

## WPT ASSIGNMENT-(16-12-22)

PRAJWAL\_77

1. Create a module and import it in other programs
2. Install a module/package using npm
3. Write a program to create a new file and write some content to it in synchronous mode and read and display file contents on standard output in async mode
4. Build a simple Node.js web application serving both HTTP GET and POST methods

Q1


MYMATH.JS

```
export function sum(n1, n2) {  
    return n1 + n2;  
}  
  
export function mul(n1, n2) {  
    return n1 * n2;  
}
```

MAIN.JS

```
import { sum } from "./mymath.js";  
import { mul } from "./mymath.js";  
  
function main() {  
    let output = sum(1, 1);  
    console.log("Output", output);  
  
    let output1 = mul(2, 5);  
    console.log("Mul", output1);  
}  
  
main();
```

OUTPUT:-



The screenshot shows the VS Code interface with the 'OUTPUT' tab selected. The terminal displays the following output:

```
[Running] node "d:\DAC-WPT\node-project-demo\src\main.js"  
Output 2  
Mul 10  
  
[Done] exited with code=0 in 0.254 seconds
```

The status bar at the bottom indicates the file is 'Ln 13, Col 1 (230 selected)' and the encoding is 'UTF-8'.

Q3.Creating file

```

// writestream
// fs_write.js

import { open, write, close } from 'fs';

// specify the path to the file, and create a buffer with characters we want
// to write
let path = "E:\\\\DAC KOCHI\\hello1.txt";
let buffer = new Buffer('YOU KNOW? ....STYLE IS A WAY TO SAY WHO YOU ARE
WITHOUT SPEAKING...');

// open the file in writing mode, adding a callback function where we do the
// actual writing
open(path, 'w', function(err, fd) {
  if (err) {
    throw 'could not open file: ' + err;
  }

  // write the contents of the buffer, from position 0 to the end, to the
  // file descriptor returned in opening our file
  write(fd, buffer, 0, buffer.length, null, function(err) {
    if (err) throw 'error writing file: ' + err;
    close(fd, function() {
      console.log('wrote the file successfully');
    });
  });
});
});

```

## Reading file

```

// import {sum} from "./mymath.js"; // local module
import { readFile } from "node:fs/promises"; // node fs module

async function main() {
  console.log("READ FILE DEMO");

  let filepath = "E:\\\\DAC KOCHI\\hello1.txt";

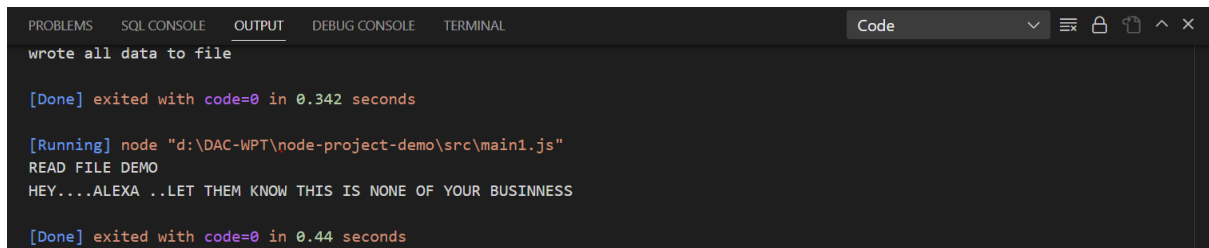
  // let output = readFile(filepath);
  // let output = await readFile(filepath);
  let output = await readFile(filepath, { encoding: "utf8" });

  console.log(output);
}

main();

```

## FINAL OUTPUT: -



```
PROBLEMS  SQL CONSOLE  OUTPUT  DEBUG CONSOLE  TERMINAL
wrote all data to file

[Done] exited with code=0 in 0.342 seconds

[Running] node "d:\DAC-WPT\node-project-demo\src\main1.js"
READ FILE DEMO
HEY...ALEXA ..LET THEM KNOW THIS IS NONE OF YOUR BUSINESS

[Done] exited with code=0 in 0.44 seconds
```

## Q4&2 .SIMPLE NODEJS-WEB APPLICATION -

```
import express from "express";
import { createConnection } from "mysql";
import bluebird from "bluebird";

const app = express();
app.use(express.json());
app.use(express.urlencoded({ extended: true }));
/* http://localhost:3000/messages */
app.get("/messages/", async (req, res) => {
  let connectUri = {
    host: "localhost",
    user: "root",
    password: "Prajwal@123",
    database: "cdac",
  };
  let connection = createConnection(connectUri);
  bluebird.promisifyAll(connection);

  await connection.connectAsync();

  // let sql = `SELECT * FROM message ORDER BY id DESC`;
  let sql = `SELECT * FROM message`;
  let results = await connection.queryAsync(sql);

  await connection.endAsync();

  res.json(results);
});

/* http://localhost:3000/message */
app.post("/message", async (req, res) => {
  let connectUri = {
```

```

    host: "localhost",
    user: "root",
    password: "Prajwal@123",
    database: "cdac",
  };
  let connection = createConnection(connectUri);
  bluebird.promisifyAll(connection);

  await connection.connectAsync();
  let message = req.body.message;
  let reply = req.body.reply;

  let sql = `INSERT INTO message (message, reply) VALUES ('${message}',
  ${reply})`;
  await connection.queryAsync(sql);

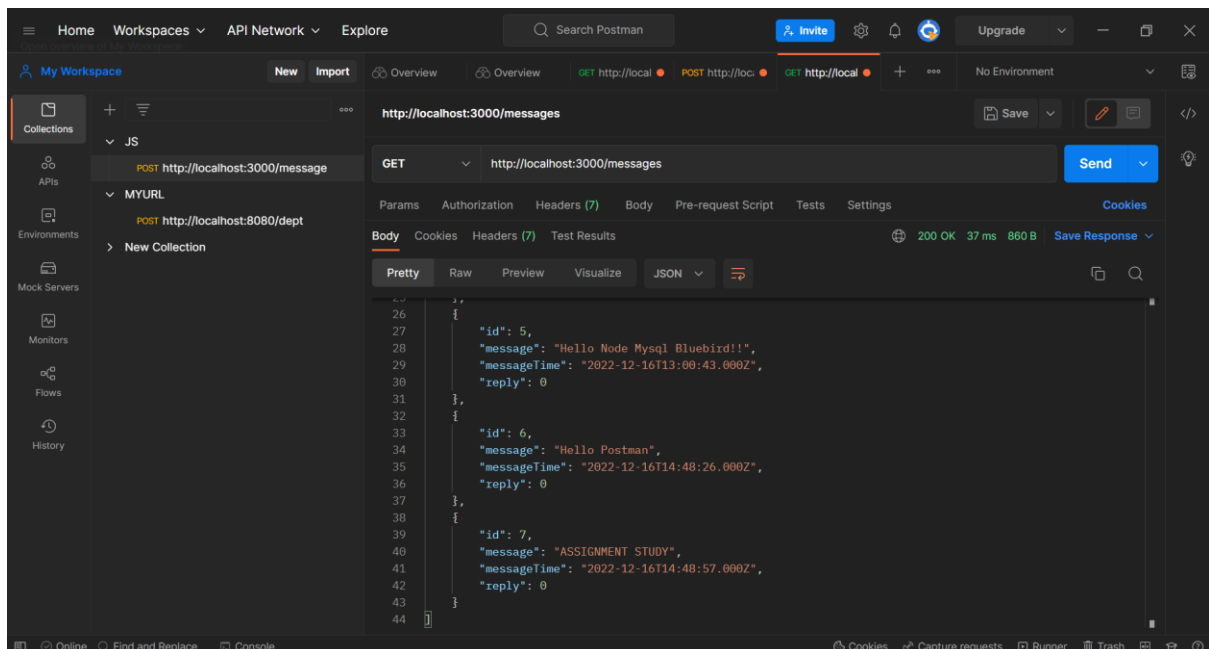
  connection.endAsync();

  res.json({ msg: "Record added!" });
});

app.listen(3000);

```

## GET-REQ -OUTPUT



## POST -REQ OUTPUT-

