

Module 2 – Description

Tested on both the systems below-

Macbook Pro (13 inch, 2017, Two Thunderbolt 3 ports)

Operating system: mac Catalina (10.15.7)

Processor: 2.3 GHz Dual-Core Intel Core i5

Memory 8GB 2133 MHz LPDDR3

Graphics Intel Iris Plus Graphics 540 1536 MB

Safari Version 14.0

Operating system: Windows 10 home (20H2) – created on this system

OS build:19042.662

Experience: Windows feature experience pack 120.2212.551.0

HP ENVY X360 convertible 15-DS1XXX

Processor AMD Rizen 5 4500U with Radeon graphics 2.38GHz

Memory 16GB

Browser version: Microsoft edge (87.0.664.57)

Tested using windows CMD terminal

Node.js version (v14.15.1)

Installation steps:

1. To start, first go to <http://nodejs.org> to download and install Node.js
2. Once the installation completes, make sure to install other dependencies that installer will point of to you like visual studio-built tools and chocholaty script comes with the installer to install the dependencies
3. Once installation finishes you should be able to type the node command on a terminal and enter the interactive mode of node.js.
4. At the console, type npm install

Project creation steps:

1. Npm install express on terminal and check the installed packages using npm list –depth=0
2. To initialize a new project, type npm init on terminal, this will create a package.json file
3. Added express as the dependency of the project by executing npm install express –save
4. Add this line to package.json file in the script section: “start”:”node server.js” this automatically starts server.js by using command npm start on terminal.

Explanation of the file: server.js

This file creates the server using http module and asks it to listen on port 80. It parses the incoming http request using the URL module and it serves different html or image recourses based on the request for e.g., when a request comes from localhost:80 URL then it serves the static mainpage.html resource. For each image in the main page of html is requested from the server and the server serves the static image resources in response to it.

The page loads 3 images files namely: allu-parantha_290_220.jpg, bhelpuri_290_220.jpg and logo_small.jpg When server gets the request for one of these images, it writes the http header with response code 200 and value {“content-type”:”image/jpeg”}

Then it sends the image file in the response file body and finally ends the response.

Expected outcome:

```
npm
C:\Users\heman\OneDrive\Documents\nodejs\firstproject>npm start

> firstproject@1.0.0 start C:\Users\heman\OneDrive\Documents\nodejs\firstproject
> node server.js

/
inside /
/images/allu-parantha_290_220.jpg
fetch allu parantha
/images/bhelpuri_290_220.jpg
inside bhelpuri
/images/dark-wood.jpg
/images/logo_small.jpg
logo
/introduction.html
/introduction.html
/
inside /
/images/allu-parantha_290_220.jpg
fetch allu parantha
/images/bhelpuri_290_220.jpg
inside bhelpuri
/images/dark-wood.jpg
/images/logo_small.jpg
logo

```

