Assignment 10

Roll No 71   T14 Batch

Aim : Program in node.js to create, read, write, rename, append and delete files.

LO Mapped : LO 6

Theory :

The File System module (fs) in Node.js is a built-in module that provides an interface for interacting with the file system of your computer. It enables you to perform various file operations like creating, reading, writing, renaming, appending, and deleting files. This module is crucial for handling file-related tasks in Node.js applications.

1. Creating Files:

The fs.writeFile() function is used to create a new file or overwrite an existing one with the specified data. It takes a filename, data to write, and a callback function to handle any errors that might occur during the operation.

2. Reading Files:

The fs.readFile() function is used to read the contents of a file. It allows you to specify the file's encoding, such as 'utf8' for text files. You provide a callback function that receives the file's content as a parameter, which you can then process.

3. Writing Files:

To write data to a file without overwriting its existing content, you can use fs.appendFile(). This function appends the specified data to the end of the file. If the file doesn't exist, it will be created.

4. Renaming Files:

The fs.rename() function is used to rename a file. You provide the old filename and the new filename as parameters. This operation can be useful for managing and organizing files in your application.

5. Appending to Files:

Appending data to a file is accomplished using fs.appendFile(). This function allows you to add content to an existing file without affecting its previous data. It also creates the file if it doesn't exist.

6. Deleting Files:

To delete files, you can use fs.unlink(). This function takes the filename as a parameter and deletes the file if it exists. Deleting files can be essential for managing resources and cleaning up unused data.

The File System module provides these operations asynchronously, meaning that they won't block the Node.js event loop. You typically pass callback functions to these operations to handle errors or process the results once the operation is complete. While these functions are powerful, you should always handle errors gracefully to ensure the robustness of your application. Additionally, Node.js also provides synchronous versions of these functions, but they can potentially block the event loop and should be used with caution, especially in performance-critical applications.

Conclusion :

Understood the fs module in node.js and the various operations in it. Went through the documentation on web for the same. Explored various other operations in fs module.

LO 6 was achieved.