node:internal/modules/run_main:81:12)
    at node:internal/main/run_main_module:23:47

Node.js v18.17.1
[nodemon] app crashed - waiting for file changes before starting...
[nodemon] restarting due to changes...
[nodemon] starting `node index.js`
listening on port 8080
[nodemon] restarting due to changes...
[nodemon] starting `node index.js`
listening on port 8080
[nodemon] restarting due to changes...
[nodemon] starting `node index.js`
listening on port 8080
[nodemon] restarting due to changes...
[nodemon] starting `node index.js`
listening on port 8080
```

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport"
    content="width=device-width, initial-scale=1.0">
    <title>Home</title>
</head>
<body>
    <h1>this is heading</h1>
    <p>Lorem ipsum, dolor sit amet consectetur
    adipisicing elit. Omnis, placeat? Blanditiis,
    iste esse rem odio in aliquid illo quia tempore
    consequatur fugit veritatis, perspiciatis est
    debitis mollitia, unde repellendus quo.</p>
    <hr>
    <button>submit</button>
</body>
</html>
```

Do changes if you call indexjs from folder inside which views folder is not there so



```
Kartik Marik@DESKTOP-TQD9ETD MINGW64 ~
$ cd Desktop

Kartik Marik@DESKTOP-TQD9ETD MINGW64 ~/Desktop
$ cd parent

Kartik Marik@DESKTOP-TQD9ETD MINGW64 ~/Desktop/parent
$ nodemon ejsdi/index.js
  Usage: nodemon [nodemon options] [script.js] [args]

  See "nodemon --help" for more.


Kartik Marik@DESKTOP-TQD9ETD MINGW64 ~/Desktop/parent
$ nodemon index.js
  Usage: nodemon [nodemon options] [script.js] [args]
[nodemon] 3.0.1
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting `node ejsdir/index.js`
listening on port 8080
```

localhost:8080

this is new heading

Lorem ipsum, dolor sit amet consectetur adipisicing elit. Omnis, placeat? Blanditiis, iste esse rem odio in aliquid illo quia tempore consequatur fugit veritatis, perspiciatis est debitis mollitia, unde repellendus quo.

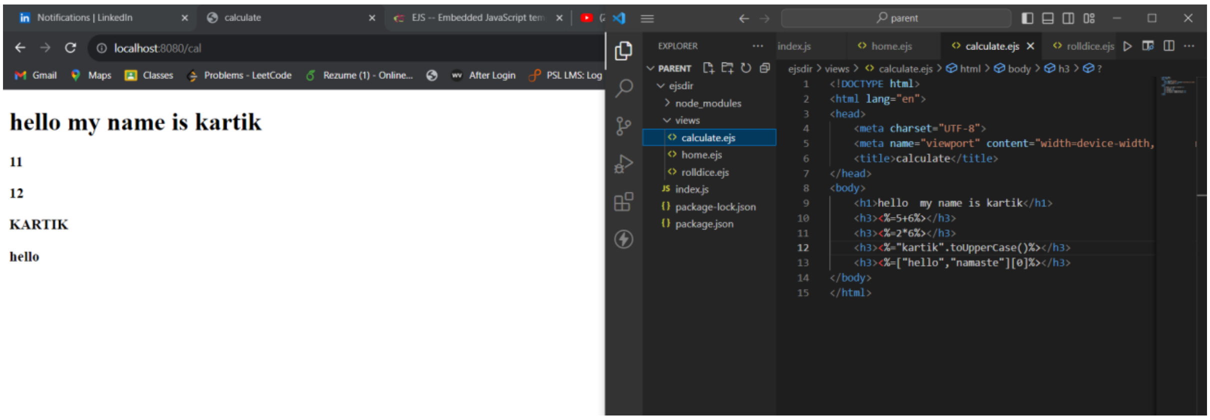
submit

ejsdir > index.js

```
const express=require("express");
const app=express();
const port=8080;
const path=require("path");

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


// app.set("view engine","ejs");
app.set("views",path.join(__dirname,"/views"));
app.get("/",(req,res)=>{
    res.render("home.ejs");
})
```

Interpolation syntax

- Interpolation refers to embedding expression into marked up text.
- Documentation of ejs----->https://ejs.co/#docs
- Go to tags and see syntax
- Parsing data to ejs



localhost:8080/cal

hello my name is kartik

11

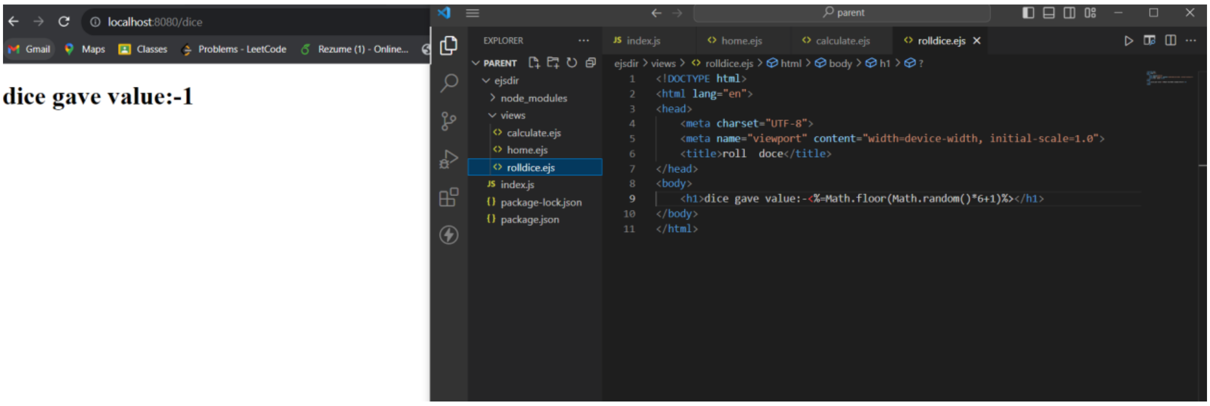
12

KARTIK

hello

ejsdir > views > calculate.ejs > html > body > h3 > ?

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
    <title>calculate</title>
</head>
<body>
    <h1>hello  my name is kartik</h1>
    <h3><%=5+6%></h3>
    <h3><%=2*6%></h3>
    <h3><%="kartik".toUpperCase()%></h3>
    <h3><%=["hello","namaste"][0]%></h3>
</body>
</html>
```

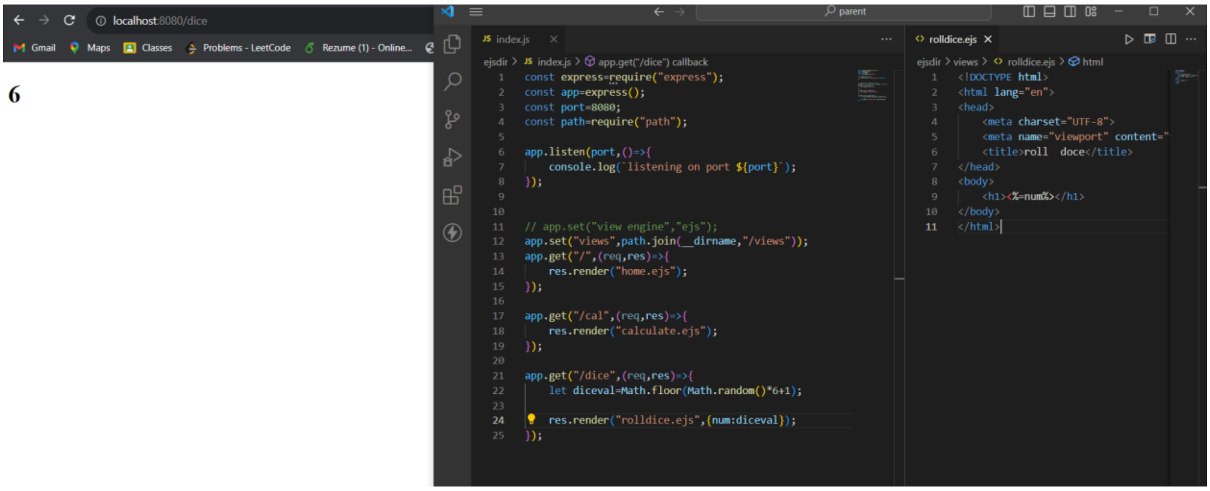


localhost:8080/dice

dice gave value:-1

ejsdir > views > rolldice.ejs > html > body > h1 > ?

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>roll  doce</title>
</head>
<body>
    <h1>dice gave value:-<%=Math.floor(Math.random()*6+1)%></h1>
</body>
</html>
```

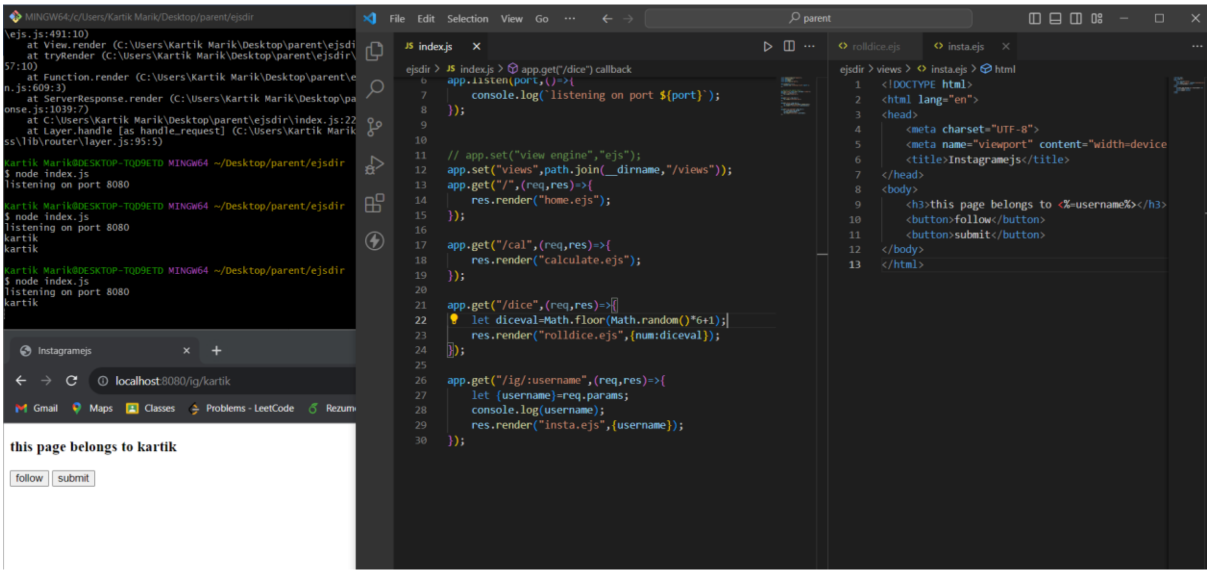


6

```
const express=require("express");
const app=express();
const port=8080;
const path=require("path");

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


// app.set("view engine","ejs");
app.set("views",path.join(__dirname,"/views"));
app.get("/",(req,res)=>{
    res.render("home.ejs");
});

app.get("/cal",(req,res)=>{
    res.render("calculate.ejs");
});

app.get("/dice",(req,res)=>{
    let diceval=Math.floor(Math.random()*6+1);

    res.render("rolldice.ejs",{num:diceval});
});
```

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="
    <title>roll  doce</title>
</head>
<body>
    <h1><%=num%></h1>
</body>
</html>
```

Instagram ejs



```
\ejs.js:491:10)
    at View.render (C:\Users\Kartik Marik\Desktop\parent\ejsdi
    at tryRender (C:\Users\Kartik Marik\Desktop\parent\ejsdir\
57:10)
    at Function.render (C:\Users\Kartik Marik\Desktop\parent\e
n.js:609:3)
    at ServerResponse.render (C:\Users\Kartik Marik\Desktop\pa
onse.js:1039:7)
    at C:\Users\Kartik Marik\Desktop\parent\ejsdir\index.js:22
    at Layer.handle [as handle_request] (C:\Users\Kartik Marik
ss\lib\router\layer.js:95:5)

Kartik Marik@DESKTOP-TQD9ETD MINGW64 ~/Desktop/parent/ejsdir
$ node index.js
listening on port 8080

Kartik Marik@DESKTOP-TQD9ETD MINGW64 ~/Desktop/parent/ejsdir
$ node index.js
listening on port 8080
kartik
kartik

Kartik Marik@DESKTOP-TQD9ETD MINGW64 ~/Desktop/parent/ejsdir
$ node index.js
listening on port 8080
kartik
```

this page belongs to kartik

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


// app.set("view engine","ejs");
app.set("views",path.join(__dirname,"/views"));
app.get("/",(req,res)=>{
    res.render("home.ejs");
});

app.get("/cal",(req,res)=>{
    res.render("calculate.ejs");
});

app.get("/dice",(req,res)=>{
    let diceval=Math.floor(Math.random()*6+1);
    res.render("rolldice.ejs",{num:diceval});
});

app.get("/ig/:username",(req,res)=>{
    let {username}=req.params;
    console.log(username);
    res.render("insta.ejs",{username});
});
```

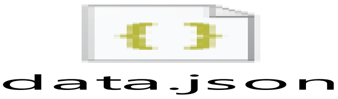
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device
    <title>Instagramejs</title>
</head>
<body>
    <h3>this page belongs to <%=username%></h3>
    <button>follow</button>
    <button>submit</button>
</body>
</html>
```

Conditional statement in ejs

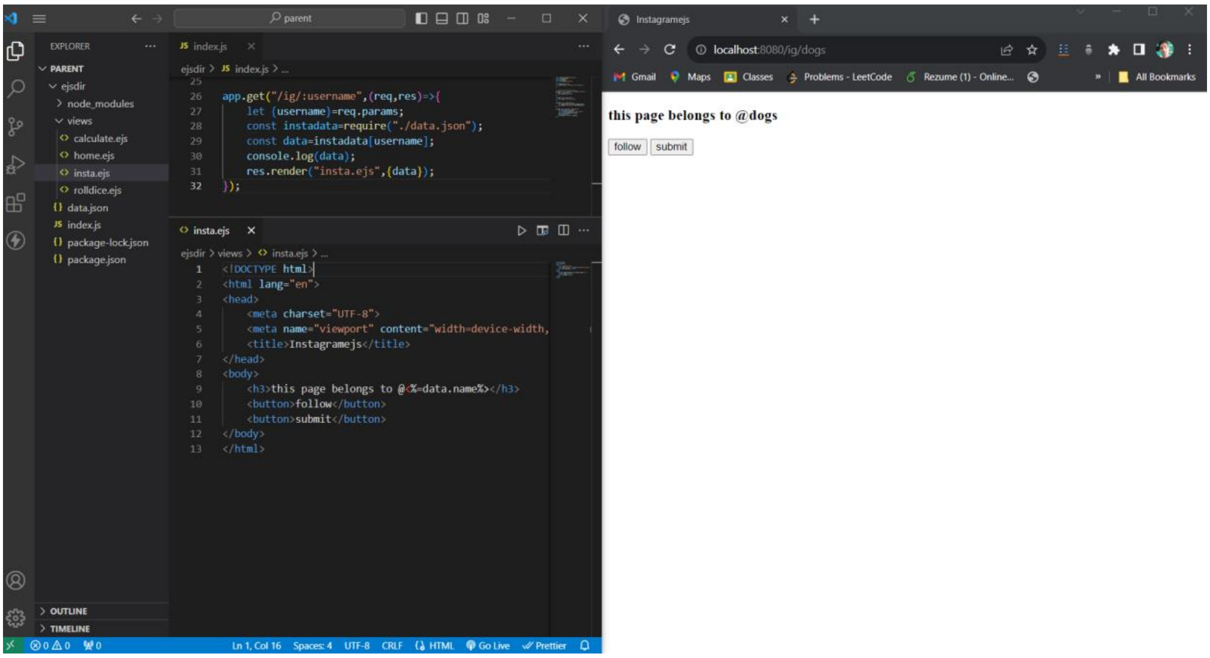
Loops in ejs

Instagram page with ejs

- Data.json file---->



-
- Include this data.json file inside ejsdir folder



ejsdir > index.js > ...

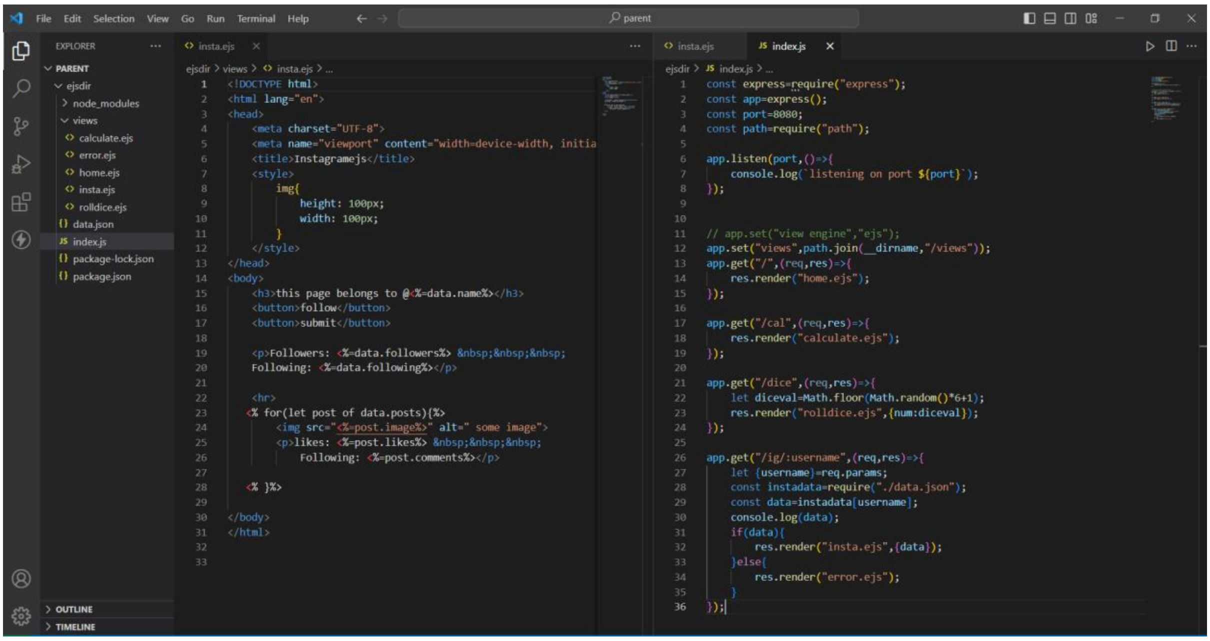
```js
app.get("/ig/:username",(req,res)=>{
    let {username}=req.params;
    const instadata=require("./data.json");
    const data=instadata[username];
    console.log(data);
    res.render("insta.ejs",{data});
});
```

ejsdir > views > insta.ejs > ...

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
    <title>Instagramejs</title>
</head>
<body>
    <h3>this page belongs to @<%=data.name%></h3>
    <button>follow</button>
    <button>submit</button>
</body>
</html>
```

this page belongs to @dogs

follow submit

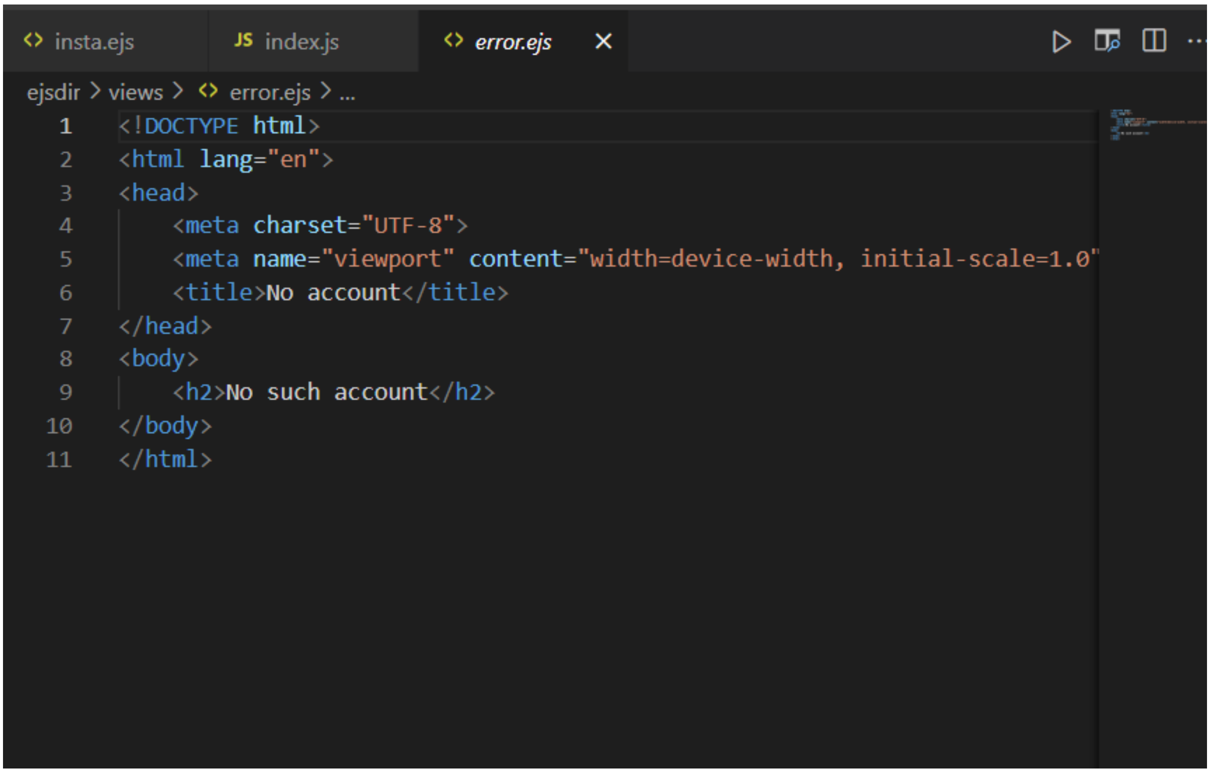


ejsdir > views > insta.ejs > ...

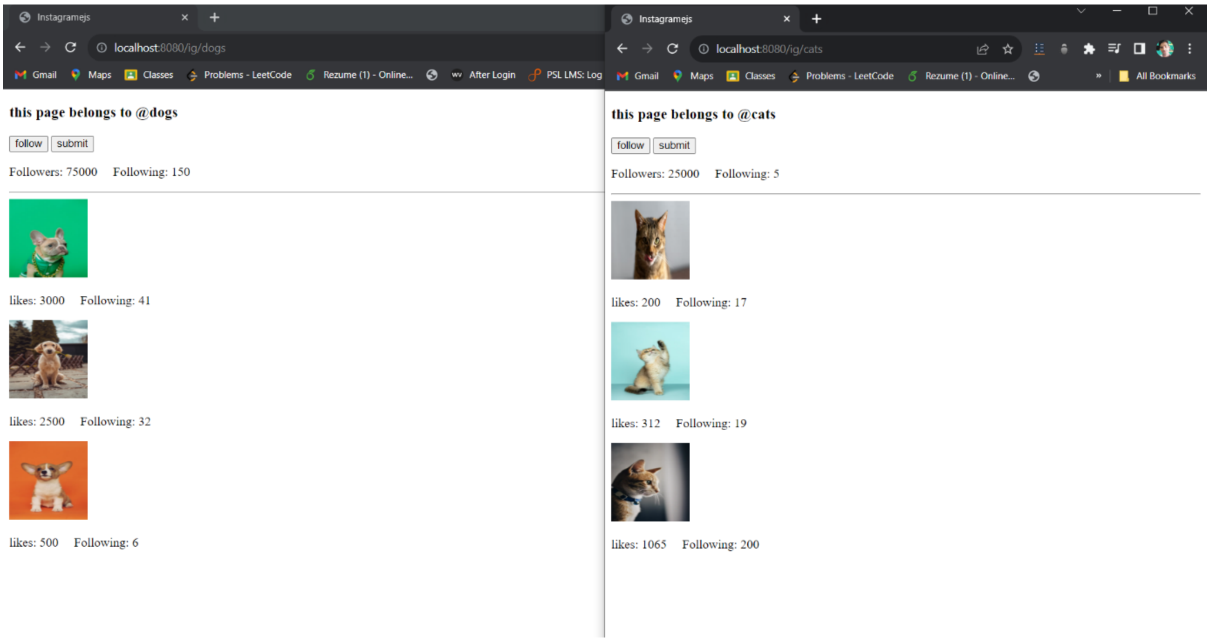
```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initia
    <title>Instagramejs</title>
    <style>
        img{
            height: 100px;
            width: 100px;
        }
    </style>
</head>
<body>
    <h3>this page belongs to @<%=data.name%></h3>
    <button>follow</button>
    <button>submit</button>

    <p>Followers: <%=data.followers%> &nbsp;&nbsp;&nbsp;
    Following: <%=data.following%></p>

    <hr>
   <% for(let post of data.posts){%>
        <img src="<%=post.image%>" alt=" some image">
        <p>likes: <%=post.likes%> &nbsp;&nbsp;&nbsp;
            Following: <%=post.comments%></p>

   <% }%>

</body>
</html>


```

ejsdir > index.js > ...

```js
const express=require("express");
const app=express();
const port=8080;
const path=require("path");

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


// app.set("view engine","ejs");
app.set("views",path.join(__dirname,"/views"));
app.get("/",(req,res)=>{
    res.render("home.ejs");
});

app.get("/cal",(req,res)=>{
    res.render("calculate.ejs");
});

app.get("/dice",(req,res)=>{
    let diceval=Math.floor(Math.random()*6+1);
    res.render("rolldice.ejs",{num:diceval});
});

app.get("/ig/:username",(req,res)=>{
    let {username}=req.params;
    const instadata=require("./data.json");
    const data=instadata[username];
    console.log(data);
    if(data){
        res.render("insta.ejs",{data});
    }else{
        res.render("error.ejs");
    }
});
```



```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0"
    <title>No account</title>
</head>
<body>
    <h2>No such account</h2>
</body>
</html>
```



this page belongs to @dogs

follow submit

Followers: 75000   Following: 150

likes: 3000   Following: 41

likes: 2500   Following: 32

likes: 500   Following: 6

this page belongs to @cats

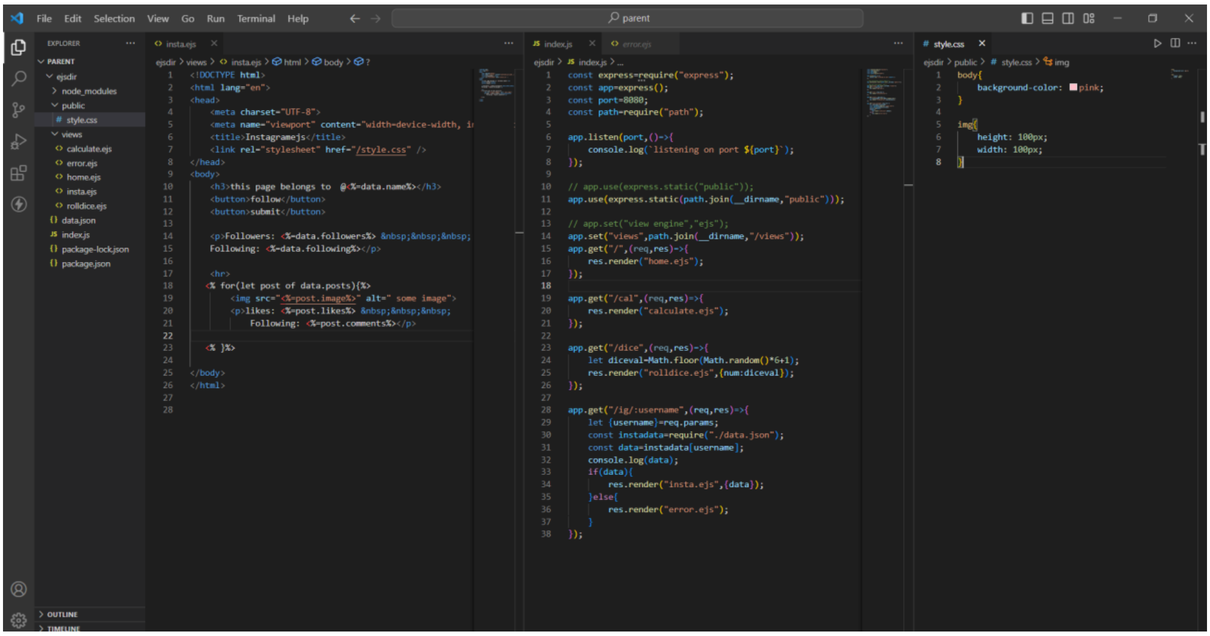
follow submit

Followers: 25000   Following: 5

likes: 200   Following: 17

likes: 312   Following: 19

likes: 1065   Following: 200



ejsdir > views > insta.ejs > html > body > ?

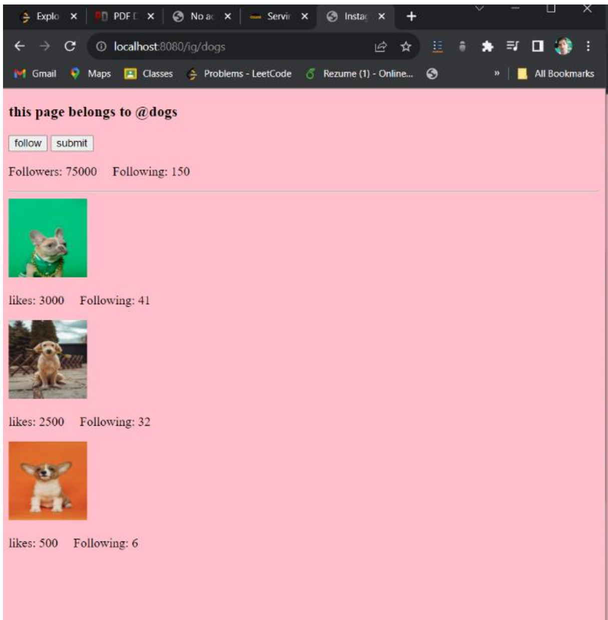
```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, i
    <title>Instagramejs</title>
    <link rel="stylesheet" href="/style.css" />
</head>
<body>
    <h3>this page belongs to  @<%=data.name%></h3>
    <button>follow</button>
    <button>submit</button>

    <p>Followers: <%=data.followers%> &nbsp;&nbsp;&nbsp;
    Following: <%=data.following%></p>

    <hr>
   <% for(let post of data.posts){%>
        <img src="<%=post.image%>" alt=" some image">
        <p>likes: <%=post.likes%> &nbsp;&nbsp;&nbsp;
            Following: <%=post.comments%></p>

   <% }%>

</body>
</html>


```

ejsdir > index.js > ...

```javascript
const express=require("express");
const app=express();
const port=8080;
const path=require("path");

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

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

// app.set("view engine","ejs");
app.set("views",path.join(__dirname,"/views"));
app.get("/",(req,res)=>{
    res.render("home.ejs");
});

app.get("/cal",(req,res)=>{
    res.render("calculate.ejs");
});

app.get("/dice",(req,res)=>{
    let diceval=Math.floor(Math.random()*6+1);
    res.render("rolldice.ejs",{num:diceval});
});

app.get("/ig/:username",(req,res)=>{
    let {username}=req.params;
    const instadata=require("./data.json");
    const data=instadata[username];
    console.log(data);
    if(data){
        res.render("insta.ejs",{data});
    }else{
        res.render("error.ejs");
    }
});
```

ejsdir > public > style.css > img

```css
body{
    background-color: pink;
}

img{
    height: 100px;
    width: 100px;
}
```



this page belongs to @dogs

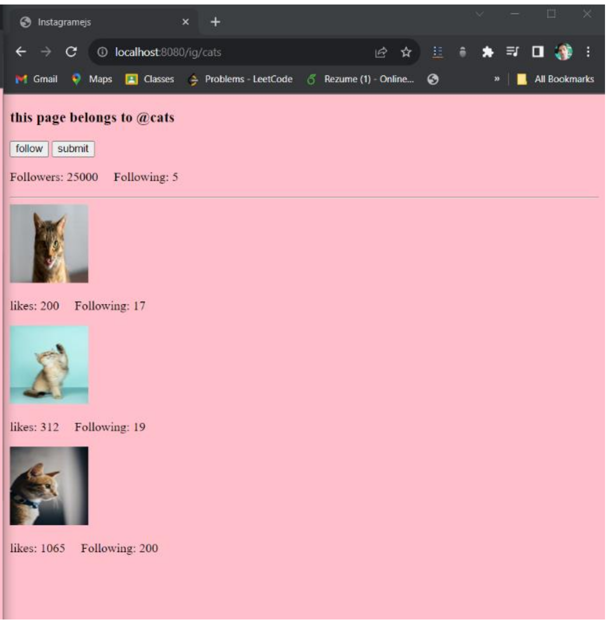
follow submit

Followers: 75000   Following: 150

likes: 3000   Following: 41

likes: 2500   Following: 32

likes: 500   Following: 6



this page belongs to @cats

follow submit

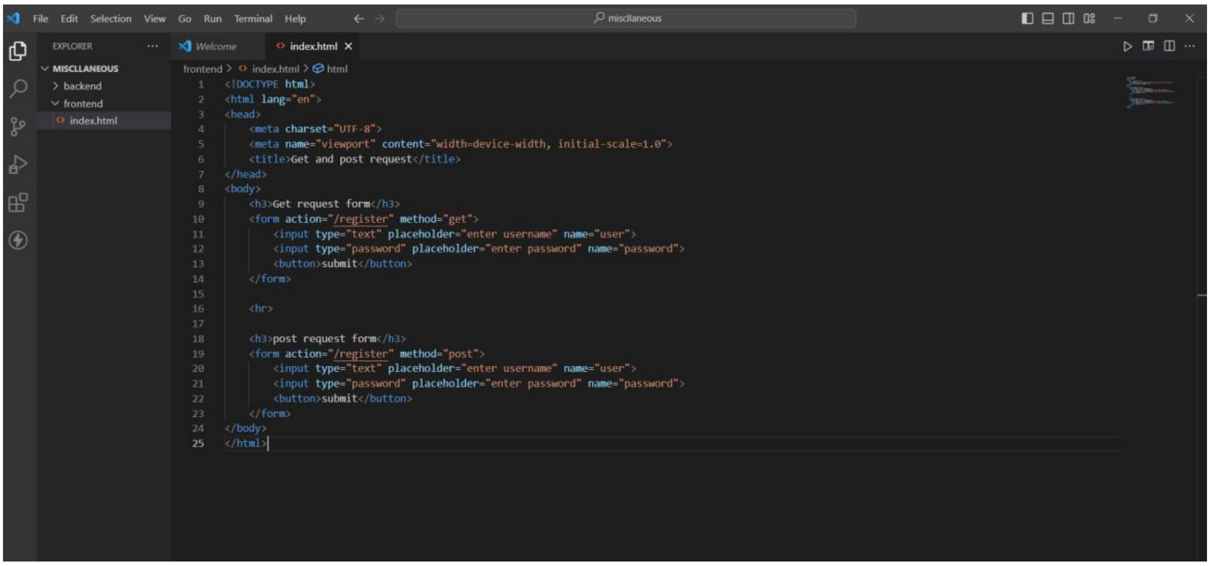
Followers: 25000   Following: 5

likes: 200   Following: 17

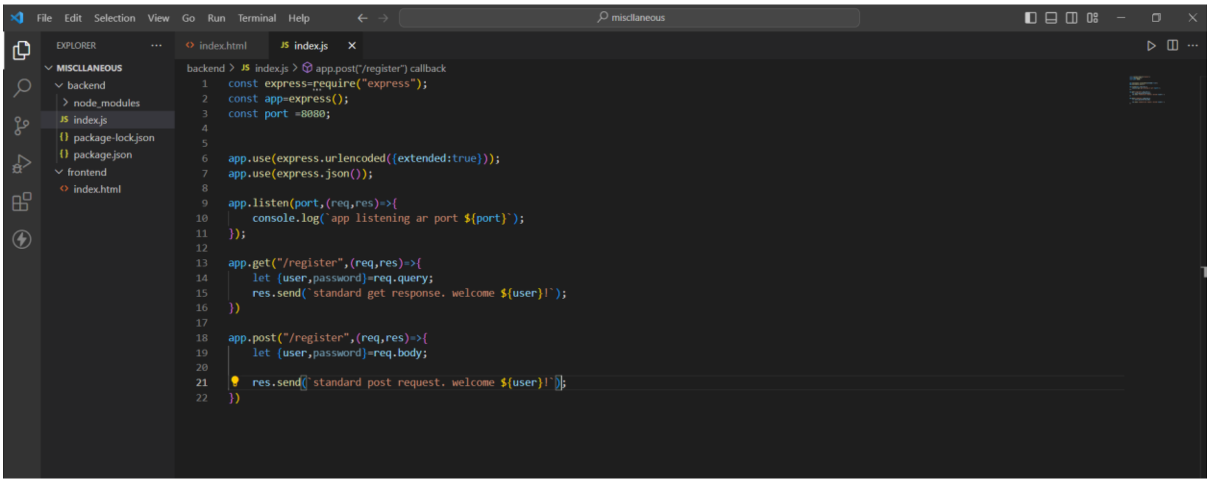
likes: 312   Following: 19

likes: 1065   Following: 200

Get and post request and handling resposnes



```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Get and post request</title>
</head>
<body>
    <h3>Get request form</h3>
    <form action="/register" method="get">
        <input type="text" placeholder="enter username" name="user">
        <input type="password" placeholder="enter password" name="password">
        <button>submit</button>
    </form>

    <hr>

    <h3>post request form</h3>
    <form action="/register" method="post">
        <input type="text" placeholder="enter username" name="user">
        <input type="password" placeholder="enter password" name="password">
        <button>submit</button>
    </form>
</body>
</html>
```



```javascript
const express=require("express");
const app=express();
const port =8080;


app.use(express.urlencoded({extended:true}));
app.use(express.json());

app.listen(port,(req,res)=>{
    console.log(`app listening ar port ${port}`);
});

app.get("/register",(req,res)=>{
    let {user,password}=req.query;
    res.send(`standard get response. welcome ${user}!`);
})

app.post("/register",(req,res)=>{
    let {user,password}=req.body;

    res.send(`standard post request. welcome ${user}!`);
})
```

- Run index.js
- Then run index.html and see difference in handling responses of get and post request