Assignment No : 6

Create a basic server with Node.js and Express.js that serves your personal website static files.

Objective:

- Learn to create a basic web server using Node.js and Express.js, understanding the fundamentals of server-side JavaScript.
- Gain hands-on experience with routing and middleware in Express.js, essential concepts for web application development.
- Develop practical skills in serving static files, improving understanding of file system operations in a web server context.
- Learn to handle basic error scenarios, such as 404 errors, enhancing the robustness of the web application.

Theory:

1. Server Creation

Node.js allows developers to create web servers using JavaScript. Express.js, built on top of Node.js, simplifies this process by providing a robust set of features for web and mobile applications. The express() function is the top- level function exported by the Express module, used to create an Express application.

2. Routing

Routing in Express refers to how an application's endpoints (URIs) respond to client requests. Express provides methods that correspond to HTTP methods; for example, app.get() to handle GET requests and app.post() to handle POST requests. These methods specify a callback function (sometimes called "handler functions") to be executed when the application receives a request to the specified route and HTTP method.

3. Middleware

Middleware functions are functions that have access to the request object (req), the response object (res), and the next middleware function in the application's request-response cycle, commonly denoted by a variable named next. Middleware can execute any code, make changes to the request and

response objects, end the request-response cycle, and call the next middleware in the stack.

4. Static File Serving

Express provides a built-in middleware function express.static() to serve static files, such as images, CSS, JavaScript, etc. You can specify the directory from which to serve static assets by mounting the express.static middleware on a specific path.

5. Error Handling

Express comes with a default error handler, which takes care of any errors that might be encountered in the app. However, you can write your own error handling middleware to customize error responses. A common use case is handling 404 errors for routes that don't exist in your application.

Conclusion:

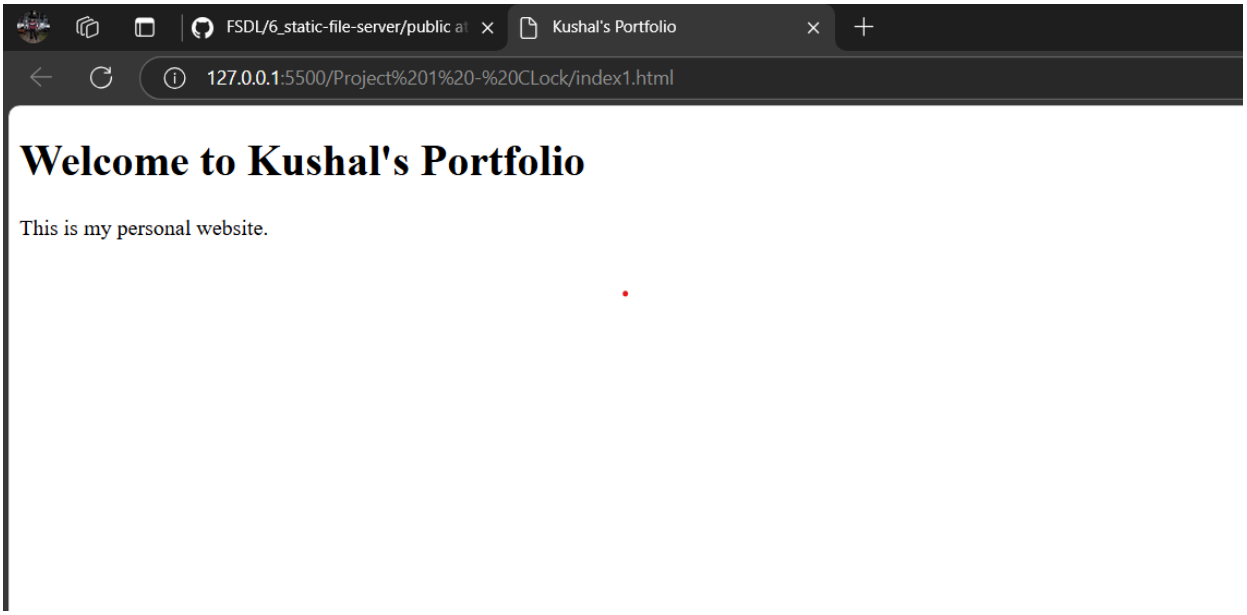
Creating a basic server with Node.js and Express.js to serve static files for a personal website is a fundamental task in modern web development. This exercise demonstrates the power and flexibility of JavaScript for both client-side and server-side programming. Through this task, developers gain practical experience with key concepts such as routing, middleware usage, and static file serving.

GitHub Repo with Source Code :

FSDL Assn 6

https://github.com/iamkushal2003/FDSL-Assignment-6

Output :-



404 - Page Not Found