2024

File Operations with Logging

JOLIANA EMAD KAMAL NAGUIB 20P3292

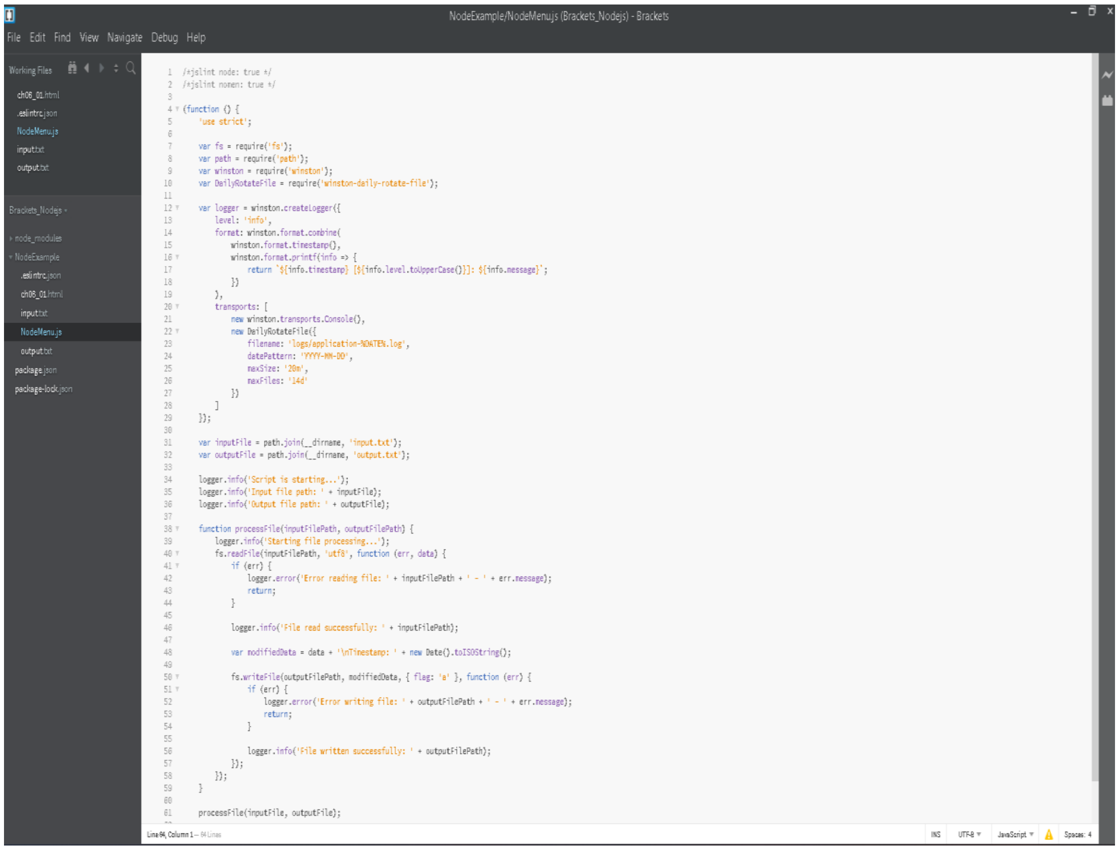
SAMSUNG INNOVATION CAMPUS

TASK Node.js

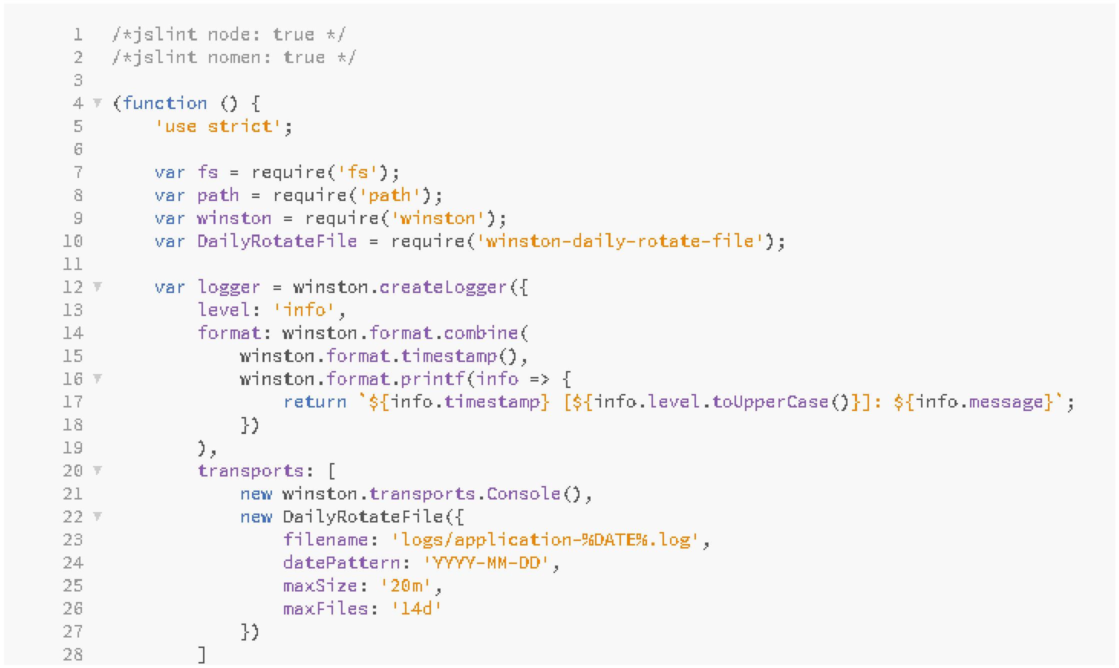
```
File Operations with Logging:

Write a Node.js script that reads the contents of a file asynchronously, modifies the contents by appending a timestamp, and writes the
updated contents to a new file.
Ensure the file operations are performed asynchronously using the fs module.
Logging:

Use the winston module to log the file operations. Each time a file is read, written, or an error occurs, log a message with the details of
the operation.
Rotate the log files on a daily basis using the winston-daily-rotate-file module.
Steps:

Install necessary modules: npm install winston winston-daily-rotate-file fs.
Create a function that reads a file asynchronously (readFile), appends a timestamp, and writes it to a new file (writeFile).
Implement logging using the winston module to log the file read, write, and any errors.
Ensure log rotation happens daily.
Expected Outcome:
A Node.js script that performs the file operations and logs the process with daily log rotation.
This task involves working with file operations and implementing logging, combining concepts from both sections.
```

NodeMenu.js

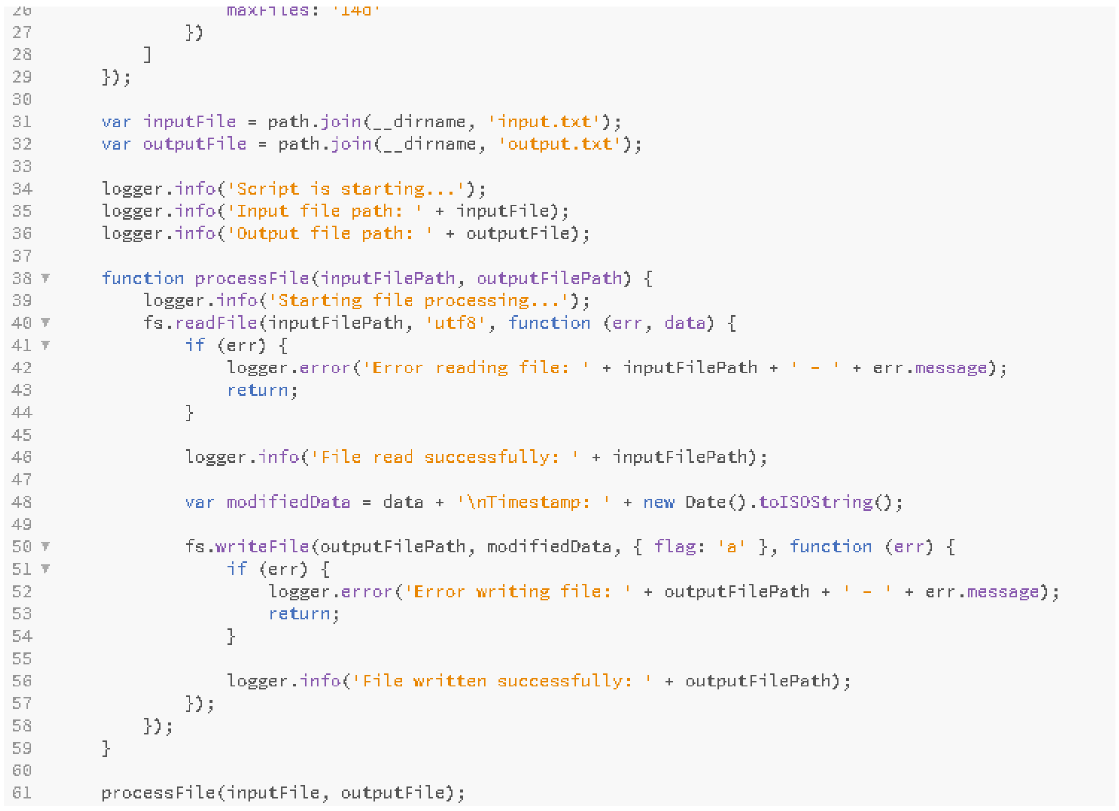
->NodeMenu.js script inside NodeExample folder that reads the contents of a file asynchronously, modifies the contents by appending a timestamp and writes updated to a new file.



```javascript
/*jslint node: true */
/*jslint nomen: true */

(function () {
    'use strict';

    var fs = require('fs');
    var path = require('path');
    var winston = require('winston');
    var DailyRotateFile = require('winston-daily-rotate-file');

    var logger = winston.createLogger({
        level: 'info',
        format: winston.format.combine(
            winston.format.timestamp(),
            winston.format.printf(info => {
                return `${info.timestamp} [${info.level.toUpperCase()}]: ${info.message}`;
            })
        ),
        transports: [
            new winston.transports.Console(),
            new DailyRotateFile({
                filename: 'logs/application-%DATE%.log',
                datePattern: 'YYYY-MM-DD',
                maxSize: '20m',
                maxFiles: '14d'
            })
        ]
    });

    var inputFile = path.join(__dirname, 'input.txt');
    var outputFile = path.join(__dirname, 'output.txt');

    logger.info('Script is starting...');
    logger.info('Input file path: ' + inputFile);
    logger.info('Output file path: ' + outputFile);

    function processFile(inputFilePath, outputFilePath) {
        logger.info('Starting file processing...');
        fs.readFile(inputFilePath, 'utf8', function (err, data) {
            if (err) {
                logger.error('Error reading file: ' + inputFilePath + ' - ' + err.message);
                return;
            }

            logger.info('File read successfully: ' + inputFilePath);

            var modifiedData = data + '\nTimestamp: ' + new Date().toISOString();

            fs.writeFile(outputFilePath, modifiedData, { flag: 'a' }, function (err) {
                if (err) {
                    logger.error('Error writing file: ' + outputFilePath + ' - ' + err.message);
                    return;
                }

                logger.info('File written successfully: ' + outputFilePath);
            });
        });
    }

    processFile(inputFile, outputFile);
```



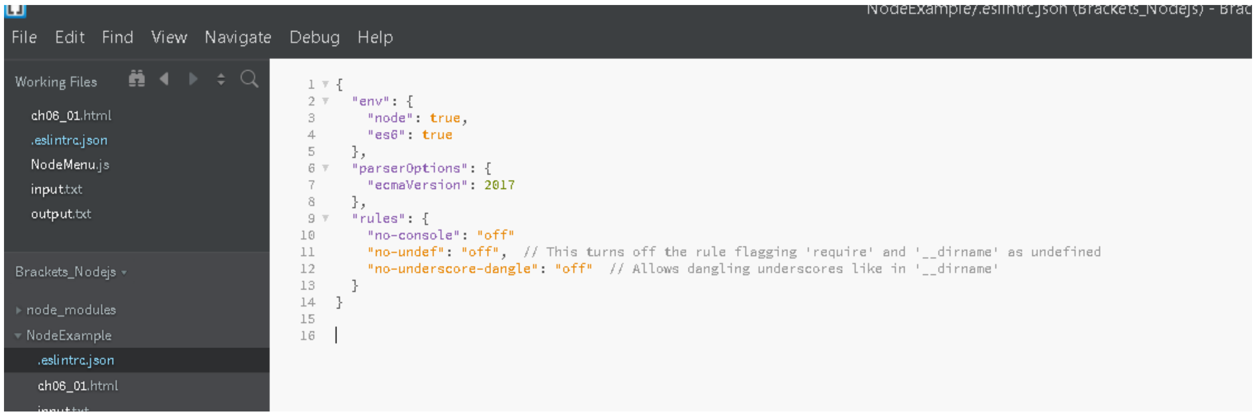
```
/*jslint node: true */
/*jslint nomen: true */

(function () {
    'use strict';

    var fs = require('fs');
    var path = require('path');
    var winston = require('winston');
    var DailyRotateFile = require('winston-daily-rotate-file');

    var logger = winston.createLogger({
        level: 'info',
        format: winston.format.combine(
            winston.format.timestamp(),
            winston.format.printf(info => {
                return `${info.timestamp} [${info.level.toUpperCase()}]: ${info.message}`;
            })
        ),
        transports: [
            new winston.transports.Console(),
            new DailyRotateFile({
                filename: 'logs/application-%DATE%.log',
                datePattern: 'YYYY-MM-DD',
                maxSize: '20m',
                maxFiles: '14d'
            })
        ]
```



```
                maxFiles: '14d'
            })
        ]
```

```
    });

    var inputFile = path.join(__dirname, 'input.txt');
    var outputFile = path.join(__dirname, 'output.txt');

    logger.info('Script is starting...');
    logger.info('Input file path: ' + inputFile);
    logger.info('Output file path: ' + outputFile);

    function processFile(inputFilePath, outputFilePath) {
        logger.info('Starting file processing...');
        fs.readFile(inputFilePath, 'utf8', function (err, data) {
            if (err) {
                logger.error('Error reading file: ' + inputFilePath + ' - ' + err.message);
                return;
            }

            logger.info('File read successfully: ' + inputFilePath);

            var modifiedData = data + '\nTimestamp: ' + new Date().toISOString();

            fs.writeFile(outputFilePath, modifiedData, { flag: 'a' }, function (err) {
                if (err) {
                    logger.error('Error writing file: ' + outputFilePath + ' - ' + err.message);
                    return;
                }

                logger.info('File written successfully: ' + outputFilePath);
            });
        });
    }

    processFile(inputFile, outputFile);
```

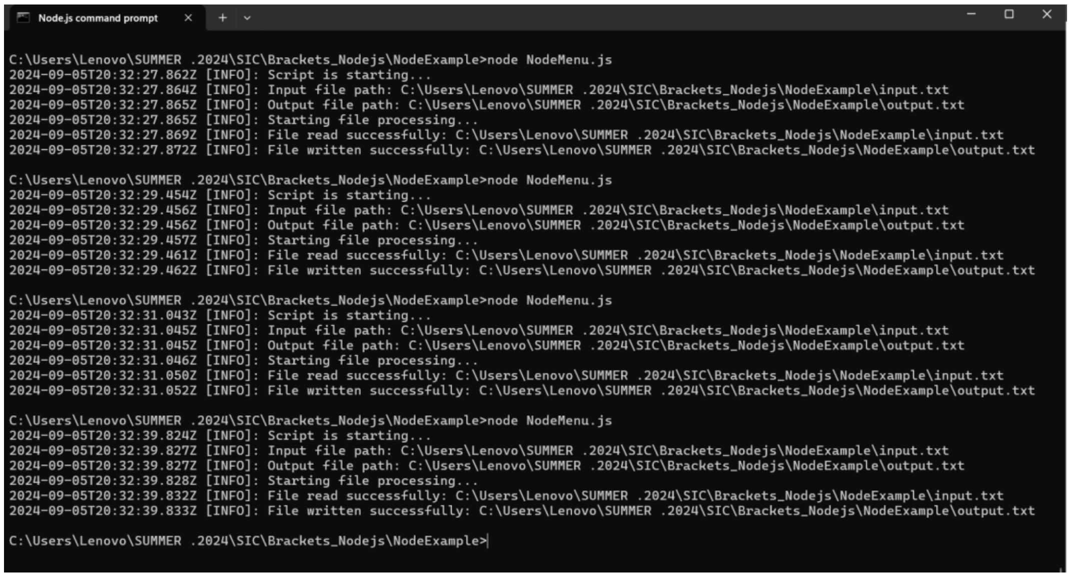
ESLint configuration file



```json
{
  "env": {
    "node": true,
    "es6": true
  },
  "parserOptions": {
    "ecmaVersion": 2017
  },
  "rules": {
    "no-console": "off",
    "no-undef": "off",  // This turns off the rule flagging 'require' and '__dirname' as undefined
    "no-underscore-dangle": "off"  // Allows dangling underscores like in '__dirname'
  }
}

```

Node.js Terminal

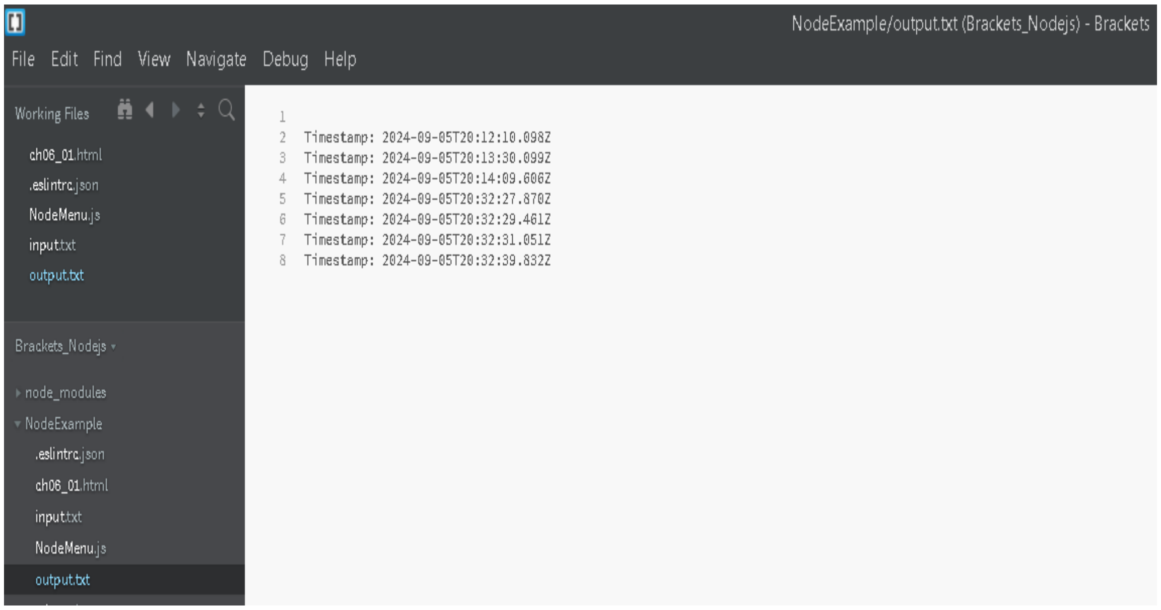
➔ Use Node NodeMenu.js to run file



```
C:\Users\Lenovo\SUMMER .2024\SIC\Brackets_Nodejs\NodeExample>node NodeMenu.js
2024-09-05T20:32:27.862Z [INFO]: Script is starting...
2024-09-05T20:32:27.864Z [INFO]: Input file path: C:\Users\Lenovo\SUMMER .2024\SIC\Brackets_Nodejs\NodeExample\input.txt
2024-09-05T20:32:27.865Z [INFO]: Output file path: C:\Users\Lenovo\SUMMER .2024\SIC\Brackets_Nodejs\NodeExample\output.txt
2024-09-05T20:32:27.865Z [INFO]: Starting file processing...
2024-09-05T20:32:27.869Z [INFO]: File read successfully: C:\Users\Lenovo\SUMMER .2024\SIC\Brackets_Nodejs\NodeExample\input.txt
2024-09-05T20:32:27.872Z [INFO]: File written successfully: C:\Users\Lenovo\SUMMER .2024\SIC\Brackets_Nodejs\NodeExample\output.txt

C:\Users\Lenovo\SUMMER .2024\SIC\Brackets_Nodejs\NodeExample>node NodeMenu.js
2024-09-05T20:32:29.454Z [INFO]: Script is starting...
2024-09-05T20:32:29.456Z [INFO]: Input file path: C:\Users\Lenovo\SUMMER .2024\SIC\Brackets_Nodejs\NodeExample\input.txt
2024-09-05T20:32:29.456Z [INFO]: Output file path: C:\Users\Lenovo\SUMMER .2024\SIC\Brackets_Nodejs\NodeExample\output.txt
2024-09-05T20:32:29.457Z [INFO]: Starting file processing...
2024-09-05T20:32:29.461Z [INFO]: File read successfully: C:\Users\Lenovo\SUMMER .2024\SIC\Brackets_Nodejs\NodeExample\input.txt
2024-09-05T20:32:29.462Z [INFO]: File written successfully: C:\Users\Lenovo\SUMMER .2024\SIC\Brackets_Nodejs\NodeExample\output.txt

C:\Users\Lenovo\SUMMER .2024\SIC\Brackets_Nodejs\NodeExample>node NodeMenu.js
2024-09-05T20:32:31.043Z [INFO]: Script is starting...
2024-09-05T20:32:31.045Z [INFO]: Input file path: C:\Users\Lenovo\SUMMER .2024\SIC\Brackets_Nodejs\NodeExample\input.txt
2024-09-05T20:32:31.045Z [INFO]: Output file path: C:\Users\Lenovo\SUMMER .2024\SIC\Brackets_Nodejs\NodeExample\output.txt
2024-09-05T20:32:31.046Z [INFO]: Starting file processing...
2024-09-05T20:32:31.050Z [INFO]: File read successfully: C:\Users\Lenovo\SUMMER .2024\SIC\Brackets_Nodejs\NodeExample\input.txt
2024-09-05T20:32:31.052Z [INFO]: File written successfully: C:\Users\Lenovo\SUMMER .2024\SIC\Brackets_Nodejs\NodeExample\output.txt

C:\Users\Lenovo\SUMMER .2024\SIC\Brackets_Nodejs\NodeExample>node NodeMenu.js
2024-09-05T20:32:39.824Z [INFO]: Script is starting...
2024-09-05T20:32:39.827Z [INFO]: Input file path: C:\Users\Lenovo\SUMMER .2024\SIC\Brackets_Nodejs\NodeExample\input.txt
2024-09-05T20:32:39.827Z [INFO]: Output file path: C:\Users\Lenovo\SUMMER .2024\SIC\Brackets_Nodejs\NodeExample\output.txt
2024-09-05T20:32:39.828Z [INFO]: Starting file processing...
2024-09-05T20:32:39.832Z [INFO]: File read successfully: C:\Users\Lenovo\SUMMER .2024\SIC\Brackets_Nodejs\NodeExample\input.txt
2024-09-05T20:32:39.833Z [INFO]: File written successfully: C:\Users\Lenovo\SUMMER .2024\SIC\Brackets_Nodejs\NodeExample\output.txt

C:\Users\Lenovo\SUMMER .2024\SIC\Brackets_Nodejs\NodeExample>
```

Output text file

➔ Output file after logs the process with daily log rotation



```

Timestamp: 2024-09-05T20:12:10.098Z
Timestamp: 2024-09-05T20:13:30.099Z
Timestamp: 2024-09-05T20:14:09.606Z
Timestamp: 2024-09-05T20:32:27.870Z
Timestamp: 2024-09-05T20:32:29.461Z
Timestamp: 2024-09-05T20:32:31.051Z
Timestamp: 2024-09-05T20:32:39.832Z
```