

#### 4A.Create a Node.JS Application which serves a static website.

Node.js is an asynchronous event-driven JavaScript runtime built on Chrome's V8 JavaScript engine. With Node, you can build all types of applications from desktop to web. Let's build a Static Website with Node JS Webserver

Softwares used

Node.js

NPM

Express

VS Code

Node comes with NPM, which is package manager for JavaScript. NPM allows the discovery and usage of common packages functionality.

As mentioned above, for demonstration purpose we will use the Express framework. Express is a minimal and flexible Node.js web application framework that provides a robust set of features for web and mobile applications.

Download Node.js installable setup from its official site and install with default options. After installation, you will be able to run node and npm on the command line.

Steps:

Create a folder named NodeJSApp.

Create public subfolder in NodeJSApp and store an image/file which serve as static content

Create a views folder in NodeJSApp and create file Demo.ejs which displays the static content on browser

Navigate to NodeJSApp folder and in command prompt and run a command as-

1: npm init -y

This command will initialize the folder to be a node project and create package.json. This file holds the node js project configuration.

As we are going to use the single external framework "express", we will install this dependency package in our project with below command-

npm install express --save

Once we install this our package.json will be something like this.

```
{
  "name": "node-web-server",
  "version": "1.0.0",
  "description": "A minimalist Web Server with Node.js",
  "main": "server.js",
```

```

"scripts": {
  "test": "echo \"Error: no test specified\" && exit 1"
},
"keywords": [],
"author": "Author_Name",
"license": "ISC",
"dependencies": {
  "express": "^4.16.3"
}
}

```

Prepare Static Contents: Demo.ejs:

```

<!DOCTYPE html>
<html>
<head>
  <title>Static Middleware Demo</title>
</head>
<body>
  
</body>
</html>

```

Writing Server:

The server.js will be our main file that will create and run a webserver instance. As mentioned we are using Express for this. Below is the code which will let you initialize and start express.

```

const express = require('express')
const path = require('path')
const app = express()

// Static Middleware
app.use(express.static(path.join(__dirname, 'public')))

// View Engine Setup
app.set('views', path.join(__dirname, 'views'))
app.set('view engine', 'ejs')

app.get('/', function(req, res){
  res.render('Demo')
})

```

```
app.listen(8080, function(error){  
  if(error) throw error  
  console.log("Server created Successfully")  
})
```

Starting the Node JS Webserver:

Starting the server is as simple as running below command.

`node server.js`

That's it, your server is up and running on port 8080.