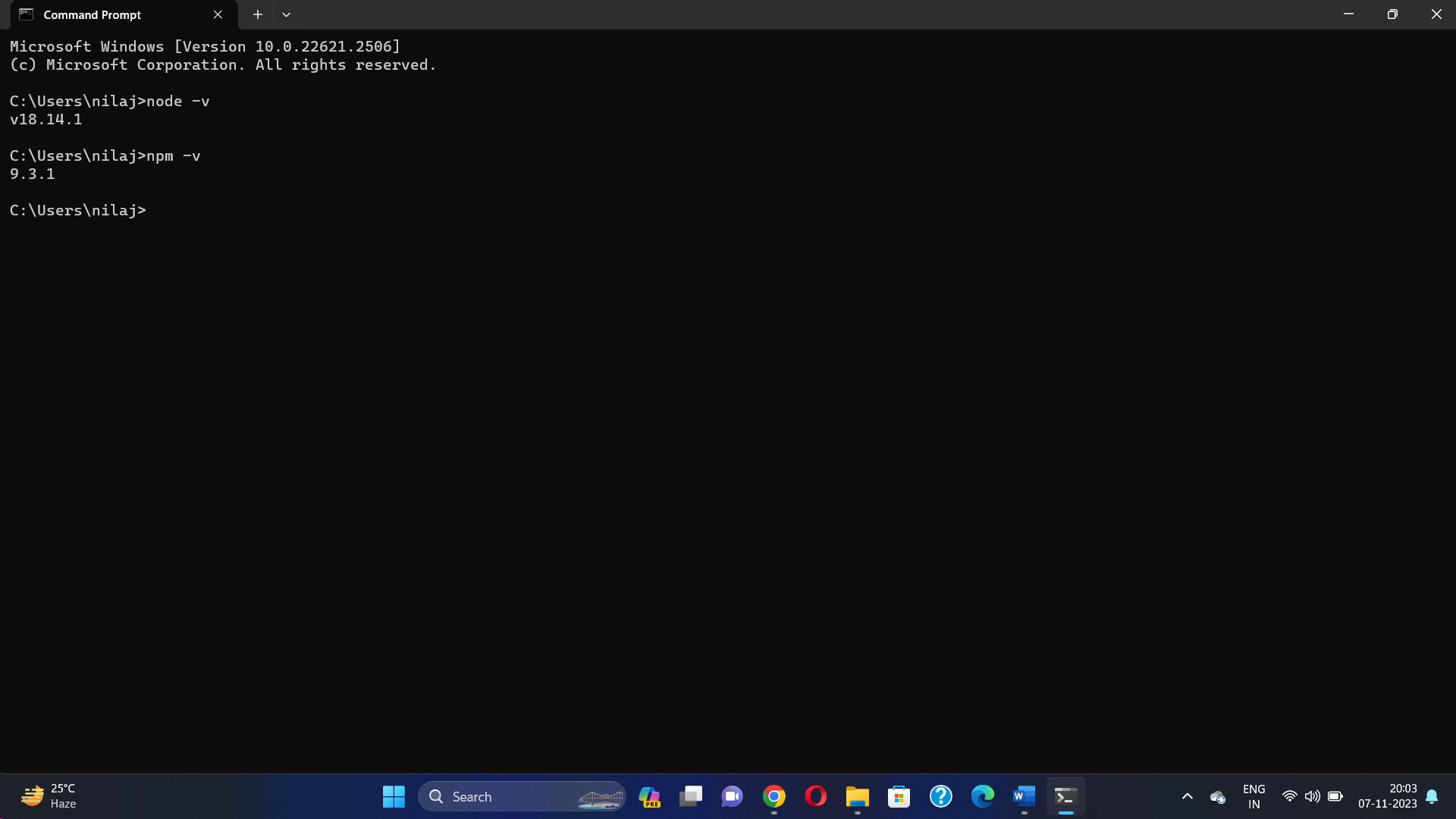


Install Node from site

Version Check node and npm



Node.js is an open-source and cross-platform JavaScript runtime environment.

It is a popular tool for almost any kind of project!

Node.js runs the V8 JavaScript engine, the core of Google Chrome, outside of the browser.

This allows Node.js to be very performant.

why we use node.js

Node.js uses asynchronous programming!

Node.js handles a file request:

Sends the task to the computer's file system.

Ready to handle the next request.

When the file system has opened and read the file, the server returns the content to the client.

What is Node.js?

------------------

Node.js is an open source server environment

Node.js is free

Node.js runs on various platforms (Windows, Linux, Unix, Mac OS X, etc.)

Node.js uses JavaScript on the server

A Node.js app runs in a single process, without creating a new thread for every request.

Node.js provides a set of asynchronous I/O primitives in its standard library that prevent

JavaScript code from blocking and generally,

libraries in Node.js are written using non-blocking paradigms, making blocking behavior the exception

rather than the norm.

When Node.js performs an I/O operation, like reading from the network, accessing a database or the filesystem,

instead of blocking the thread and wasting CPU cycles waiting, Node.js will resume the operations when

the response comes back.

This allows Node.js to handle thousands of concurrent connections with a single server without introducing

the burden of managing thread concurrency,

which could be a significant source of bugs.

Node.js has a unique advantage because millions of frontend developers that write JavaScript for the browser are

now able to write

the server-side code in addition to the client-side code without the need to learn a completely different language.

In Node.js the new ECMAScript standards can be used without problems, as you don't have to wait for all

your users to update their

browsers - you are in charge of deciding which ECMAScript version to use by changing the Node.js version,

and you can also enable specific

experimental features by running Node.js with flags.

In the browser, most of the time what you are doing is interacting with the DOM, or other Web Platform APIs

like Cookies. Those do not exist in Node.js, of course. You don't have the document,

window and all the other objects that are provided by the browser.

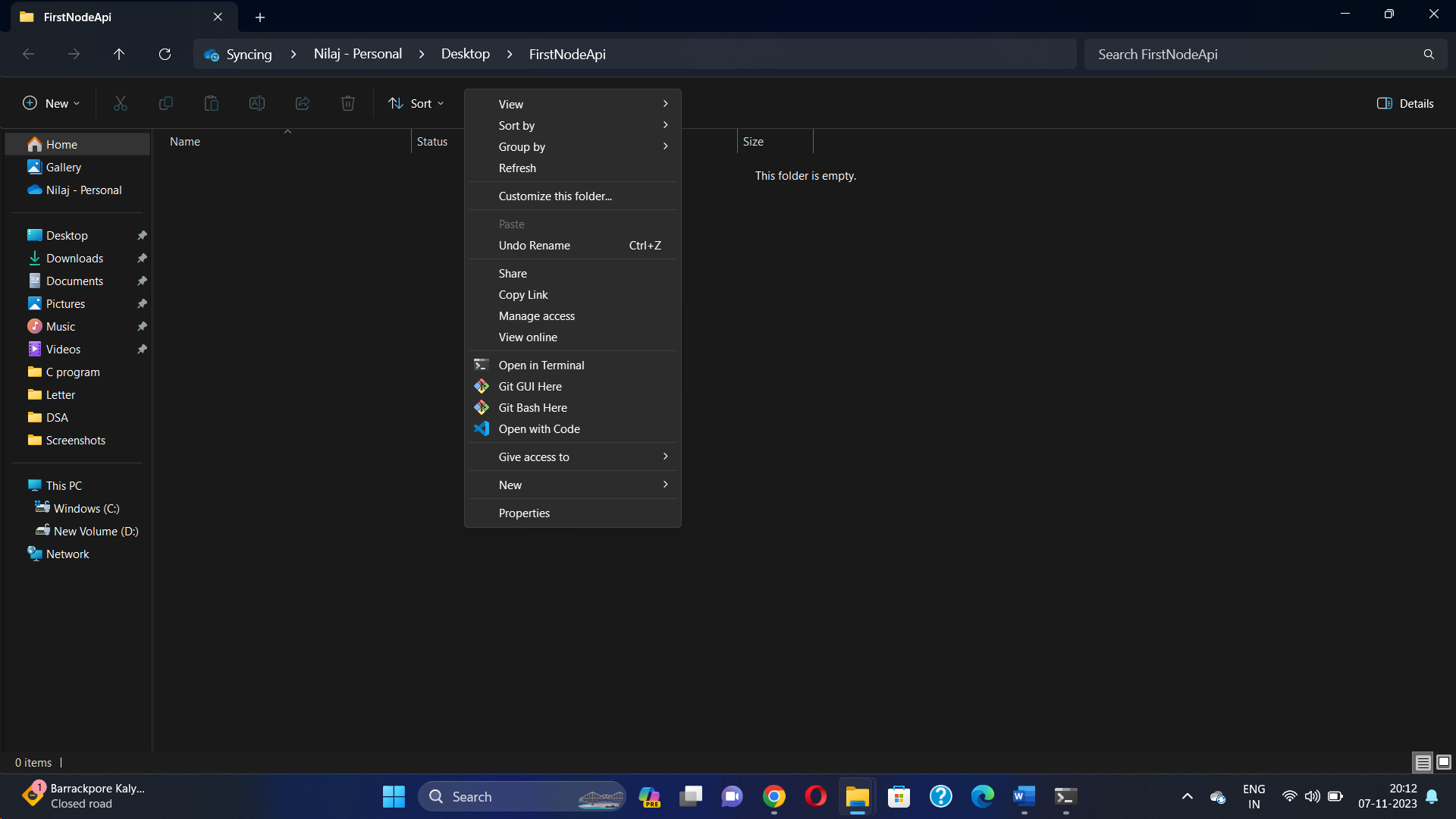
**How to create a REST API with Node.js and Express**

**1.Create a folder in desktop or any drive location mkdir FirstNodeApi**

**Or** 

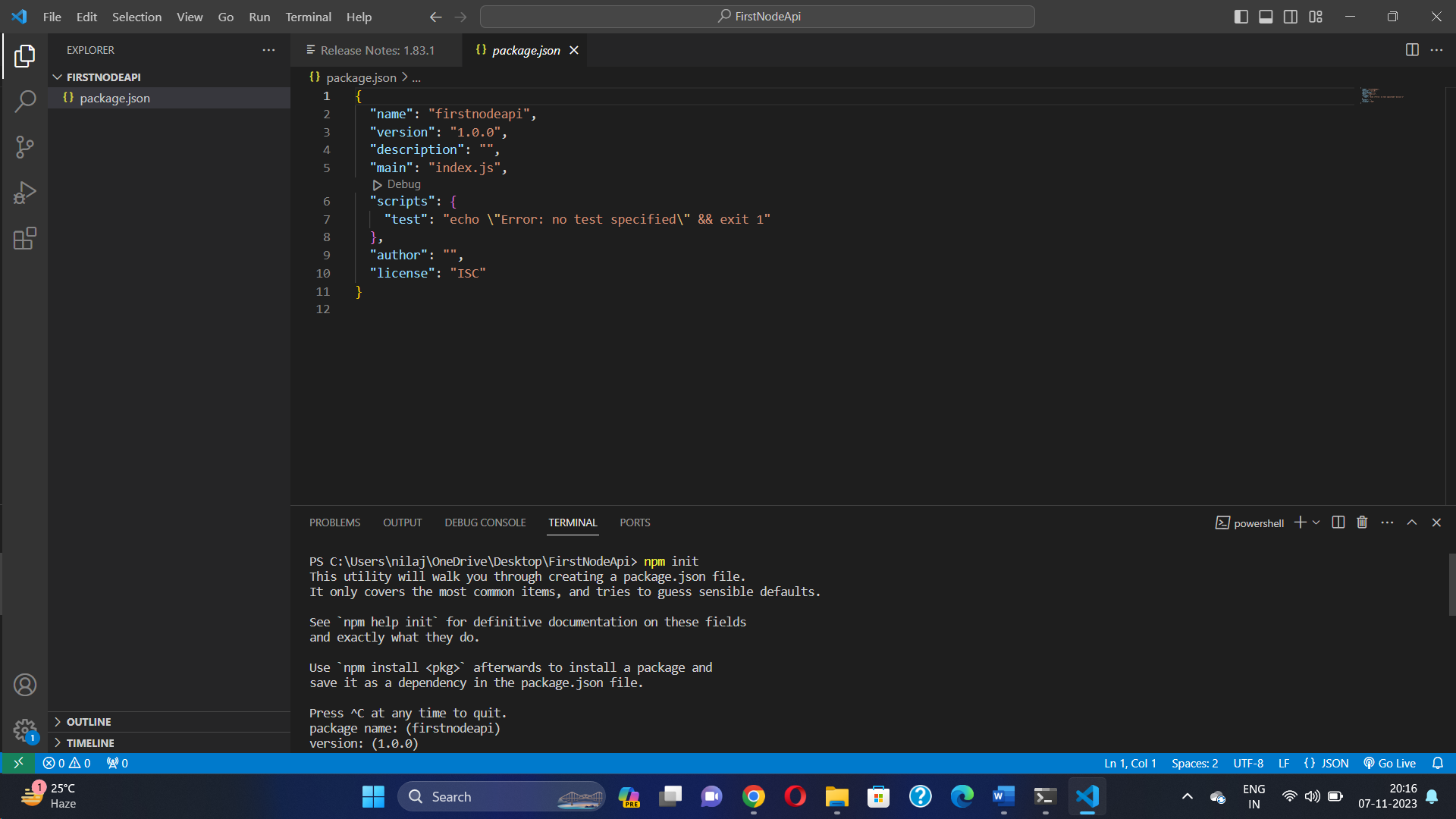
****

**Double click on folder then right click open with vs code**



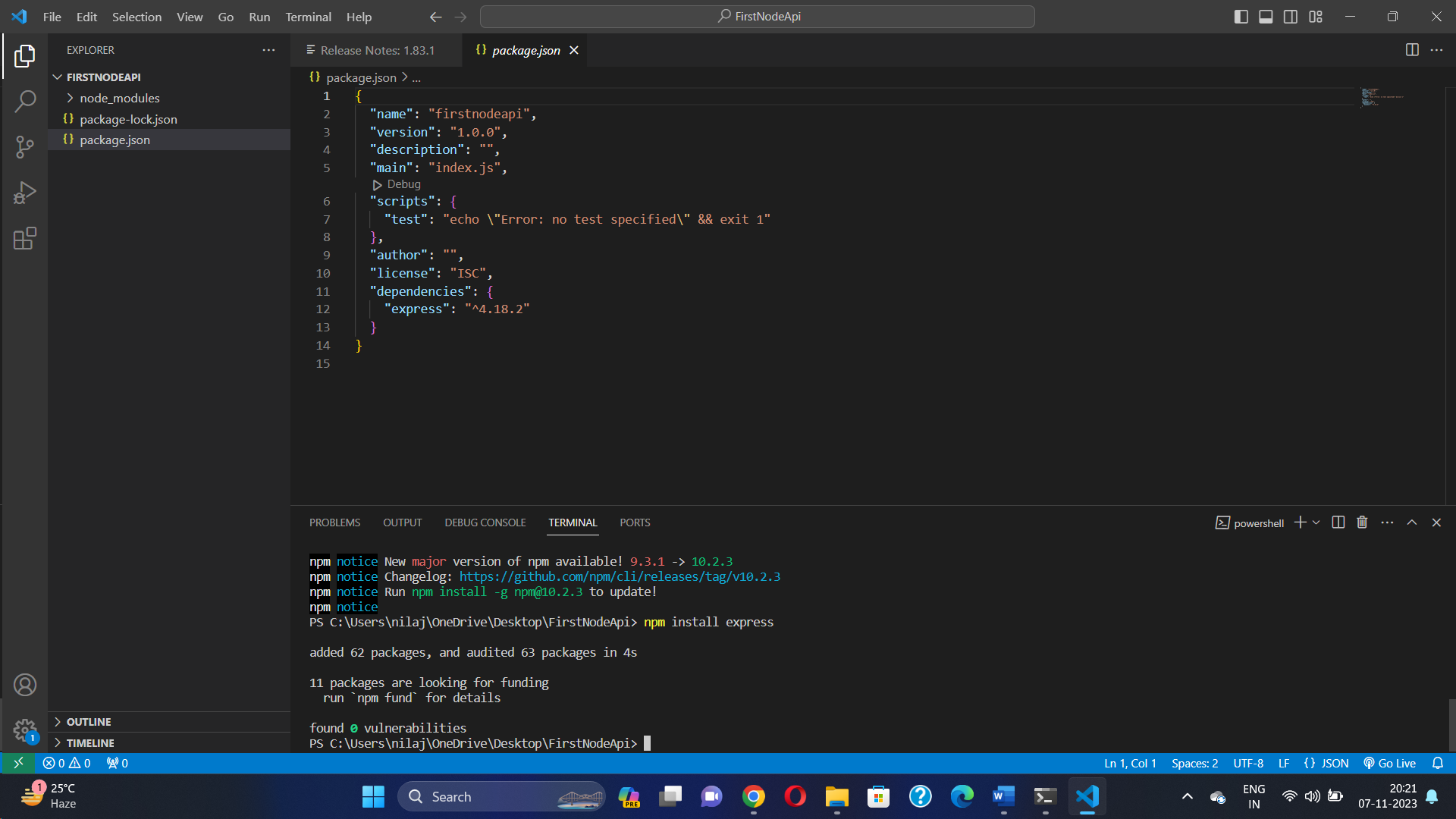
### **2.Initialize a new Node.js application**

in terminal type npm init Enter Enter Enter and so on…. See package.json is created



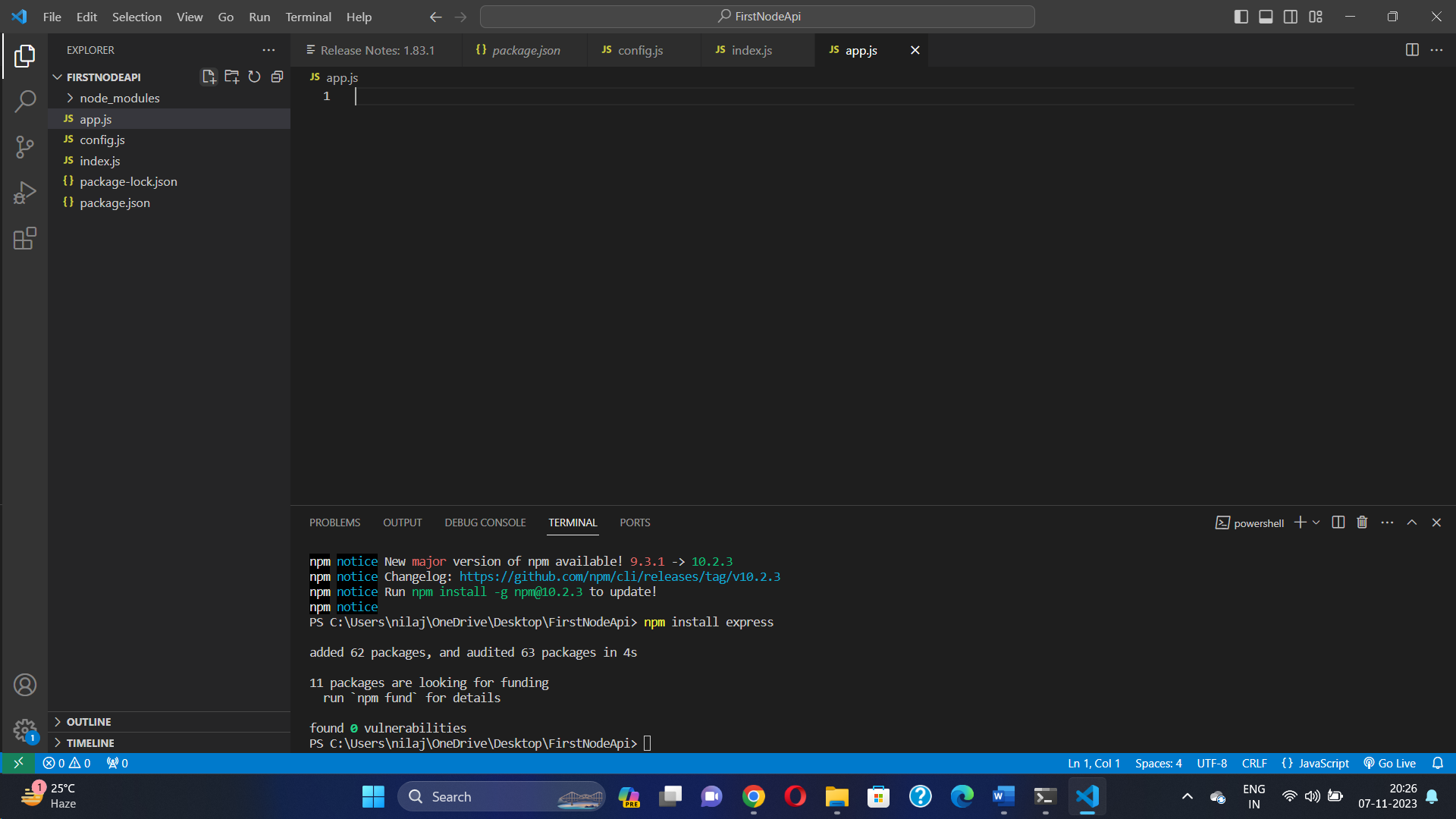
### **Step 4: Install Express and other dependencies**

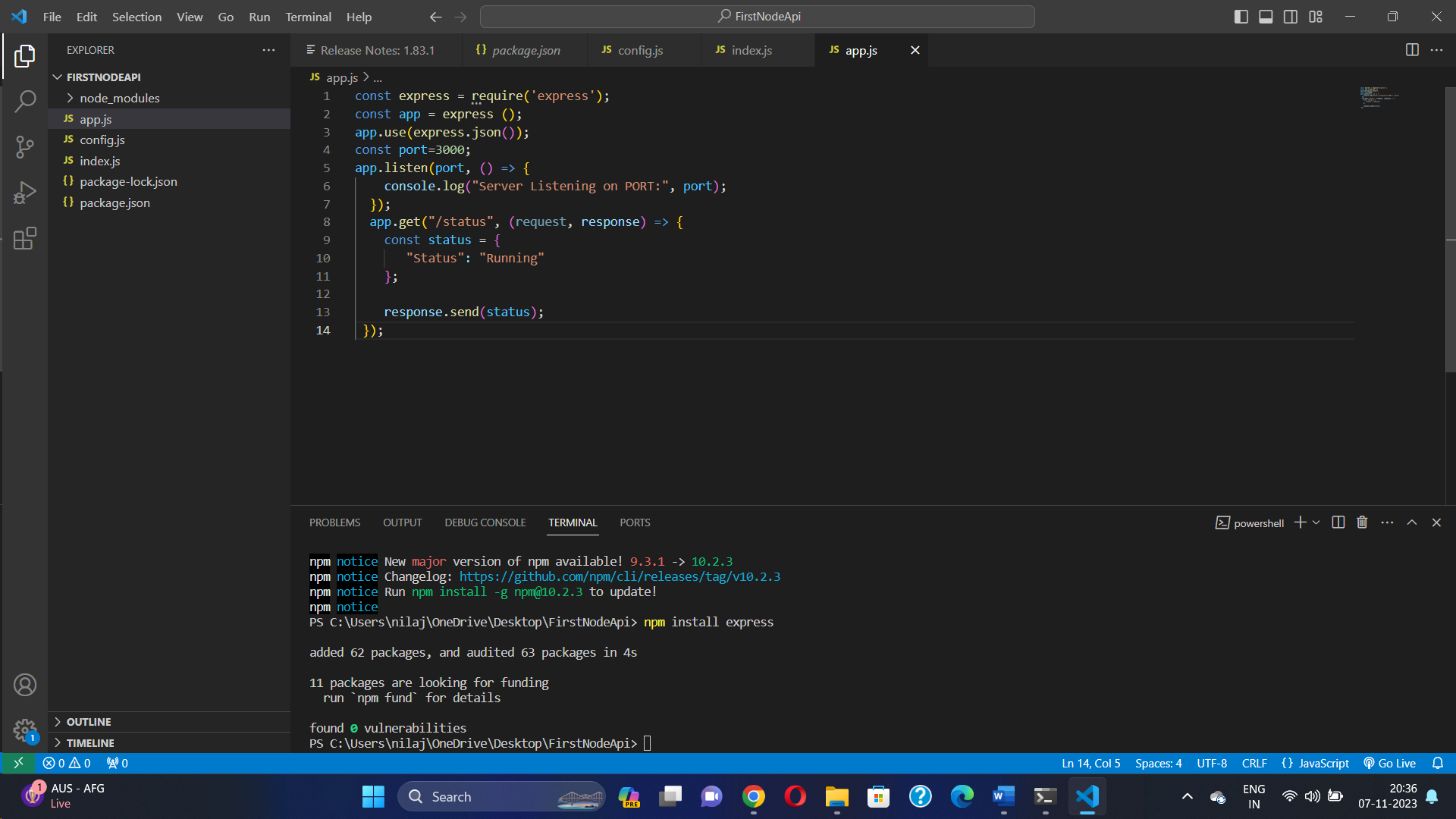
npm install express package-lock.json created



### **Step 5: Import necessary modules**

Create app.js file and config.js and index.js

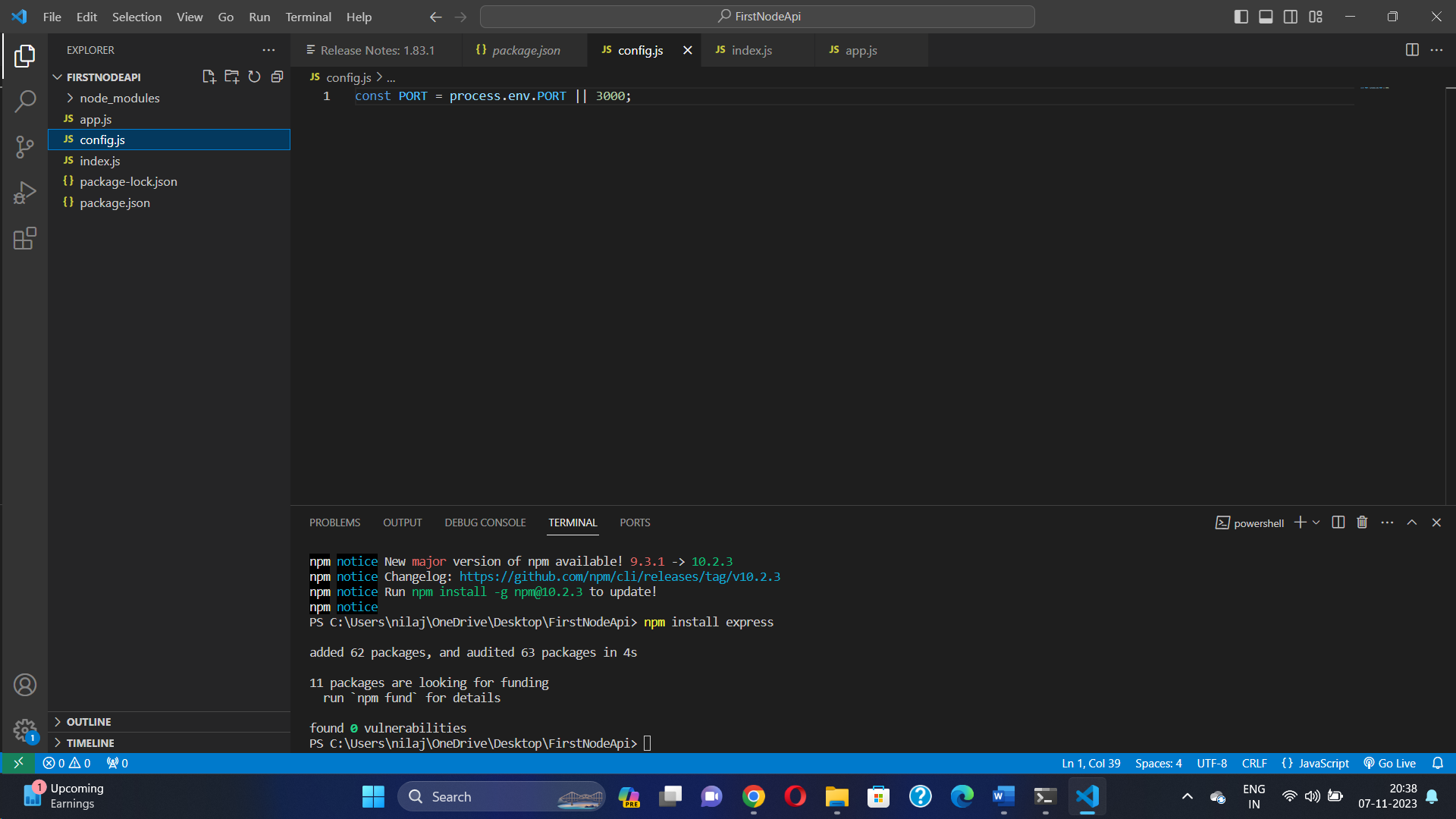
 app.js



### **Step 6: Define an endpoint**

### **Step 7: Define a route that listens to requests**

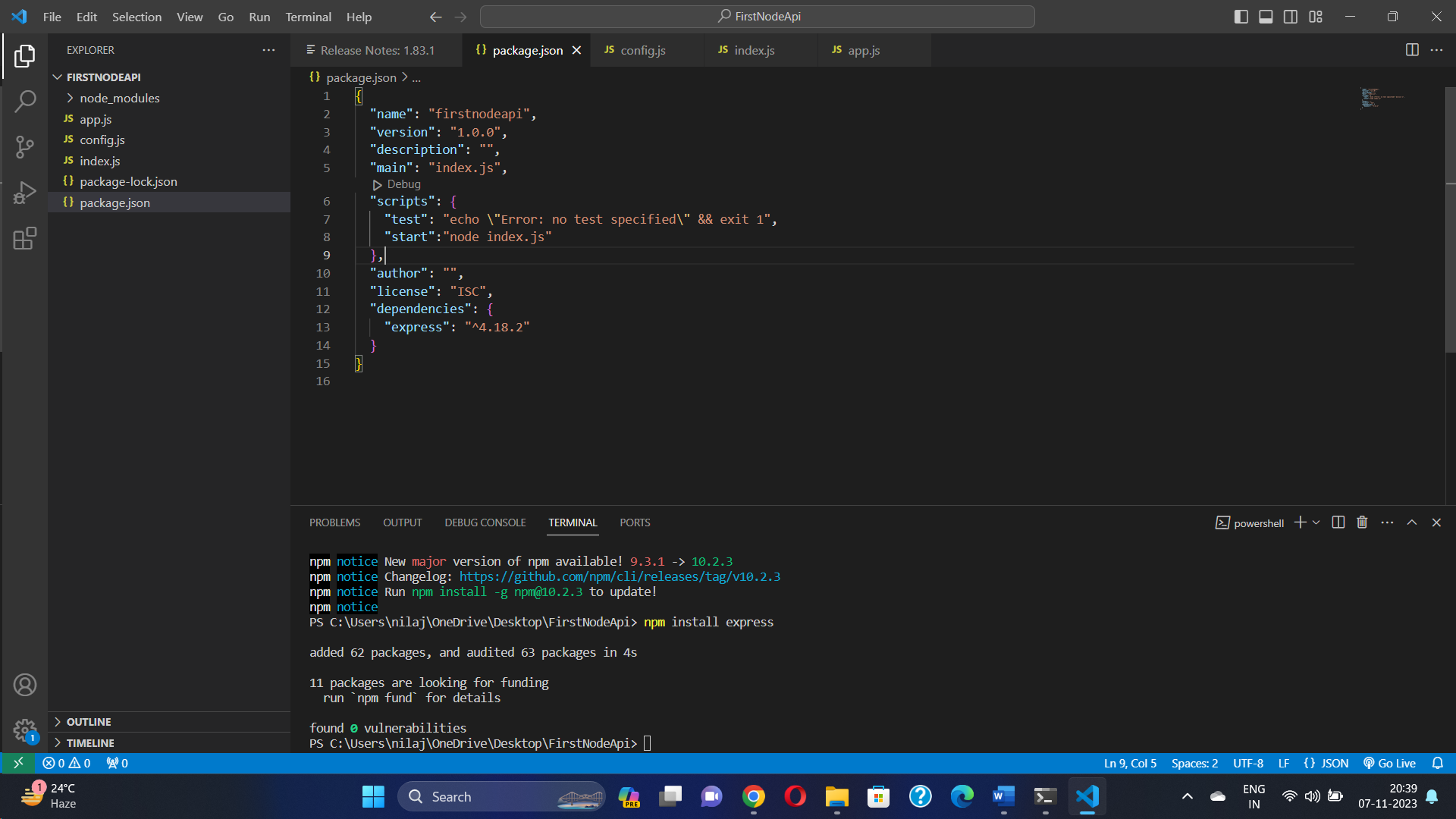
Config.js



Index.js

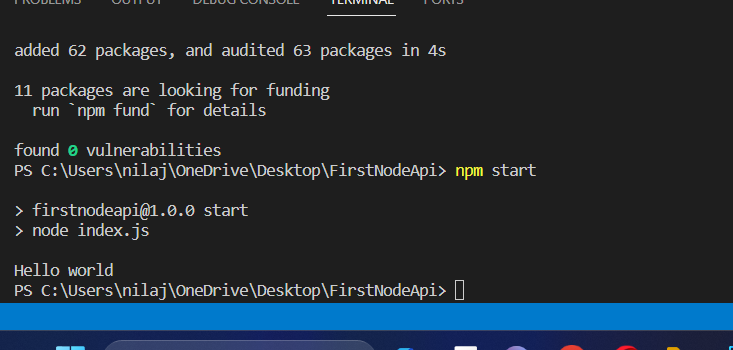


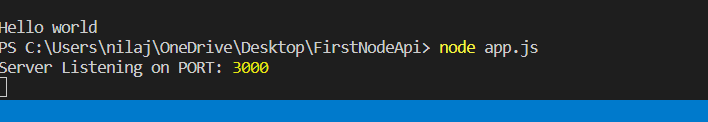
Package.json



Add in package.json in script area , “start”: “node index.js”

Step 8 Now run the npm start in terminal see output Hello world





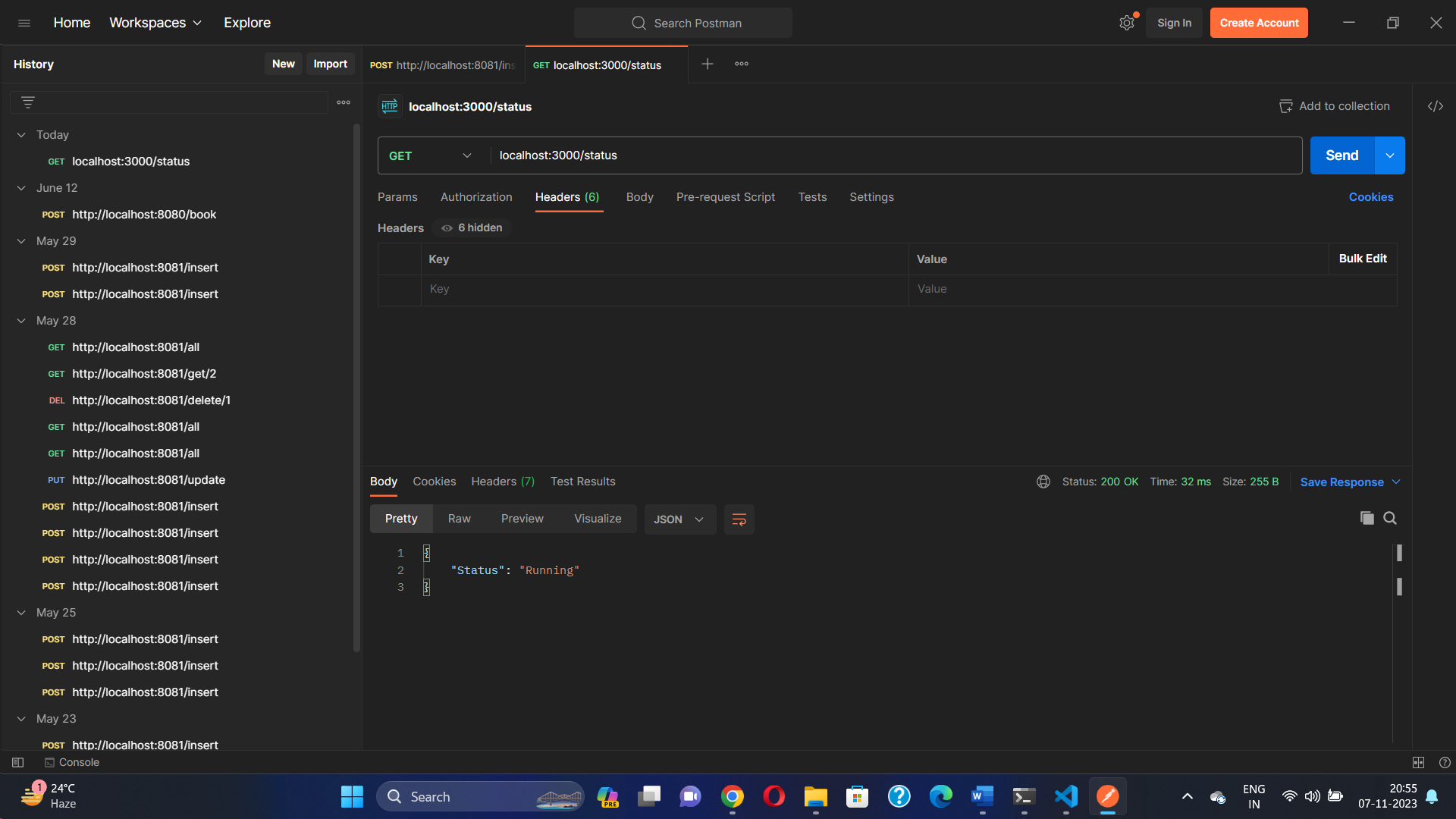
Server Listening the port 3000 run by node app.js

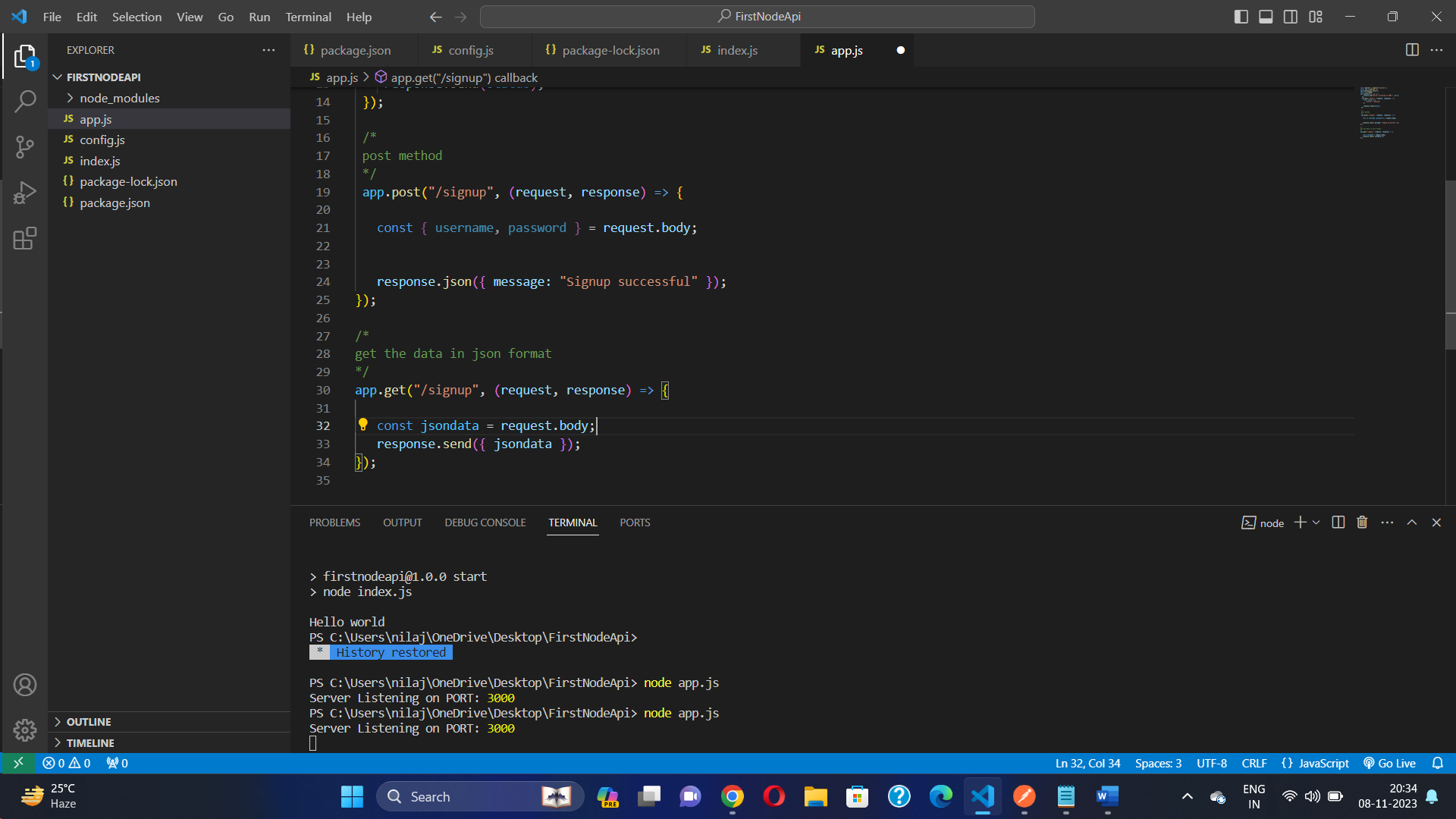
NOW GO TO POSTMAN

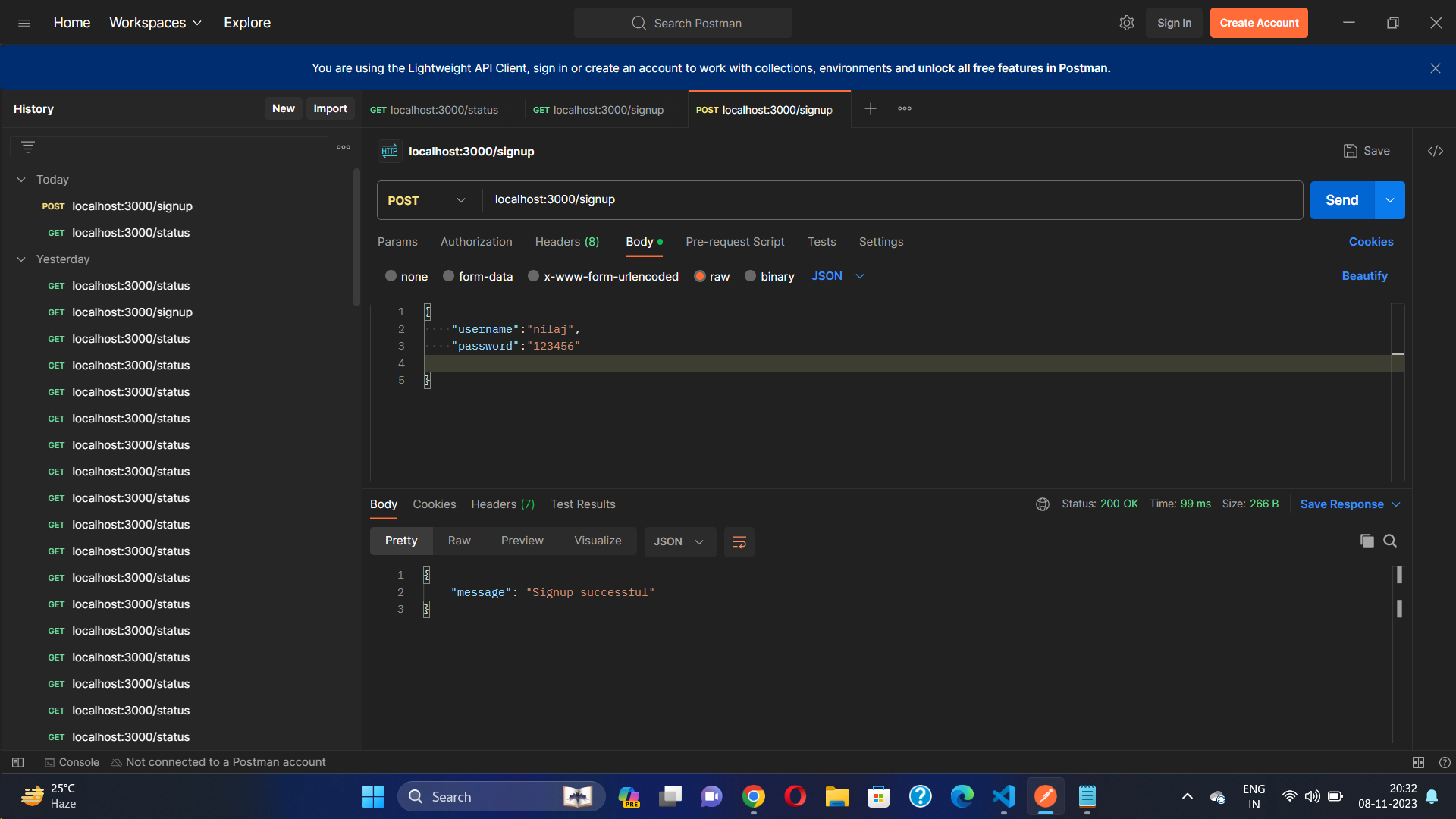


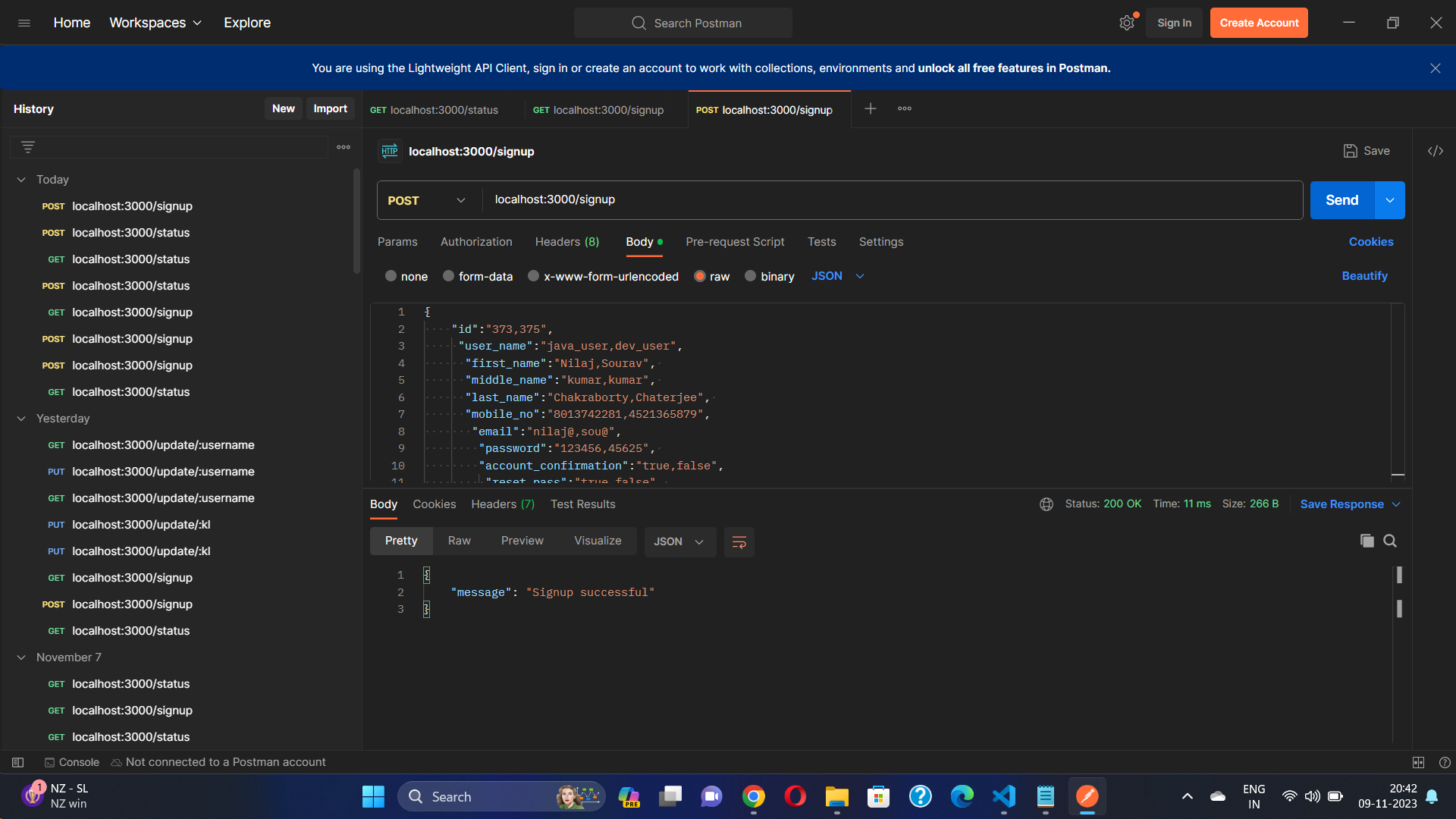
Double click on postman

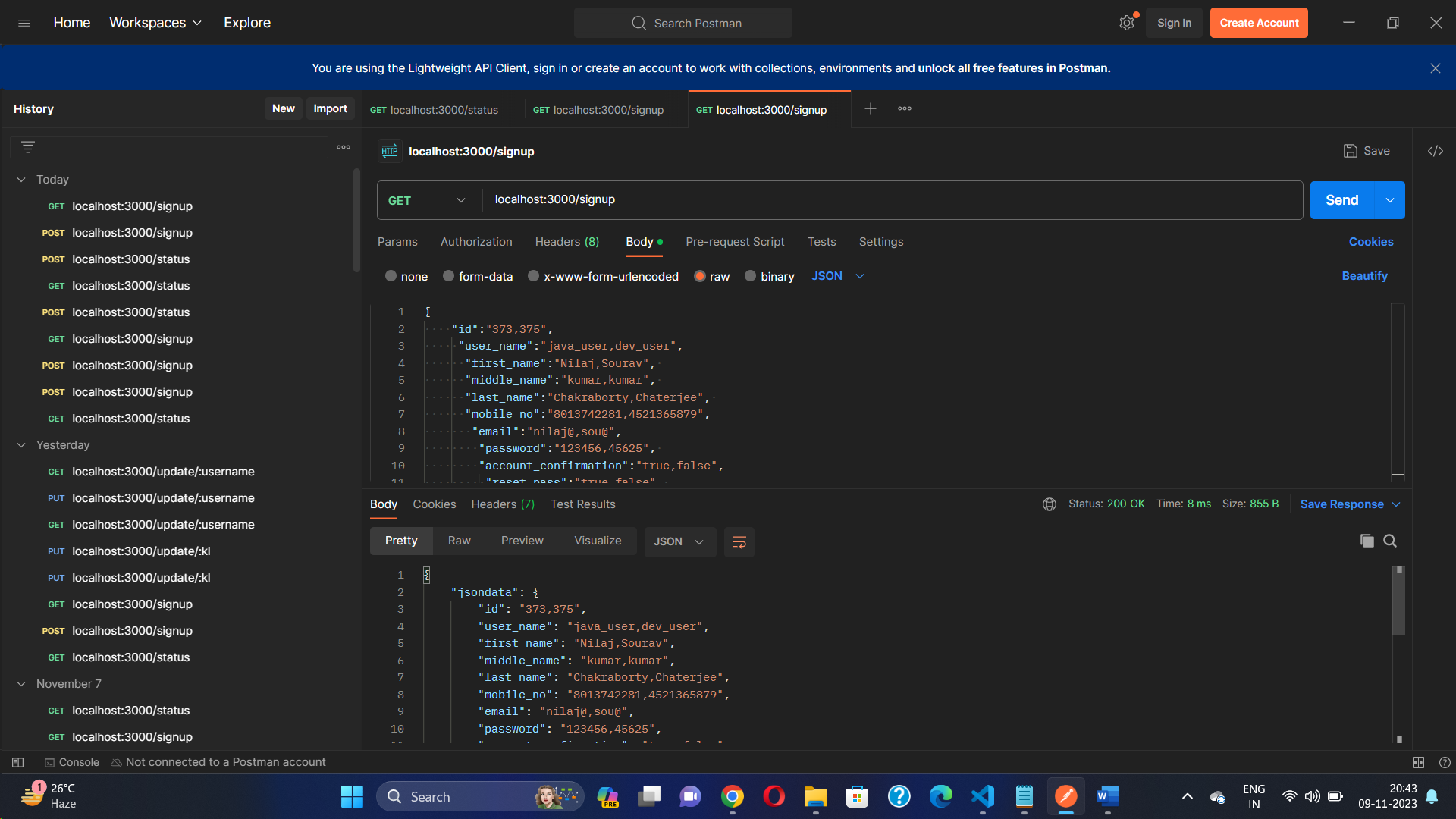
And Get https:3000/status hit send see api hitting by the postman and api is running



Post Method Sign in and getmethod to show the data in jsonfile

Here Post signup hitting the api by postman





Get method output hiiting Api using PostMan