* ***ALL ABOUT NODE JS***
* **What is node js?**
* Node is a asnyc language.
* It is a single threaded because it runs one command at a time.
* Node is not a language.
* It is a server environment.
* It connects with database.
* The code and syntax of node are not exactly same but similar to java script.
* Node is free and is open source.
* **Why we use node ?**
* To use APIs.
* It can connect the same database with webapp, mobile app.
* To become a full stack developer we use node.
* **History of node js:**
* The first release of node js was on May 27 2009
* The current version is 16.4.
* Node is written in C, C++, Javascript.
* **Difference between node and js**

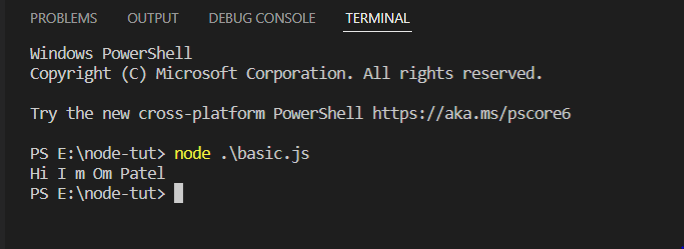
|  |  |
| --- | --- |
| **Node** | **JS** |
| Code syntax is similar. | Code syntax is same. |
| Node can connect to database. | JS cannot connect to database. |
| Node can run on server side. | JS can run on browser side. |

* **Basic works:**

**Code:**

console.log("Hi I m Om Patel")

**Output:**



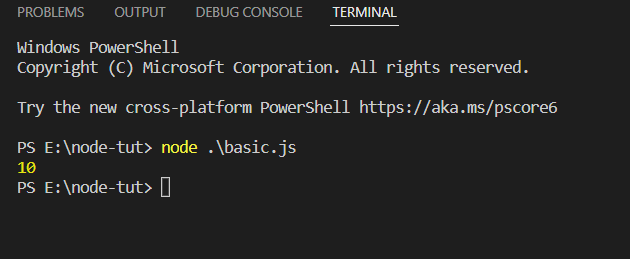
**Code:**

const a=4;

const b=6;

console.log(a+b);

**Output:**



**Code:**

let x=20

const y=10

console.log(x+y);

if(x+y===30)

{

    console.log("Value is matched");

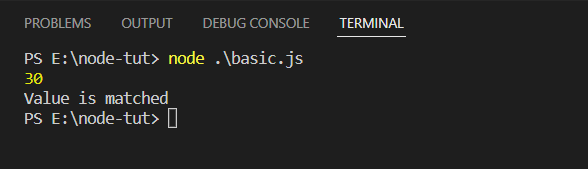
}else

{

    console.log("Value is not matched");

}

**Output:**



* **Basic operation and is run by nodemon:**

**Code:**

let x=20

const y=10

console.log(x+y);

if(x+y===30)

{

    console.log("Value is matched");

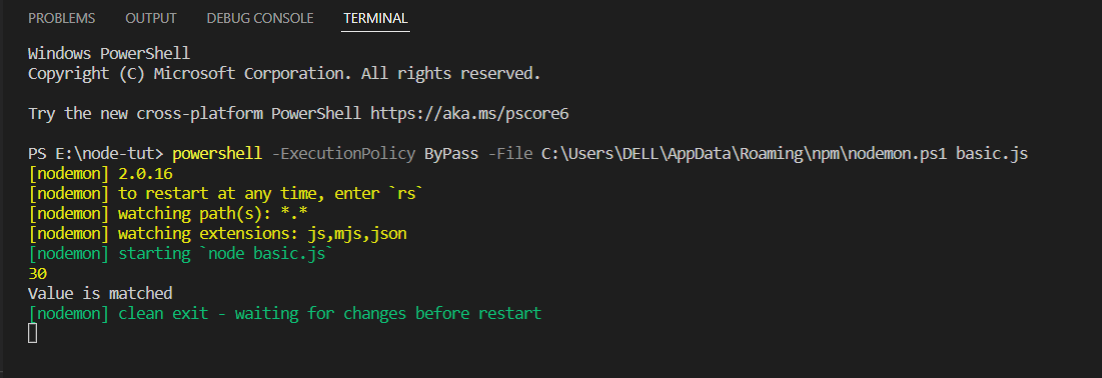
}else

{

    console.log("Value is not matched");

}

**Output:**



* **Make basic server**

**Code:**

const http = require('http');

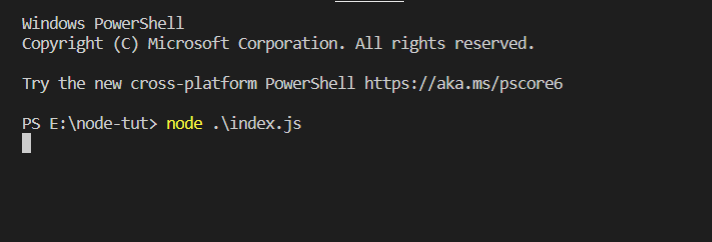
http.createServer((req,resp)=>{

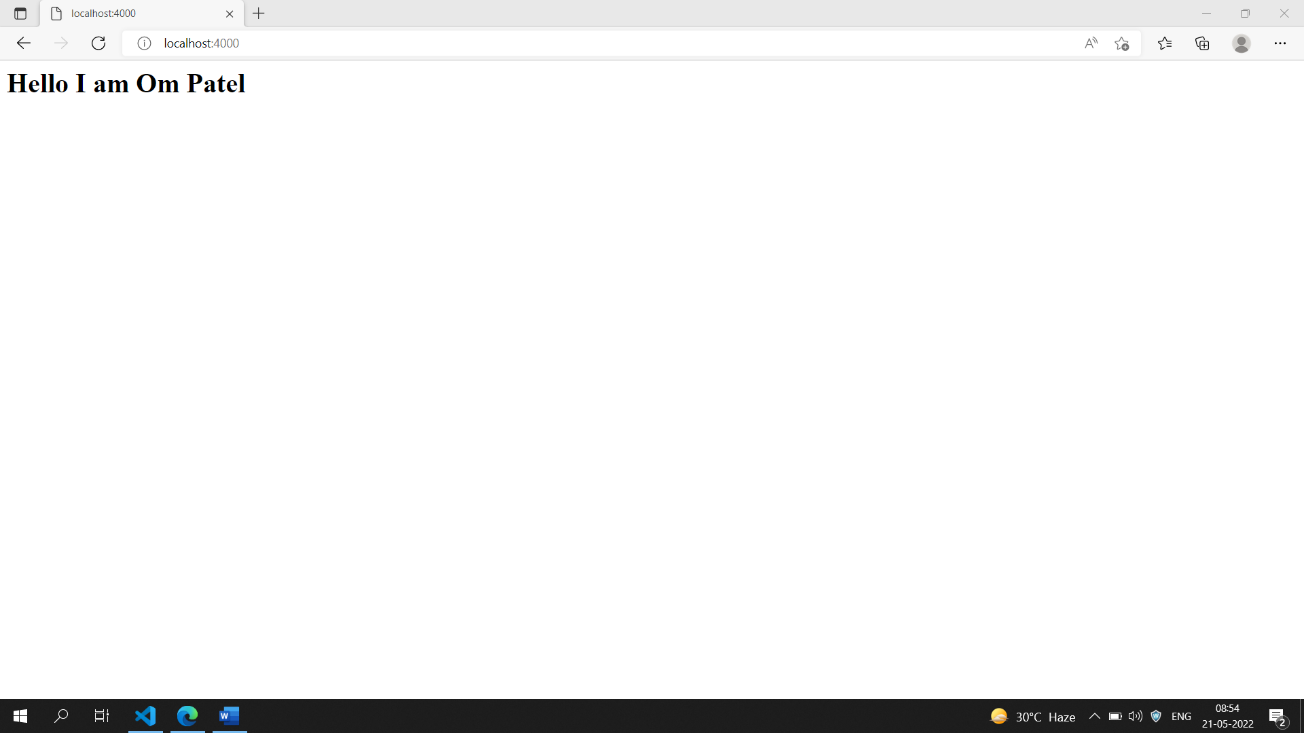
    resp.write("<h1>Hello I am Om Patel</h1>");

    resp.end();

}).listen(4000);

**Output:**





* **Server using function**

**Code:**

const http = require('http');

function dataControl(req,resp)

{

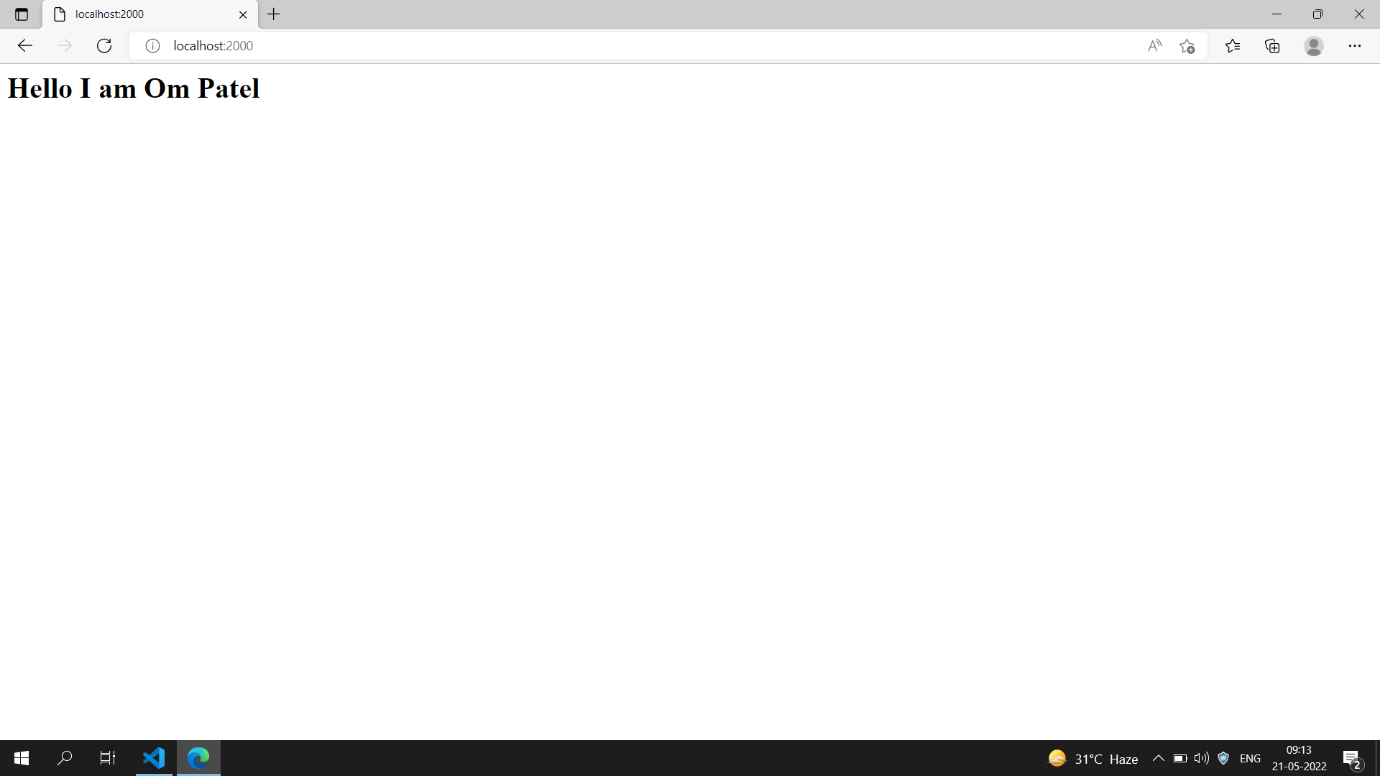
    resp.write("<h1>Hello I am Om Patel</h1>");

    resp.end();

}

http.createServer(dataControl).listen(2000);

**Output:**



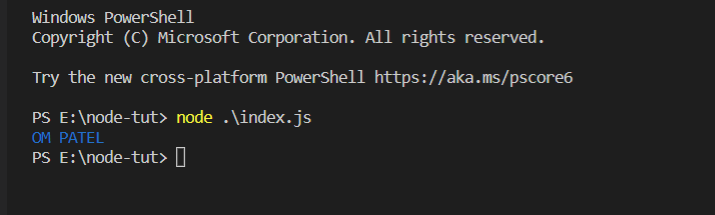
* **Installation of json packages**
* Installation of colors
* Installation of simple node logger
* Colourfull text using code

**Code:**

const colors= require('colors');

console.log("OM PATEL".blue)

**Output:**



* **How to backup if the node modules is deleted?**

-> Simply write npm install and all your packages will be backed up.

* **Creating header and API body;**

**Code:**

const http = require('http');

http.createServer((req,resp)=>{

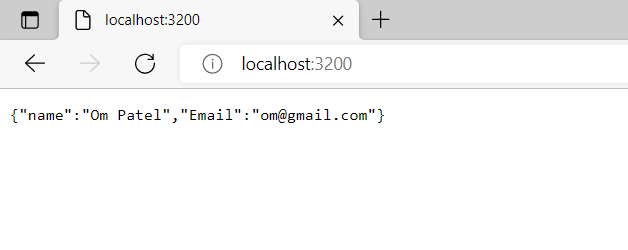
resp.writeHead(200,{'Content-Type':'application\json'});

resp.write(JSON.stringify({ name:'Om Patel', Email: 'om@gmail.com'}));

resp.end()

}).listen(3200);

**Output:**



* **Creating API with static data and put data in another file.**

**Code:**

(basic.js)

const http = require('http');

const data = require('./data');

http.createServer((req,resp)=>{

resp.writeHead(200,{'Content-Type':'application\json'});

resp.write(JSON.stringify(data));

resp.end();rmSync

}).listen(3500);

(data.js)

const data=[

    {name:'Om Patel', Email: 'om@gmail.com'},

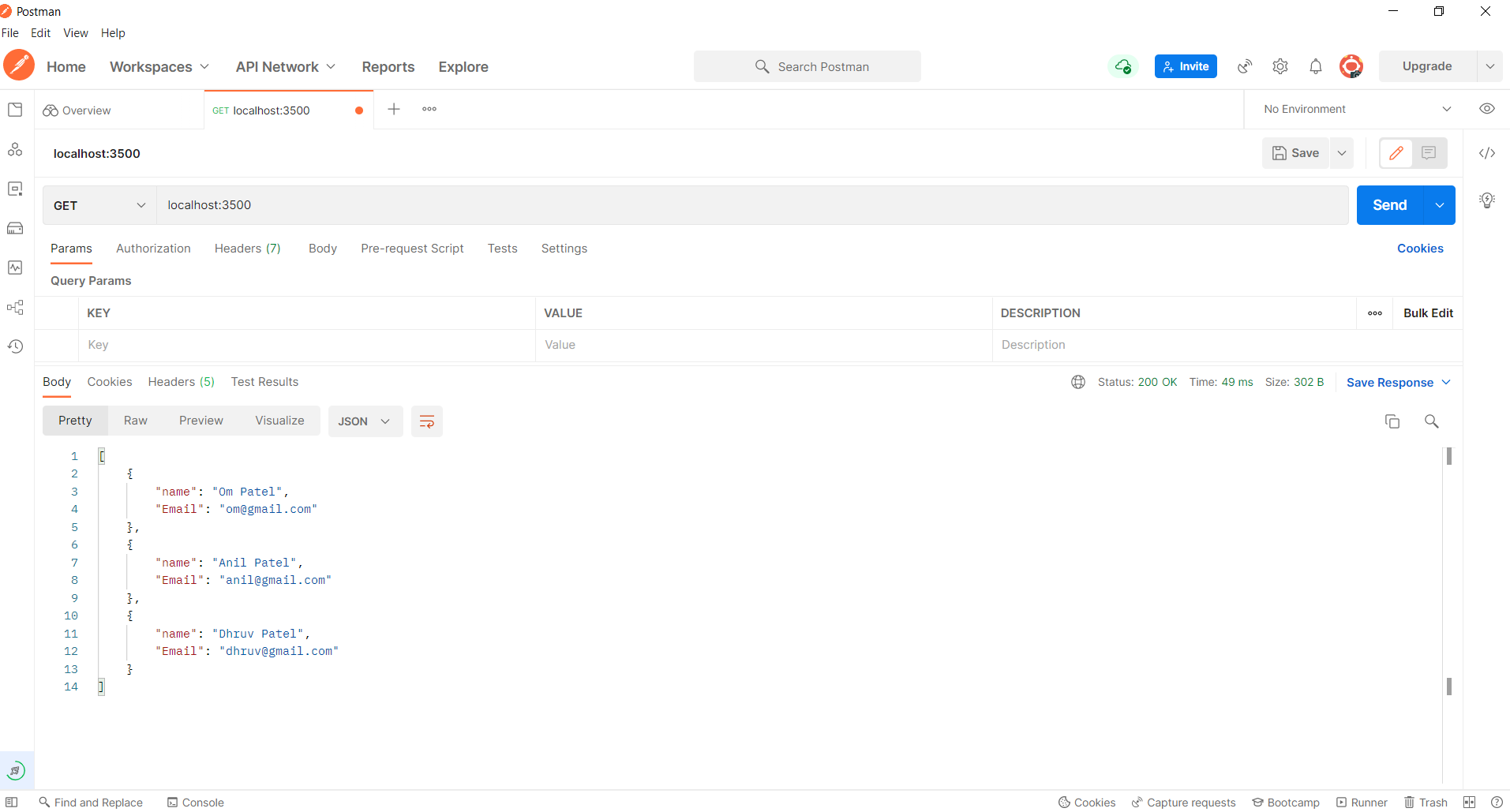
    {name:'Anil Patel', Email: 'anil@gmail.com'},

    {name:'Dhruv Patel', Email: 'dhruv@gmail.com'},

]

module.exports=data;

**Output:**



* **To create file:**

const fs = require('fs');

const input = process.argv;

fs.writeFileSync(input[2],input[3])

* **To add and remove file:**

const fs = require('fs');

const input = process.argv;

if(input[2]=='add')

{

    fs.writeFileSync(input[3],input[4])

}else if(input[2]=='remove')

{

    fs.unlinkSync(input[3])

}

else{

    console.log("This is invalid input")

}

* **To know path:**

const fs =require('fs');

const path = require('path');

const dirPath=path.join(\_\_dirname,'files');

console.warn(dirPath)

* **To create multiple files at a time:**

const fs =require('fs');

const path = require('path');

const dirPath=path.join(\_\_dirname,'files');

for(i=0;i<=5;i++)

{

    fs.writeFileSync(dirPath+"/Tata"+i+".txt", "This is a company");

}

* **Show file list:**
* **To read and list files:**
* **Get file names and list:**

**~ To list all files:**

const fs =require('fs');

const path = require('path');

const dirPath=path.join(\_\_dirname,'files');

fs.readdir(dirPath,(err,files)=>{

    console.log(files)

})

* **CRUD System with file system**

C- Create

R- Rename

U- Update

D- Delete

* **Asynchronus Programming language:**

**EXAMPLE:-**

let a=10;

const b=20;

setTimeout(()=>{

    b=20;

},2000)

console.log(a+b)

* **TO handle data**
* **To wait and print data after some specific time period:**

let a=10;

let b=20;

let waitingData =new Promise((resolve,reject)=>{

setTimeout(()=>{

        resolve(30)

    },2000)

})

waitingData.then((data)=>{

    b=data;

console.log(a+b)

})

* **How node js works?**

1. **Call stack**
2. **Node APIs**
3. **Callback queue**

**EXAMPLE:-**

console.log("Starting up")

setTimeout(()=>{

console.log('2 second log')

},2000)

setTimeout(()=>{

console.log('0 second log')

},0)

console.log("Finishing up")

* **What is Express js?**
* **Installation**
* **Creating 1 or more webpages using express:**

**Code:**

const express = require('express');

const res = require('express');

const app = express();

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

    res.send('Hello this is our first page');

});

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

    res.send('Hello this is our about page');

});

app.listen(7000);

* **Render/Display HTML and JSON:**

**Code:**

const express = require('express');

const res = require('express');

const app = express();

app.get('', (req, resp) => {

    resp.send('Hello this is our first page');

});

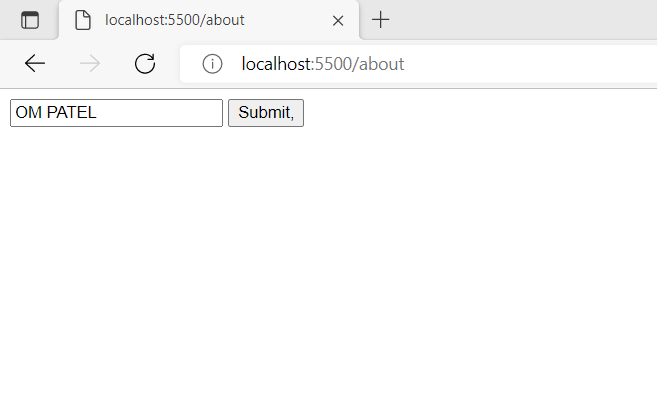
app.get("/about", (req, resp) => {

    resp.send('<input type="text", placeholder="Enter your name" /> <button> Submit</button>')

});

app.listen(5500);

**Output:**



* **To render JSON data:**

**Code:**

const express = require('express');

const res = require('express');

const app = express();

app.get('', (req, resp) => {

    resp.send('Hello this is our first page');

});

app.get("/about", (req, resp) => {

    resp.send('<input type="text", placeholder="Enter your name" /> <button> Submit</button>')

});

app.get("/help", (req, resp) => {

    resp.send({

        NAME: 'Om Patel',

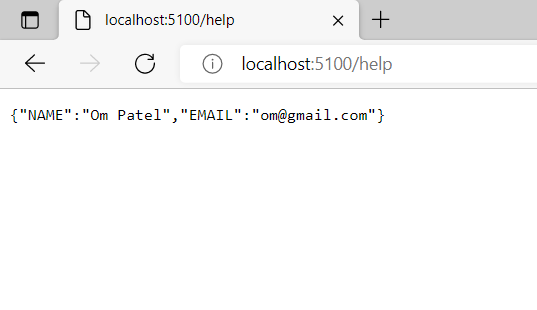
        EMAIL: 'om@gmail.com'

    });

});

app.listen(5100);

**Output:**



* **To go from one page to another:**

**Code:**

const express = require('express');

const res = require('express');

const app = express();

app.get("", (req, resp) => {

    resp.send('<h1>Hello this is our first page</h1><a href="/about" > Go to about page</a> ');

});

app.get("/about", (req, resp) => {

    resp.send('<input type="text", placeholder="Enter your name" /> <button> Submit</button>  <a href="/" > Go to home page</a>');

});

app.get("/help", (req, resp) => {

    resp.send({

        NAME: 'Om Patel',

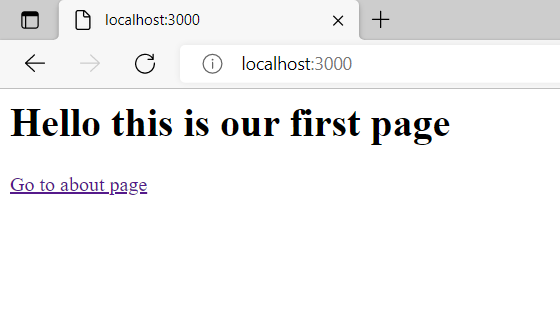
        EMAIL: 'om@gmail.com'

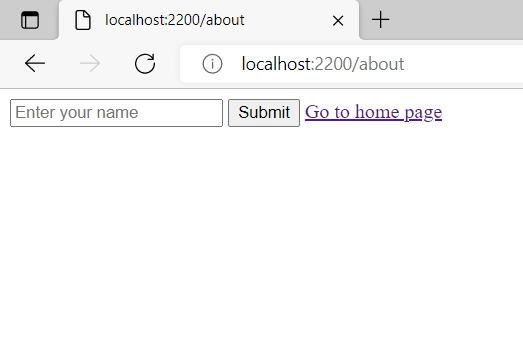
    });

});

app.listen(3000);

**Output:**





* **Make folder for HTML files and access it:**

**Code:**

**(index.js)**

const express = require('express');

const path = require('path');

const app = express();

const publicPath= path.join(\_\_dirname,'public');

app.use(express.static(publicPath));

app.listen(4000);

**(about.html)**

<!DOCTYPE html>

<html>

    <head>

        <title>About page</title>

    </head>

    <body><h1>This is about page</h1>

    </body>

</html>

* **Make HTML files**
* **Load HTML files**

**Code:**

**(index.js)**

const express = require('express');

const path=require('path');

const app = express();

const publicPath=path.join(\_\_dirname,'public');

app.use(express.static(publicPath));

app.listen(7000);

**(about.html)**

    <!DOCTYPE html>

    <html>

        <head>

            <title>About page</title>

        </head>

        <body><h1>This is about page</h1>

        </body>

    </html>

**(help.html)**

    <!DOCTYPE html>

    <html>

        <head>

            <title>Help page</title>

        </head>

        <body><h1>This is Help page</h1>

        </body>

    </html>

**(index.html)**

<!DOCTYPE html>

<html>

    <head>

        <title>Home page</title>

    </head>

    <body><h1>This is home page</h1>

    </body>

</html>

* **Middleware**

**~ How to make middleware?**

**~ Apply middleware on routes**

**Code:**

const express = require('express');

const app = express();

const reqFilter = (req, resp, next) => {

    if (!req.query.age) {

        resp.send("Please provide age")

    }

    else if (req.query.age < 18) {

        resp.send("You cannot access this page")

    }

    else {

        next();

    }

}

app.use(reqFilter)

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

    resp.send('Welcome to home page')

})

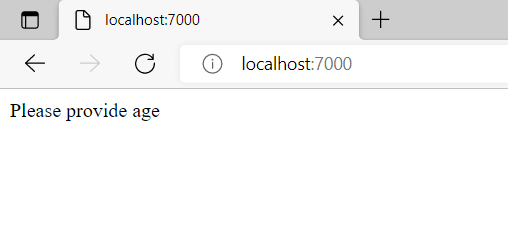
app.get('/users', (req, resp) => {

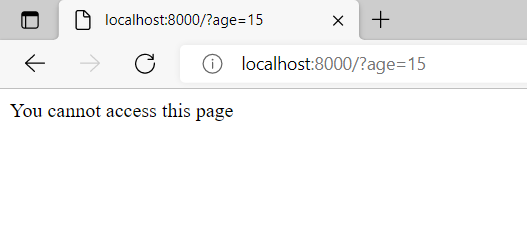
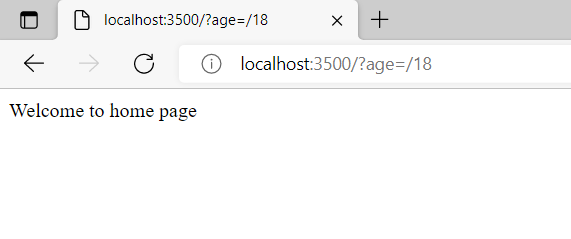
    resp.send('Welcome to users page')

})

app.listen(7000);

**Output:**





* **Types of middleware:**

**Single route level middleware**

const express = require('express');

const app = express();

const reqFilter = (req, resp, next) => {

    if (!req.query.age) {

        resp.send("Please provide age")

    }

    else if (req.query.age < 18) {

        resp.send("You cannot access this page")

    }

    else {

        next();

    }

}

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

    resp.send('Welcome to home page')

})

app.get('/users', reqFilter, (req, resp) => {

    resp.send('Welcome to users page')

})

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

    resp.send('Welcome to about page')

})

app.listen(8500);