Try practicing all example codes, step by step guide is given below for each type of project.

**How to make a NodeJs Application (install nodejs from the provided link):**

1. Create a new directory for your project and navigate to it using the terminal.
   1. mkdir my-node-app
   2. cd my-node-app
   3. npm init –y (*initializes a new Node.js project and automatically creates a default package.json file without asking any questions*)
2. Inside your project directory, create a new JavaScript file (e.g., app.js) using your favorite code editor.
3. Paste your code in your app.js file. Example code is given below (taken from slide 21):

const http = require('http’);

const hostname = '127.0.0.1’;

const port = 3000;

const server = http.createServer((req, res) => {

res.statusCode = 200;

res.setHeader('Content-Type', 'text/plain’);

res.end('Hello World’);

});

server.listen(port, hostname, () => {

console.log(`Server running at http://${hostname}:${port}/`);

});

1. Save the file and in the terminal write node app.js.
2. Go to the provided url and observe the output.

**In case you are running router-based examples without express, do the following:**

1. Steps 1 and 2 are same.
2. Paste your code in your app.js file. Example code is given below (taken from slide 57):

const http = require('http');

http.createServer(function (req, res) {

res.writeHead(200, { 'Content-Type': 'text/html' });

const url = req.url;

     const method = req.method;

     // Regular expression to match '/contact/:id'

    const contactUrlPattern = /^\/contact\/(\d+)$/;

    const match = url.match(contactUrlPattern);

 if (method === 'GET' && url === '/contact') {

        res.write(' Welcome to Get');

         res.end();

     }    else if (method === 'POST' && url === '/contact') {

        res.write(' Welcome to Post');

        res.end();

     }   else if (method === 'PUT' && match) {

        const id = match[1]; // Extract the ID from the URL

        res.write(`Updated contact with ID: ${id}`);

        res.end();

    }     else if (method === 'DELETE' && match) {

        const id = match[1]; // Extract the ID from the URL

        res.write(`Deleted contact with ID: ${id}`);

        res.end();

    } else {

        res.write('Hello World!');

         res.end();

     }

}).listen(3000, function () {

     // The server object listens on port 3000

    console.log("server start at port 3000");

});

1. Run the code using node app.js or nodemon app.js (install nodemon first).
2. Then:
   1. You can only test get command by visiting the url. For post, put and delete, you need some other testing methods like postman,curl or thunder client (VS extension).
   2. curl.exe -X POST http://localhost:3000/contact -d '{"key":"value"}' -H "Content-Type: application/json"
   3. curl.exe -X PUT http://localhost:3000/contact/1 -d '{"key":"updatedValue"}' -H "Content-Type: application/json"
   4. curl.exe -X DELETE <http://localhost:3000/contact/1>

**In case you are running router-based examples with express, do the following:**

1. Install express first.
2. Steps 1 and 2 are sameas example 1.
3. Paste your code in your app.js file. Example code is given below (taken from slide 69):

// app.js

const express = require('express');

const app = express();

const port = 3000;

// Import contact routes

const routes = require('./routes');

// Use the contact routes

app.use('/contact', routes);

// Start the server

app.listen(port, () => {

    console.log(`Server running at http://localhost:${port}`);

});

1. Create routes.js file and paste your code in your routes.js file. Example code is given below (taken from slide 68):

// routes.js

const express = require('express');

const router = express.Router();

router.get('/', (req, res) => {

    res.send('List of users’); });

router.post('/', (req, res) => {

    res.send('User created’); });

router.get('/:id', (req, res) => {

    const userId = req.params.id;

    res.send(`User details for user with ID: ${userId}`); });

router.put('/:id', (req, res) => {

    const userId = req.params.id;

    res.send(`User with ID: ${userId} updated`); });

router.delete('/:id', (req, res) => {

    const userId = req.params.id;

    res.send(`User with ID: ${userId} deleted`); });

module.exports = router;

1. Run the code using node app.js or nodemon app.js (install nodemon first).
2. Then:
   1. You can only test get command by visiting the url. For post, put and delete, you need some other testing methods like postman,curl or thunder client (VS extension).
   2. curl.exe -X POST http://localhost:3000/contact -d '{"key":"value"}' -H "Content-Type: application/json"
   3. curl.exe -X PUT http://localhost:3000/contact/1 -d '{"key":"updatedValue"}' -H "Content-Type: application/json"
   4. curl.exe -X DELETE <http://localhost:3000/contact/1>