

day-02 assignment

Name - Soham Sarkar

College Name – MCKV INSTITUTE OF ENGINEERING



June 13, 2025

**Express Routing:**

a) Create a basic Express server with three routes:

* + - / (Home route)
    - /about (About route)
    - /contact (Contact route)

What response should each route send?

b) How do you handle 404 (Not Found) routes in Express? Demonstrate with code.

c) Explain the difference between app.get() and app.post() with an example.

/\*

**a) Create a basic Express server with three routes:**

/ (Home route)

/about (About route)

/contact (Contact route)

\*/

const express = require('express');

const app = express();

const port = 3001;

// Home route

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

  res.send('Welcome to the Home Page');

});

// About route

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

  res.send('This is the About Page');

});

// Contact route

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

  res.send('Contact me at : sohamsarkarofficial000@gmail.com');

});

// Start server

app.listen(port, () => {

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

});

**/\*b) How do you handle 404 (Not Found) routes in Express? Demonstrate with code.\*/**

// 404 handler (must be last)

app.use((req, res) => {

  res.status(404).send('404 Not Found: The page you are looking for does not exist. Love From Soham Sarkar');

});

**c) Explain the difference between app.get() and app.post() with an example.**

// app.get()

// Handles HTTP GET requests.

// Typically used to retrieve data from the server.

// Data is sent via the URL query string (e.g., ?name=Soham).

// app.post()

// Handles HTTP POST requests.

// Typically used to send data to the server (e.g., submitting a form).

// Data is sent in the request body, not the URL.

//Example Code

const express = require('express');

const app = express();

const port = 3000;

// Serve the login page

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

  res.send(`

    <html>

      <head>

        <title>Login Page</title>

      </head>

      <body style="font-family: Arial; text-align: center; margin-top: 100px;">

        <h2>Login</h2>

        <form action="/login" method="POST">

          <input type="text" name="username" placeholder="Username" required/><br><br>

          <input type="password" name="password" placeholder="Password" required/><br><br>

          <button type="submit">Login</button>

        </form>

      </body>

    </html>

  `);

});

// Dummy POST Handler

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

  res.send(`Login submitted! Username: ${username}, Password: ${password}`);

});

// Start server

app.listen(port, () => {

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

});

**Node.js File System (fs) Module:**

a) Write a program to create a new file and write some content into it using fs.writeFile().

b) How to append data to an existing file using Node.js?

c) Write a Node.js script to read a file and print its content to the console.

d) How do you delete a file using Node.js? Provide code.

**//a) Write a program to create a new file and write some content into it using fs.writeFile().**

const fs = require("fs");

const fileName = "data.txt";

fs.writeFile(fileName,"My first file",(err)=>{

    if(err){

        throw err;

    }

});

**//b) How to append data to an existing file using Node.js?.**

fs.appendFile(fileName,"\nThis is append version",(err)=>{

    if(err){

        throw err;

    }

})

**//c) Write a Node.js script to read a file and print its content to the console.**

fs.readFile(fileName,'utf-8',(err,item)=>{

    if(err){

        throw err;

    }

    console.log(item);

})

**// d) How do you delete a file using Node.js? Provide code.**

fs.unlink(fileName,(err)=>{

    if(err){ throw err}

})

**Path and fs packages:**

* + a) What is the use of the path module in Node.js? Give two examples.
  + b) Use path.join() to combine directory names and filenames to create a valid path.
  + c) What is the difference between fs.readFileSync() and fs.readFile()?

**/\*a) What is the use of the path module in Node.js?**

The path module provides utilities for working with file and directory paths. It helps ensure cross-platform compatibility when dealing with file system paths.

\*/

/\*

const path = require("path");

const filePath = '\Users\docs/report.pdf';

console.log(path.basename(filePath));   //path.basename() gives us filename from a full path.

console.log(path.extname(filePath));     //path.extname() gives us extension of the file.

\*/

**/\* b) Use path.join() to combine directory names and filenames to create a valid path.**

const fs = require("fs");

const path = require("path");

const directory = path.join(\_\_dirname,'myFolder');

const filePath = path.join(directory,'file1.txt');

fs.writeFileSync(filePath,'Hello From Soham ~');

**c) What is the difference between fs.readFileSync() and fs.readFile()?**

//Reading File using readFile() Method.

// fs.readFile()

// Asynchronous (non-blocking)

// Code execution does not wait for the file to be read.

// Requires a callback (or can be used with Promises).

fs.readFile(filePath,'utf-8',(err,data)=>{

    if(err){

        throw error;

    }

    console.log(data);

});

//Reading File Using readFileSync Method.

// fs.readFileSync()

// Synchronous (blocking)

// Code execution waits until the file is completely read.

// Useful for small scripts or during app startup where blocking is acceptable.

const data = fs.readFileSync(filePath,'utf-8');

console.log(data);