Assignment

Building a Basic Node.js Web Server with File Handling

Create a simple Node.js web server that serves static files from a directory. This assignment will help you understand the basics of Node.js, including file handling and creating a web server using intrinsic Node.js modules.

Task Overview:

You will build a Node.js application that starts a web server and serves files (such as HTML, CSS, and JS files) located in a specific directory. The server should be able to handle basic file reading operations and deliver content to the browser.

Requirements:

1. Setup Node.js Environment:
   - Ensure Node.js is installed on your system.
   - Initialize a new Node.js project using `npm init`.

2. Create a Directory for Static Files:
   - Create a folder named `public` in your project.
   - Inside `public`, add some static files (e.g., an `index.html`, a `style.css`, and a `script.js`).

Note: index.html file is linked with syle.css and script.js file

3. Build the Web Server:

   - Use Node.js' `http` module to create a web server.
   - Use the `fs` (File System) module to read files from the `public` directory.
   - Serve the requested file or a 404 error if the file doesn't exist.

4. Handle Different File Types:

   - Implement logic to correctly set the `Content-Type` header based on the file type (HTML, CSS, JavaScript, etc.).

5. Listening on a Port:

   - Configure the server to listen on a suitable port (e.g., 3000).

6. Testing:

   - Start the server and test it by accessing `http://localhost:3000/` in a web browser.
   - Try accessing different files and observe the server's response.

Deliverables:

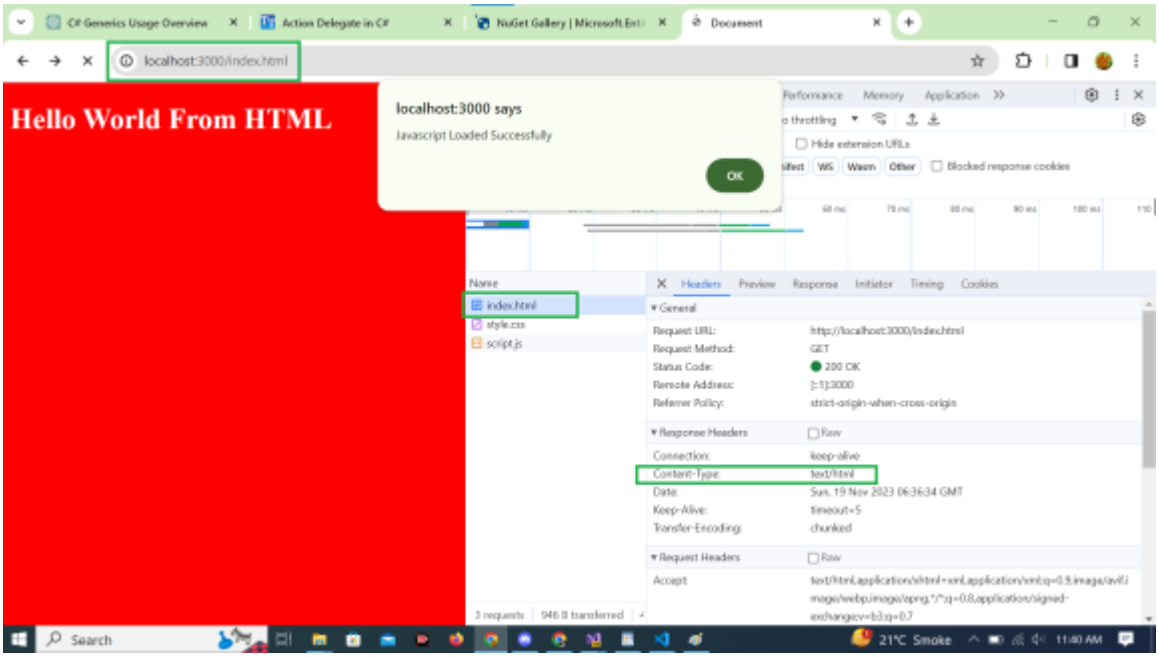
- A Node.js script that implements the web server.
- A `public` directory with sample static files.
- Instructions on how to run the server and access the static files.

Important Methods and Properties:
http.createServer ---> {for creating Http Server}
path.join() -----> { for finding path to the requested file}
String(path.extname(filePath)) ----> for extracting file extension from the completer filePath
req.url {for finding file from http request url}
fs.readFile(filePath:(error, content)=>{})

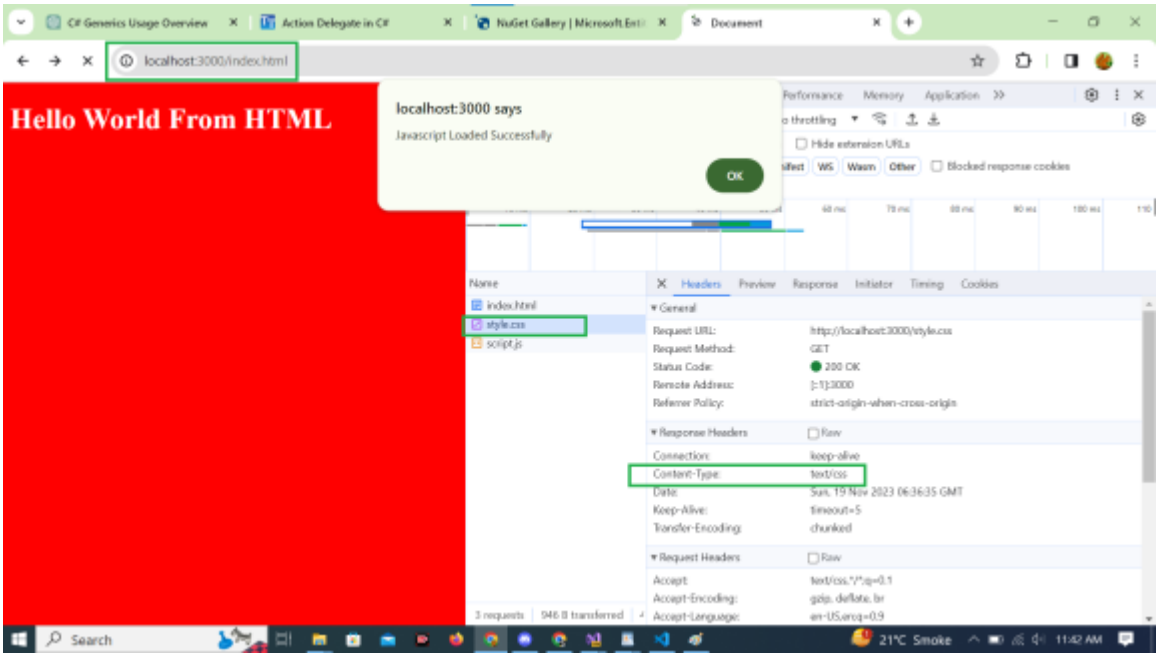
error.code == 'ENOENT' for file not found
res.writeHead(status code, {Content-Type: 'text/html' or 'text/js' or 'text/css' }
res.end(data, 'utf-8') ----> make sure there is no further data and uses utf-8 encoding

The sample output for the webpage is as under where css and js is linked with html.

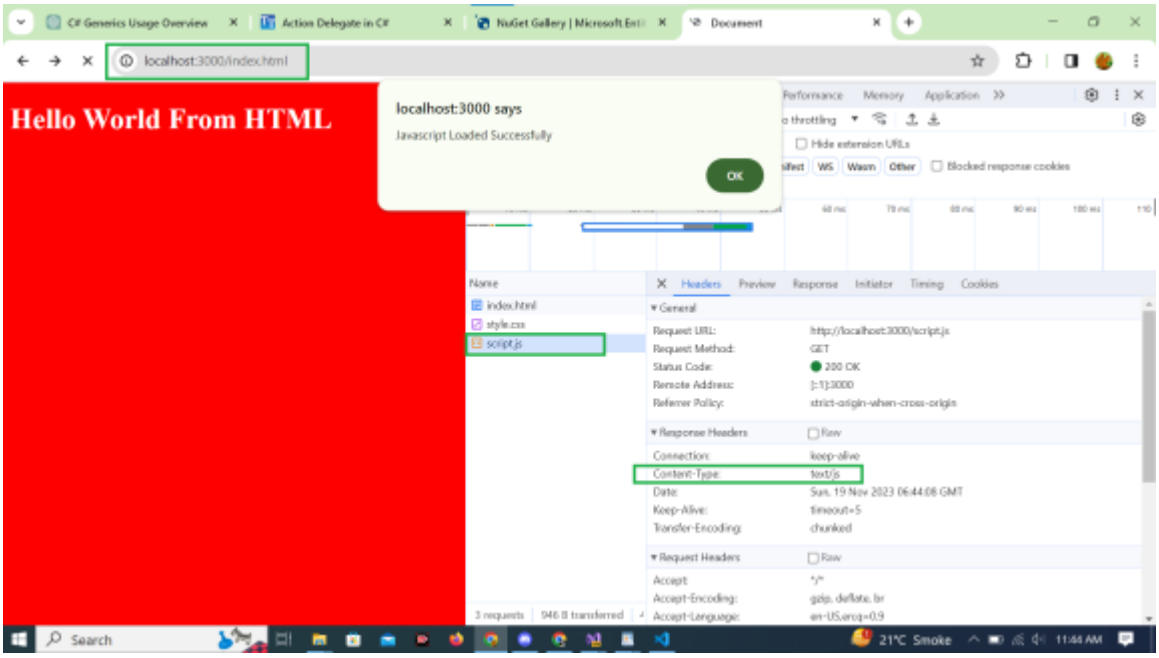
HTML Content Retrieval



CSS Content Retrieval



JS Content Retrieval


C# Generics Usage Overview
Action Delegate in C#
NuGet Gallery | Microsoft.Ext
Document
localhost:3000/index.html
Hello World From HTML
localhost:3000 says
Javascript Loaded Successfully
OK
Performance
Memory
Application
Hide extension URLs
Blocked response cookies
Name
index.html
style.css
script.js
Headers
Preview
Response
Initiator
Timing
Cookies
General
Request URL: http://localhost:3000/script.js
Request Method: GET
Status Code: 200 OK
Remote Address: [::1]:3000
Referrer Policy: strict-origin-when-cross-origin
Response Headers
Connection: keep-alive
Content-Type: text/js
Date: Sun, 19 Nov 2023 06:44:08 GMT
Keep-Alive: timeout=5
Transfer-Encoding: chunked
Request Headers
Accept: */*
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
3 requests
946 B transferred
Search
21°C Smoke
11:44 AM