

Project Progress:

1. Implementing the API to fetch the PC configuration of the user.

First step in our project is to determine the user's system configuration. To fetch user's system configuration, we are using a library named "system information".

Library:

It is meant to be used as backend/server-side library.

Basic Requirements to run the library:

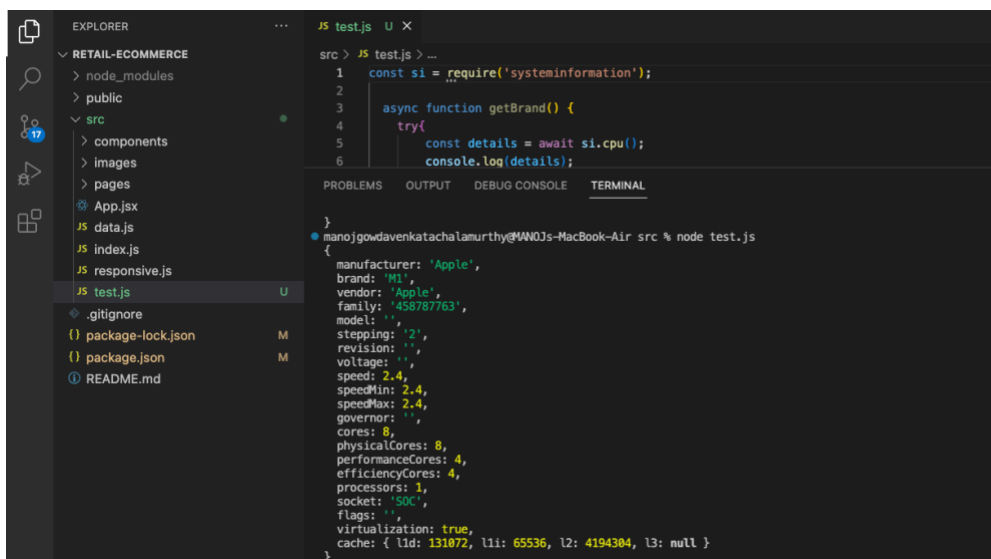
To use this package, we should have node.js version 4.0 and above installed.

Usage:

```
const si = require('systeminformation');

// promises style - new since version 3
si.cpu()
  .then(data => console.log(data))
  .catch(error => console.error(error));
```

Result:



The screenshot shows a code editor with a file explorer on the left and a terminal at the bottom. The file explorer shows a project structure for 'RETAIL-ECOMMERCE' with files like 'package-lock.json', 'package.json', and 'README.md'. The terminal shows the output of running 'node test.js', which displays a detailed JSON object representing the system configuration of a MacBook Air.

```
src > JS test.js > ...
1  const si = require('systeminformation');
2
3  async function getBrand() {
4    try{
5      const details = await si.cpu();
6      console.log(details);
7    }
8  }
9
10 manojgowdavenkatchalamurthy@MANOJS-MacBook-Air src % node test.js
{
  manufacturer: 'Apple',
  brands: 'M1',
  vendor: 'Apple',
  family: '458787763',
  model: '',
  stepping: '2',
  revision: '',
  voltage: '',
  speed: 2.4,
  speedMin: 2.4,
  speedMax: 2.4,
  governor: '',
  cores: 8,
  physicalCores: 8,
  performanceCores: 4,
  efficiencyCores: 4,
  processors: 1,
  socket: 'S0C',
  flags: '',
  virtualization: true,
  cache: { l1d: 131072, l1i: 65536, l2: 4194304, l3: null }
}
```

From fetching the above details, we will suggest better tuning for flawless gameplay.