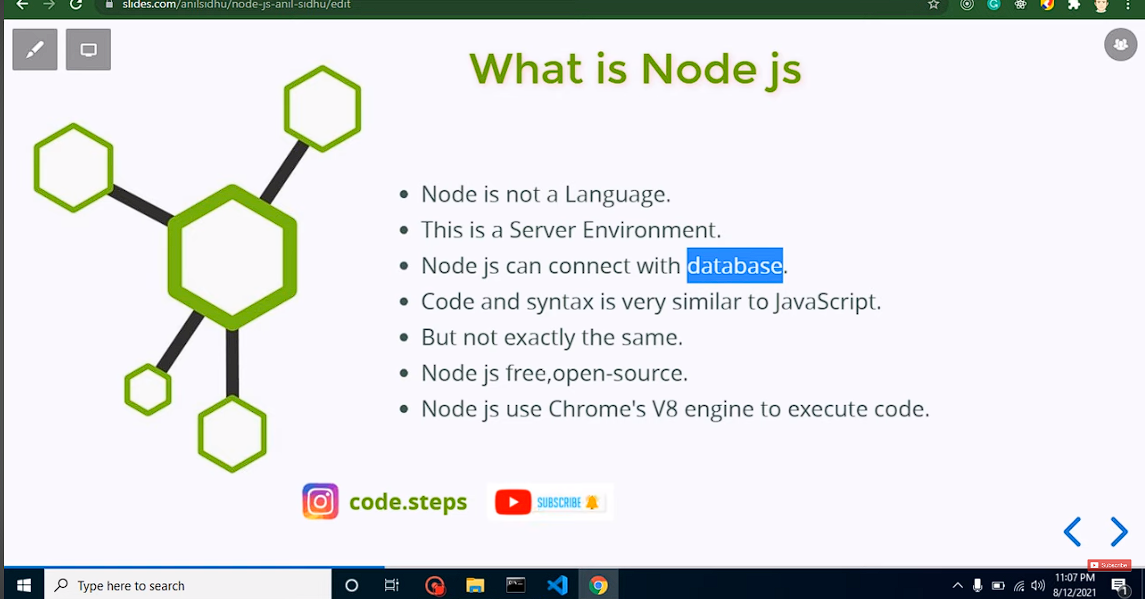
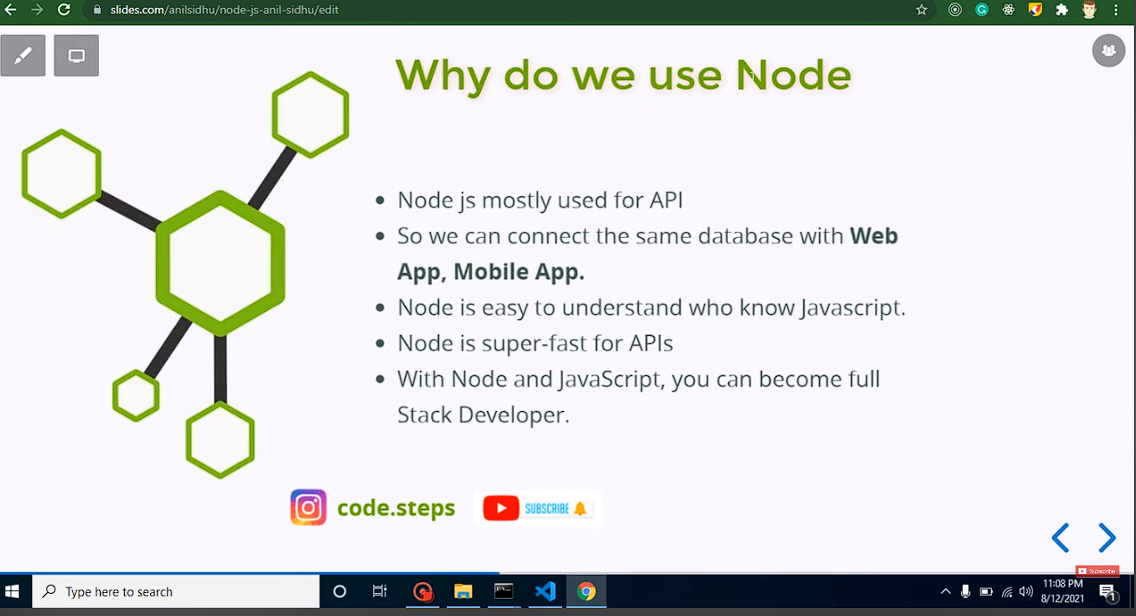
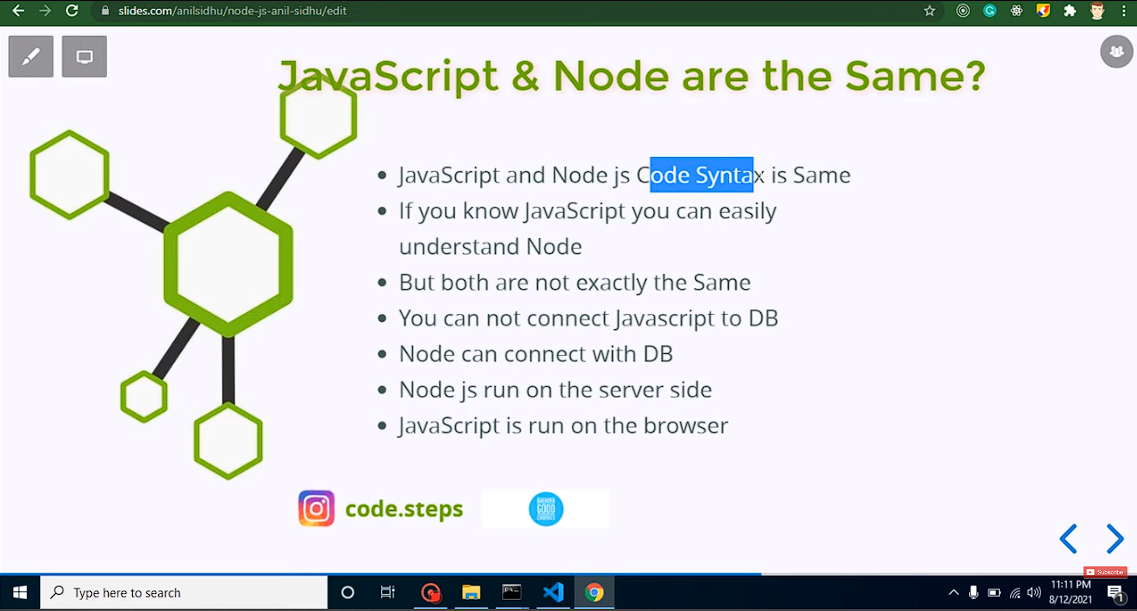
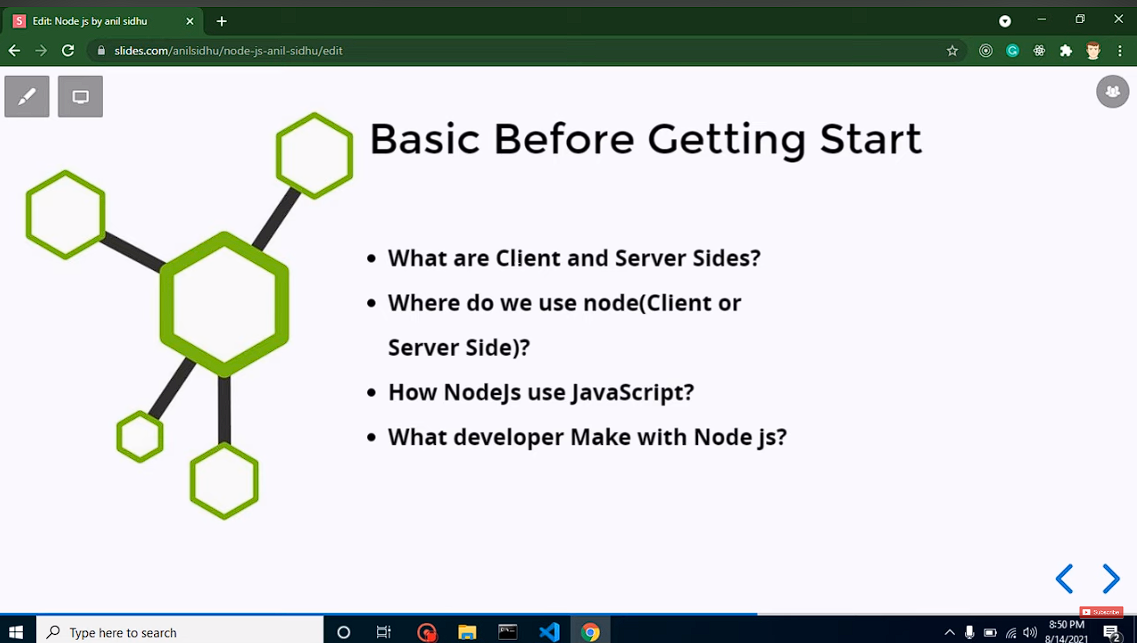
**Day 1:**

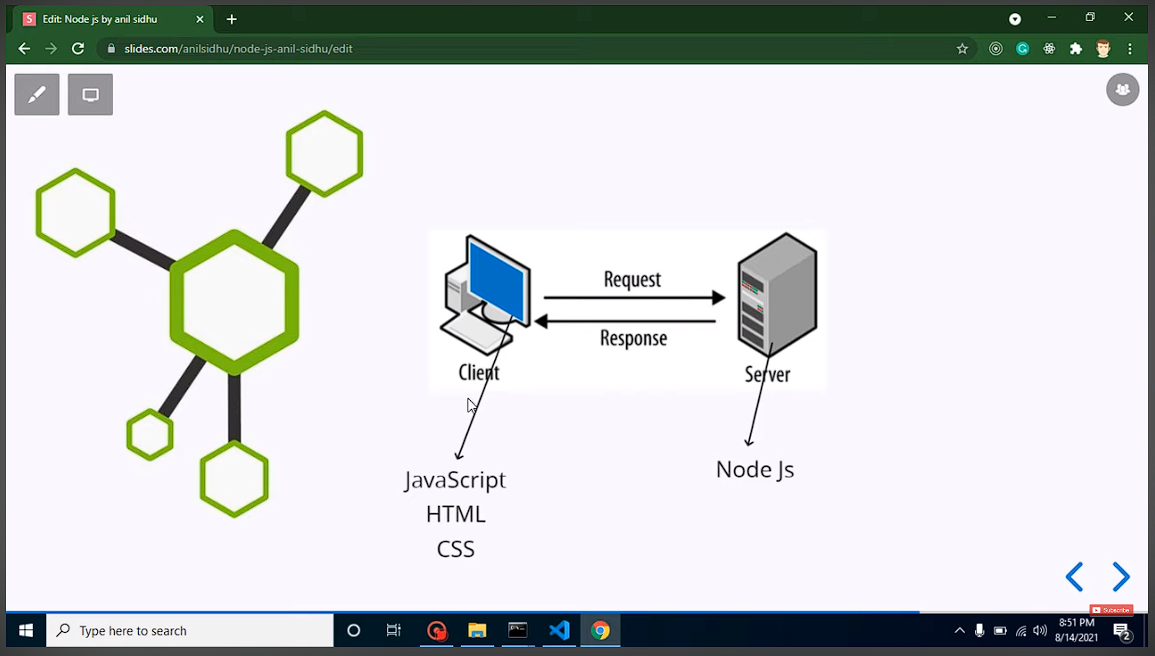


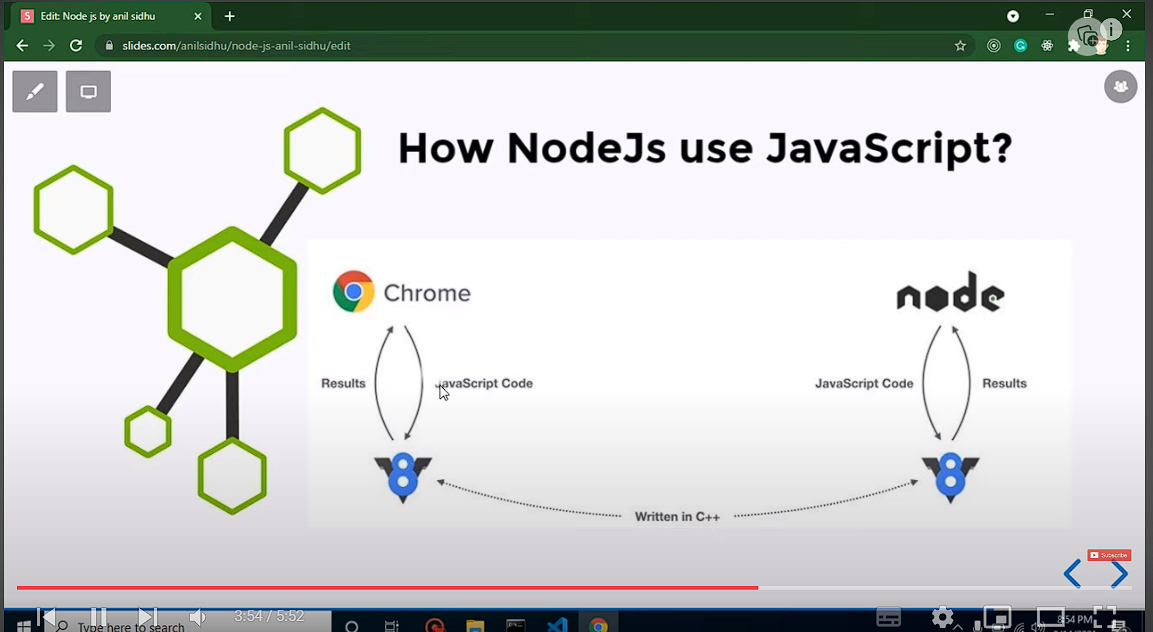




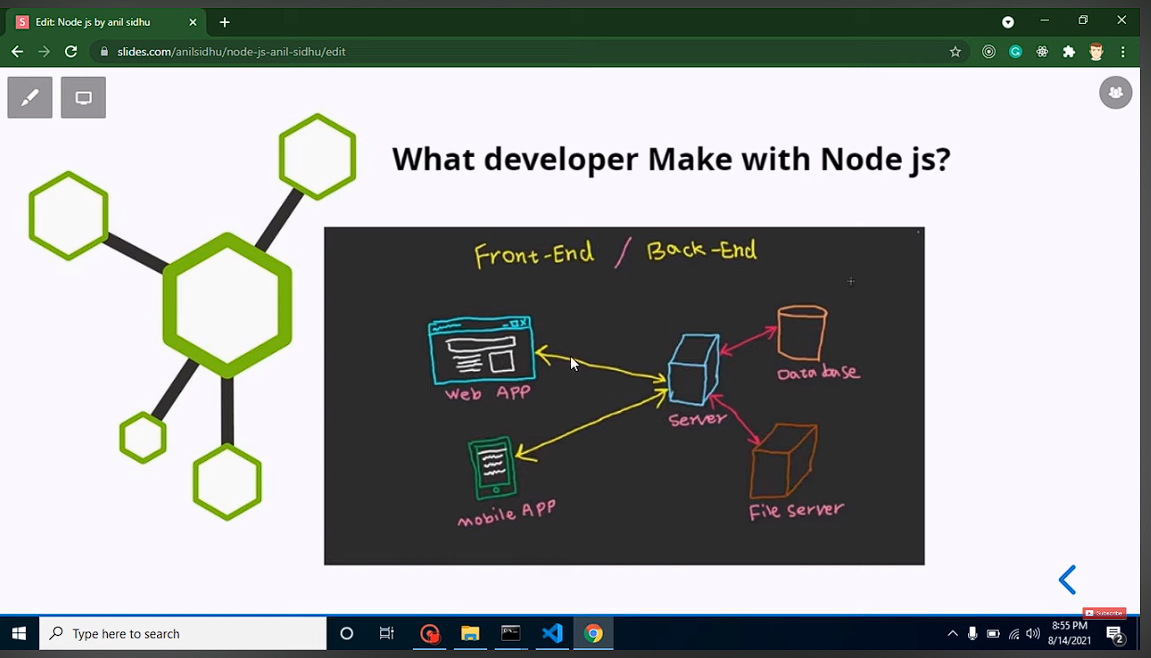


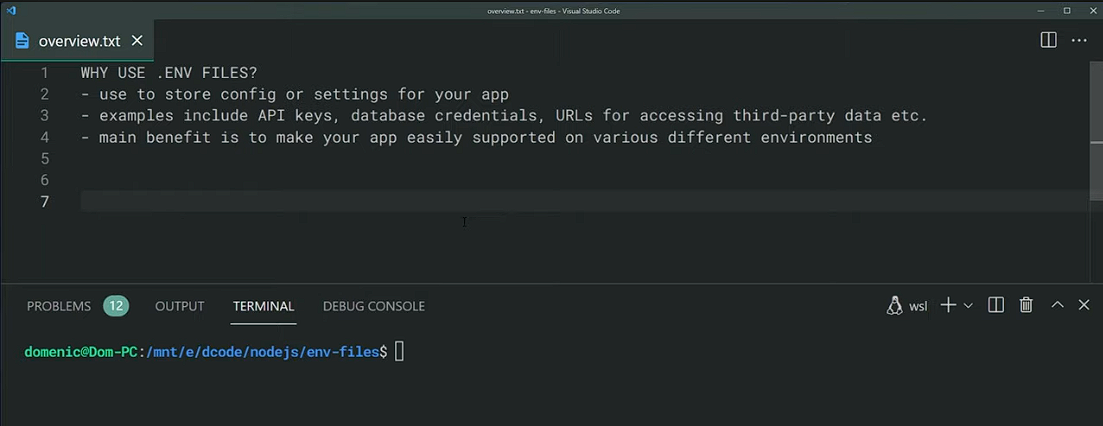












require('dotenv').config()

console.log(process.env.PORT)

A screenshot of a computer screen

Description automatically generated

A computer screen shot of a black screen

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer screen

Description automatically generated

**Q1. Node js is single thread or multi thread ?**

Node.js was created explicitly as an experiment in async processing. The theory was that doing async processing on a single thread could provide more performance and scalability under typical web loads than the typical thread-based implementation.

And you know what? In my opinion that theory's been borne out. A node.js app that isn't doing CPU intensive stuff can run thousands more concurrent connections than Apache or IIS or other thread-based servers.

The single threaded, async nature does make things complicated. But do you honestly think it's more complicated than threading? One race condition can ruin your entire month! Or empty out your thread pool due to some setting somewhere and watch your response time slow to a crawl! Not to mention deadlocks, priority inversions, and all the other gyrations that go with multithreading.

In the end, I don't think it's universally better or worse; it's different, and sometimes it's better and sometimes it's not. Use the right tool for the job.

The main event loop in NodeJS is single threaded but most of the I/O works run on separate threads. You can make it multi-threaded by creating child processes. There is a npm module napajs to create a multi-threaded JavaScript runtime.

However, the 10.5.0 release has announced multithreading in Node.js. The feature is still experimental and likely to undergo extensive changes, but it does show the direction in which Node.js is heading.

So, stay tuned!!

Q2. Nodejs is an async or sync??

Nodejs is an async server environment.

Async:

1. Send request.
2. go on with other code.
3. response come in any time on a callback.

Sync:

1. Send request.
2. Wait for response.
3. go on with other code after response.

**Day 2:**

* Http Status codes: [HTTP response status codes - HTTP | MDN (mozilla.org)](https://developer.mozilla.org/en-US/docs/Web/HTTP/Status)
* Architecture of node js: [The A-Z of Node.js Architecture - A Full Overview (simplilearn.com)](https://www.simplilearn.com/understanding-node-js-architecture-article)

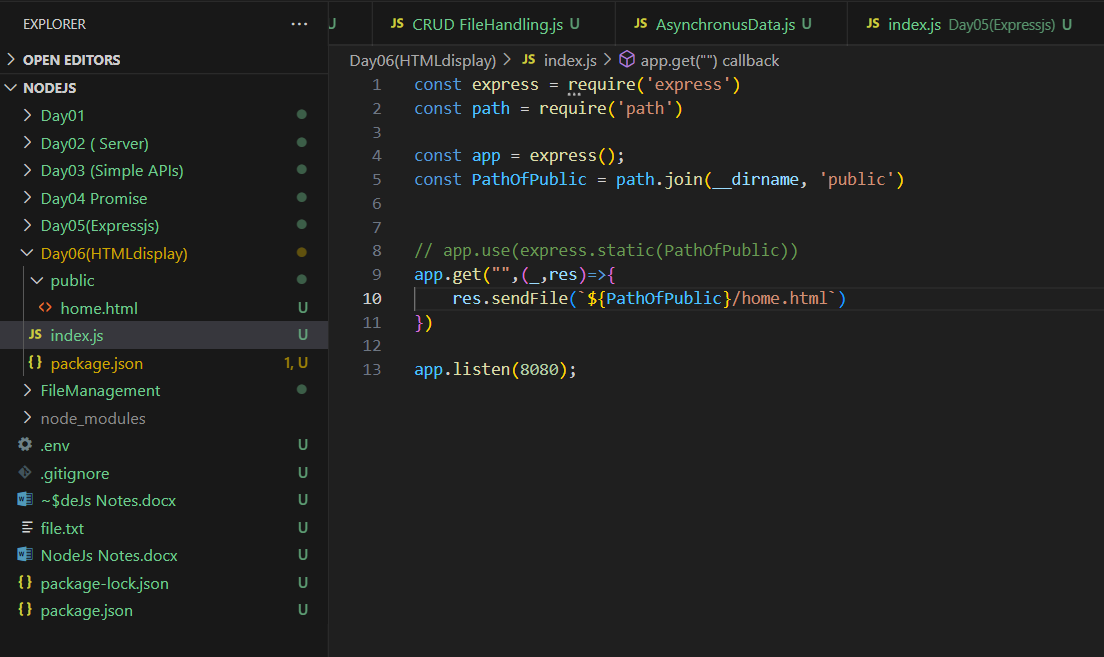
`A screenshot of a computer

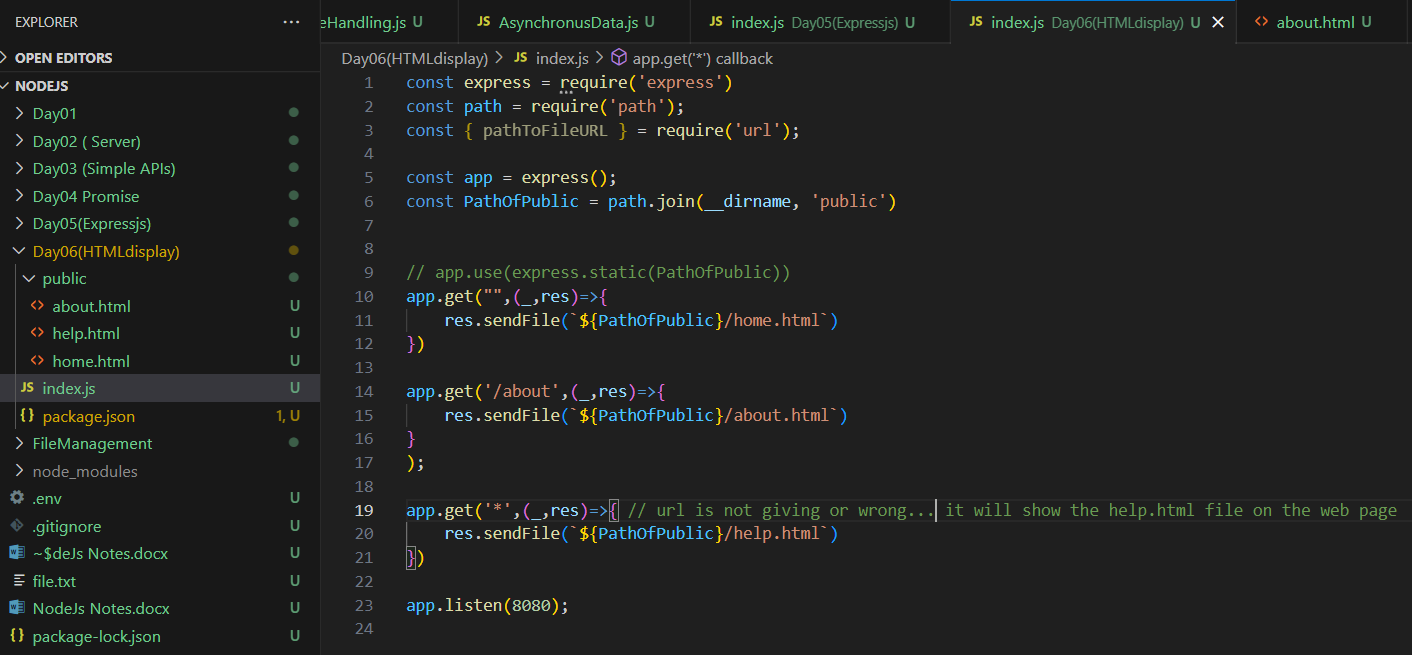
Description automatically generated

Q. What are the main features of Path module?

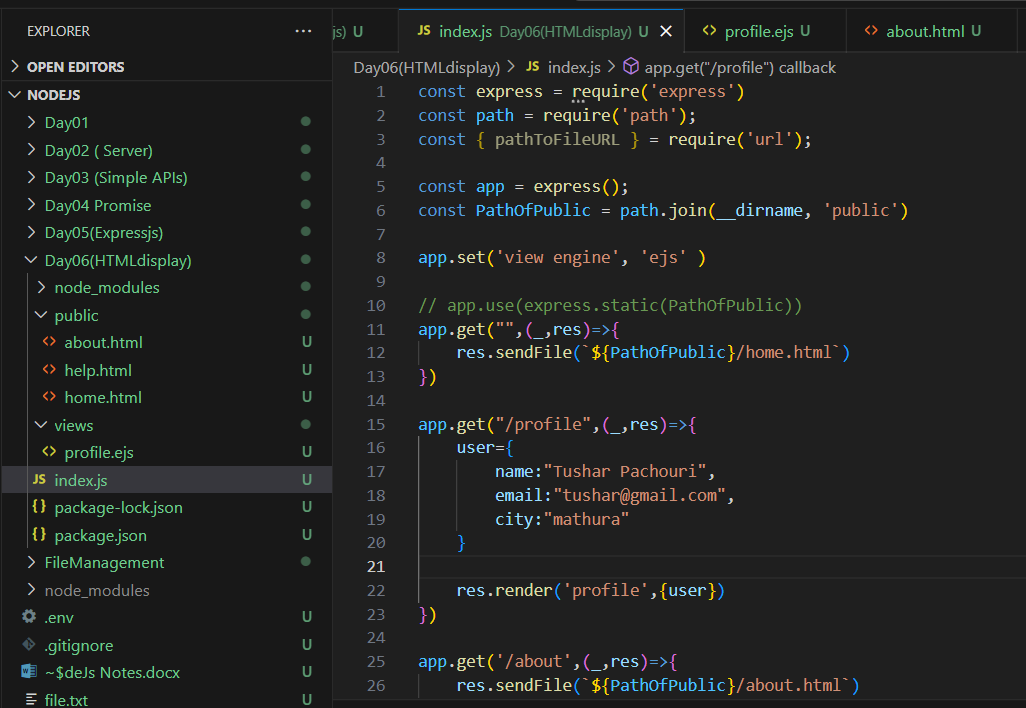
🡺 <https://www.w3schools.com/nodejs/ref_path.asp>

How to add html file or load html file from get method …. Use of path module…





Points to remember while using ejs: first we have to set the view engine as ejs and we can use it to create dynamic website which can take data from the database and show us on the website. Second point to remember is that we must create a folder as ‘views’ always.



If we have to use the data in our ejs file, we can use the following method:

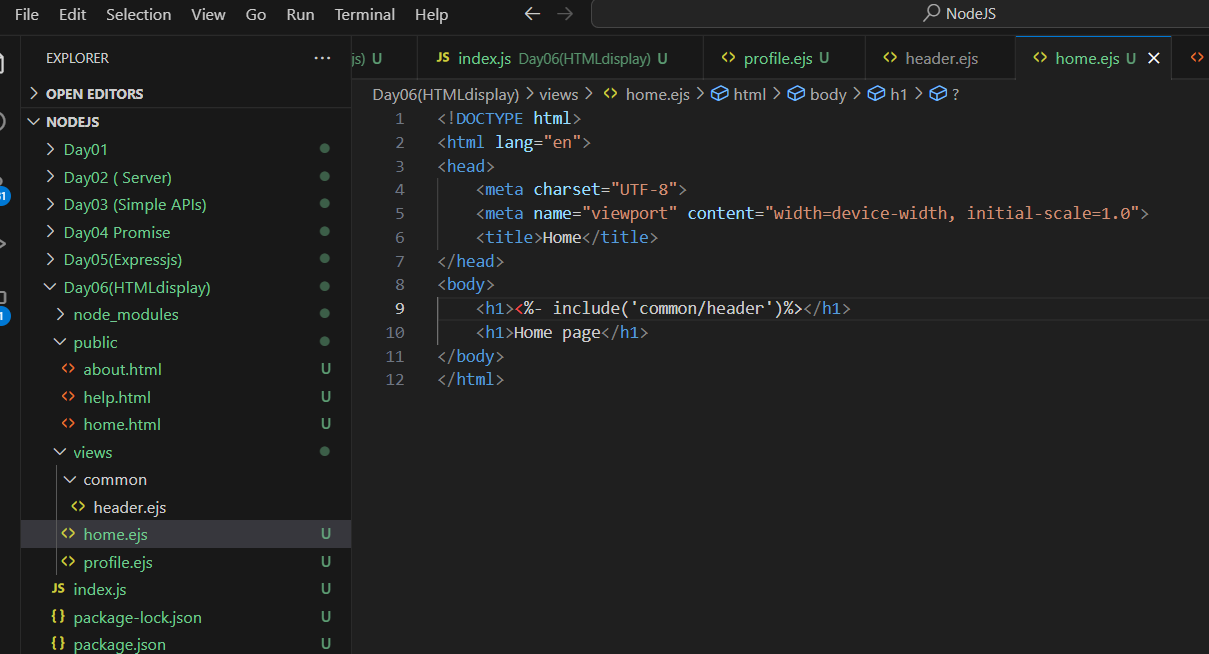
That is <%= DynamicData %>

A screen shot of a computer program

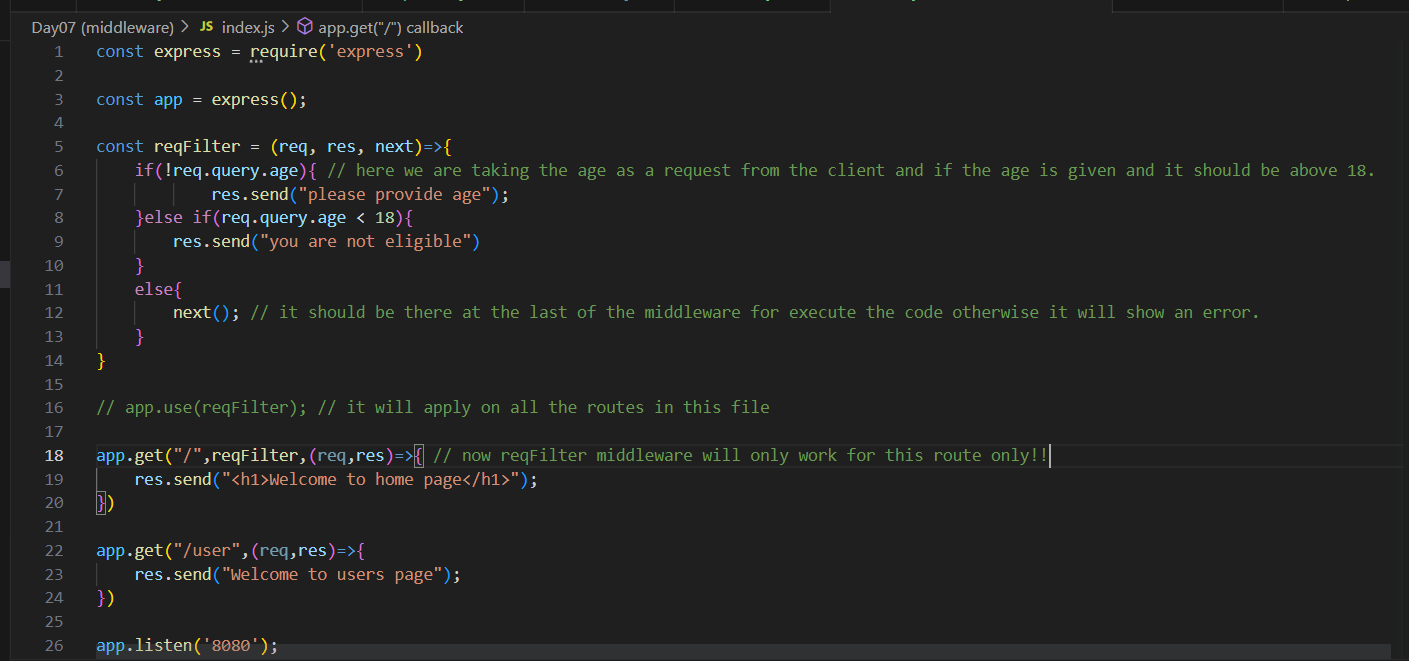
Description automatically generated

How to use ejs in nodejs : [How to use Ejs in JavaScript ? - GeeksforGeeks](https://www.geeksforgeeks.org/how-to-use-ejs-in-javascript/)

**How to add ejs file to ejs file**: Here we are adding header to home file which is stored in “common” folder.



Middleware in the same file:



Mongodb DataBase:



Can we connect two databases at the same time? YES