

## WPT Assignment – 10

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) Create a module and import it in other programs

#### Mymath.js

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

//By default functions are private so to make it accessible we use export keyword.

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

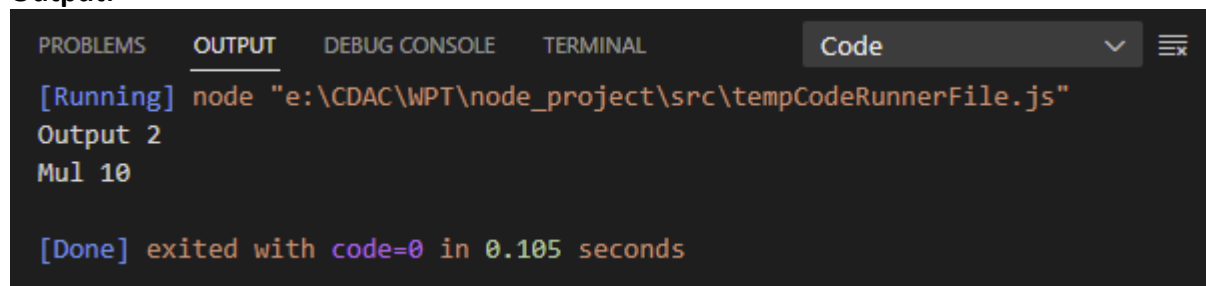
#### Main.js

```
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:



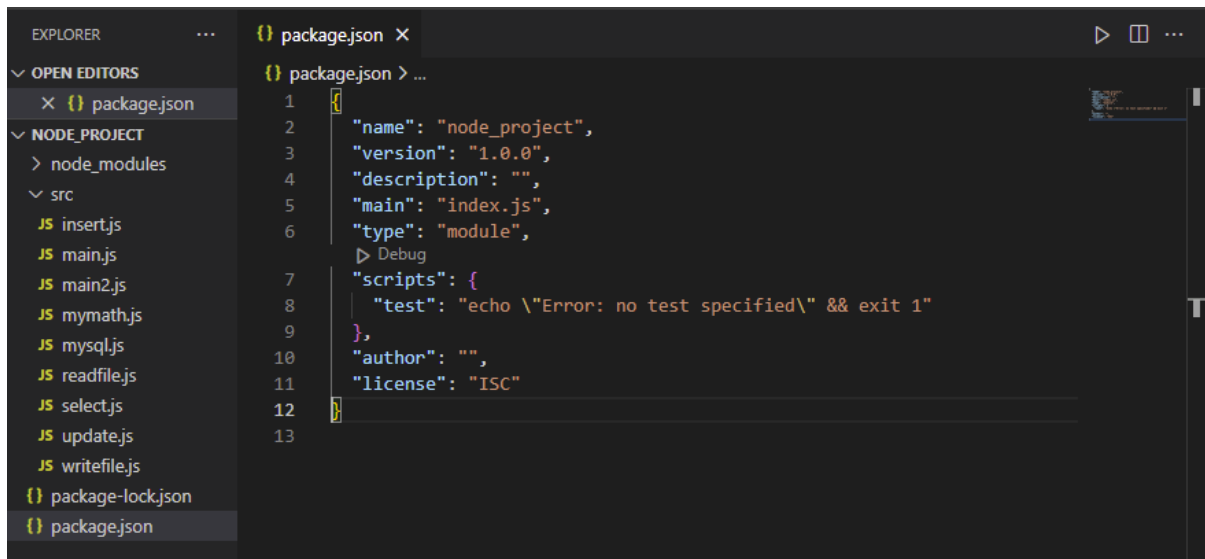
```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  Code  ⌵  ☰x  
[Running] node "e:\CDAC\WPT\node_project\src\tempCodeRunnerFile.js"  
Output 2  
Mul 10  
[Done] exited with code=0 in 0.105 seconds
```

## Q2) Install a module/package using npm.

### STEPS: -

- Create New Folder -> Example: node project
- Open cmd from newly created folder ^
  - Execute command -> **npm init** -> It will generate **package.json** file.
- Create src Folder -> Will put all the JS programs.
  - Main.js
  - Math.js
- Math Module
  - Export Sum Function
- Main Module
  - Import Sum Function
- Edit package.json file with following command -> **"type" : "module"**.

### Output:



The screenshot shows the Visual Studio Code interface. On the left, the Explorer sidebar displays the project structure: 'package.json' is open in the editor, and the 'src' folder contains several JavaScript files (insert.js, main.js, main2.js, mymath.js, mysql.js, readFile.js, select.js, update.js, writeFile.js). The main editor window shows the content of 'package.json' with the following JSON structure:

```
1 {  
2   "name": "node_project",  
3   "version": "1.0.0",  
4   "description": "",  
5   "main": "index.js",  
6   "type": "module",  
7   "scripts": {  
8     "test": "echo \\\"Error: no test specified\\\" && exit 1"  
9   },  
10  "author": "",  
11  "license": "ISC"  
12 }  
13
```

**Q 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.**

#### **Writefile.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:\\CDAC\\hello1.txt";
let buffer = new Buffer("Adding some content in a file...");

// 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");
    });
  });
});
```

#### **Readfile.js**

```
// 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:\\CDAC\\hello1.txt";
  //let output = readFile(filepath);
  //let output1 = await readFile(filepath);
  let output = await readFile(filepath, { encoding: "utf8" });

  console.log(output);
}
main();
```

#### **Output:**

```
[Running] node "e:\CDAC\WPT\node_project\src\readfile.js"
READ FILE DEMO
Adding some content in a file...
```

#### 4. Build a simple Node.js web application serving both HTTP GET and POST methods.

Install dependencies: -

- **npm install bluebird** -> To convert callback function into -> promise.
- **npm install express** -> To return the result to browser instead of terminal.
- **npm install mysql** -> To perform CRUD operations.

##### Get sql selectAll.js

```
import express from "express";
import { createConnection } from "mysql";
import bluebird from "bluebird";
const app = express();

/* GET / GIVE ME MESSAGE */

/* http://localhost:3000/messages */
app.get("/messages/", async (req, res) => {
  let connectUri = {
    host: "localhost",
    user: "root",
    password: "root@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);

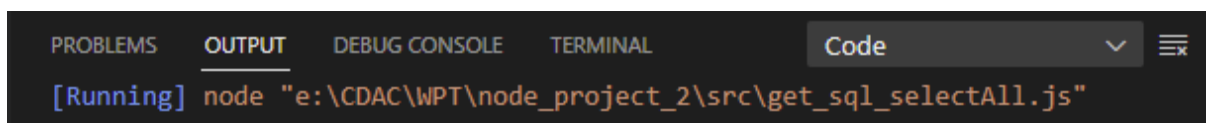
  await connection.endAsync();

  res.json(results);
});

app.listen(3000);
```

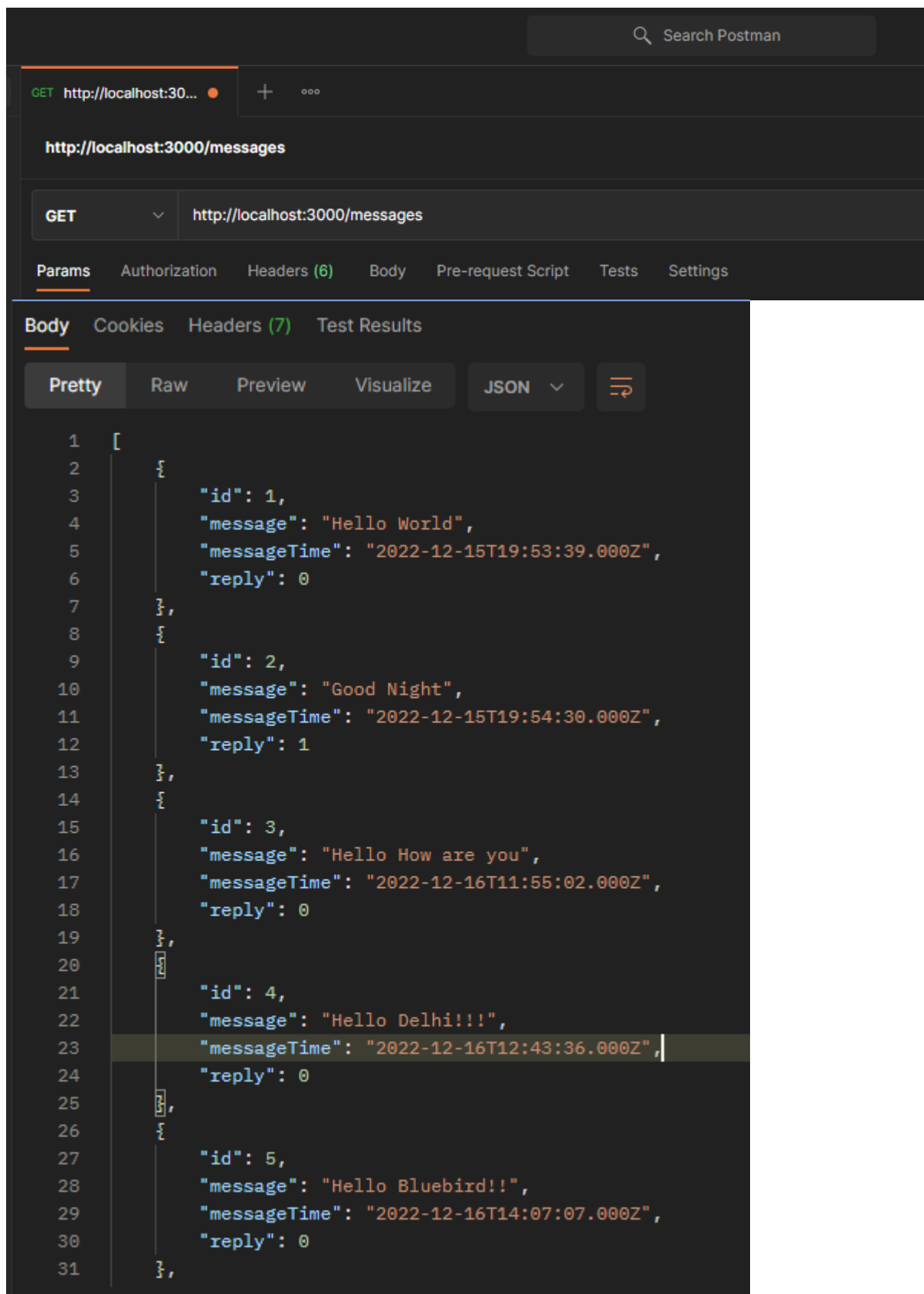
**Output:**

**Start the Nodejs. Server**

A screenshot of the Visual Studio Code interface. The top bar shows tabs for 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', and 'TERMINAL'. The 'TERMINAL' tab is active, displaying the command 'node "e:\CDAC\WPT\node\_project\_2\src\get\_sql\_selectAll.js"' in a blue font. The status bar at the bottom indicates the file is 'Code' and is running. The terminal output shows '[Running] node "e:\CDAC\WPT\node\_project\_2\src\get\_sql\_selectAll.js"' in a blue font.

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  Code
[Running] node "e:\CDAC\WPT\node_project_2\src\get_sql_selectAll.js"
```

Postman (HTTP Client / Browser that helps to make GET and POST request)



## Post\_sql\_insert.js

```
import express from "express";
import { createConnection } from "mysql";
import bluebird from "bluebird";
const app = express();

app.get("/message", (req, res) => {
  let message = { id: 1, message: "Hi", messageTime: new Date() };
  res.json(message);
});
//By default get method will be executed on browser.

/* POST / INSERT / ADD / CREATE NEW MESSAGE */
/* http://localhost:3000/message */
app.post("/message", async (req, res) => {
  let connectUri = {
    host: "localhost",
    user: "root",
    password: "root@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})`;
  let sql = `INSERT INTO message (message,reply) VALUES(?,?)`; //SQL injection safe  await

  connection.queryAsync(sql);

  connection.endAsync();

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

app.listen(3000);
```

### **Output:**

#### **Start the Nodejs. Server**

```
[Running] node "e:\CDAC\WPT\node_project_2\src\post_sql_insert.js"
```

## Postman (HTTP Client / Browser that helps to make GET and POST request)

- To make POST request.

