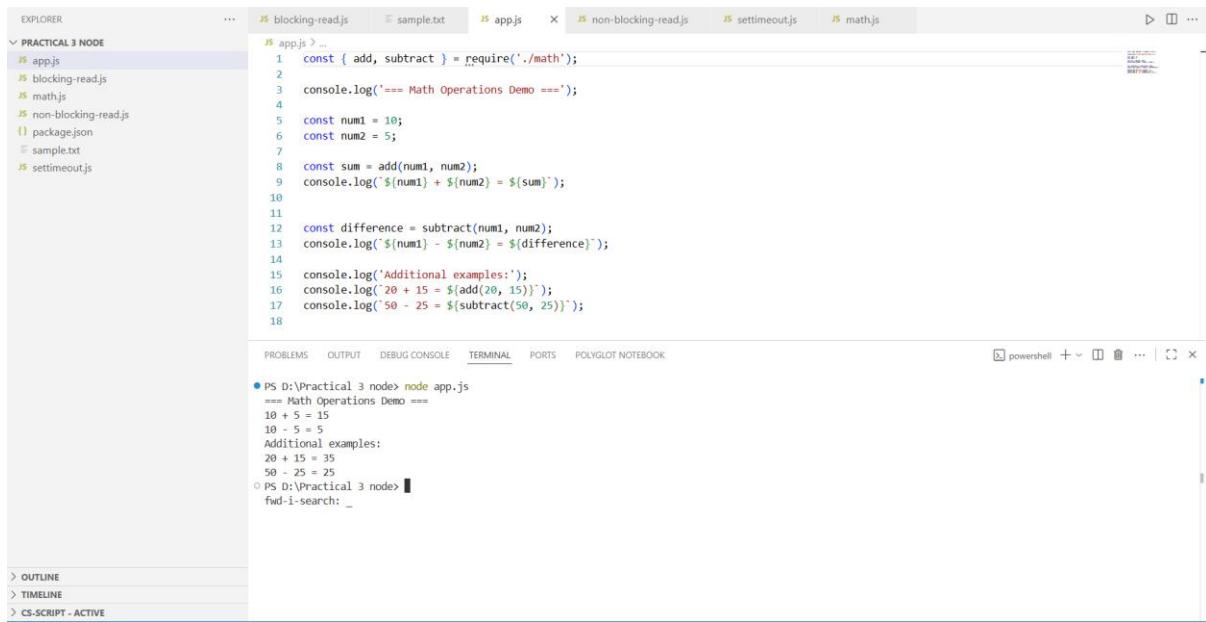


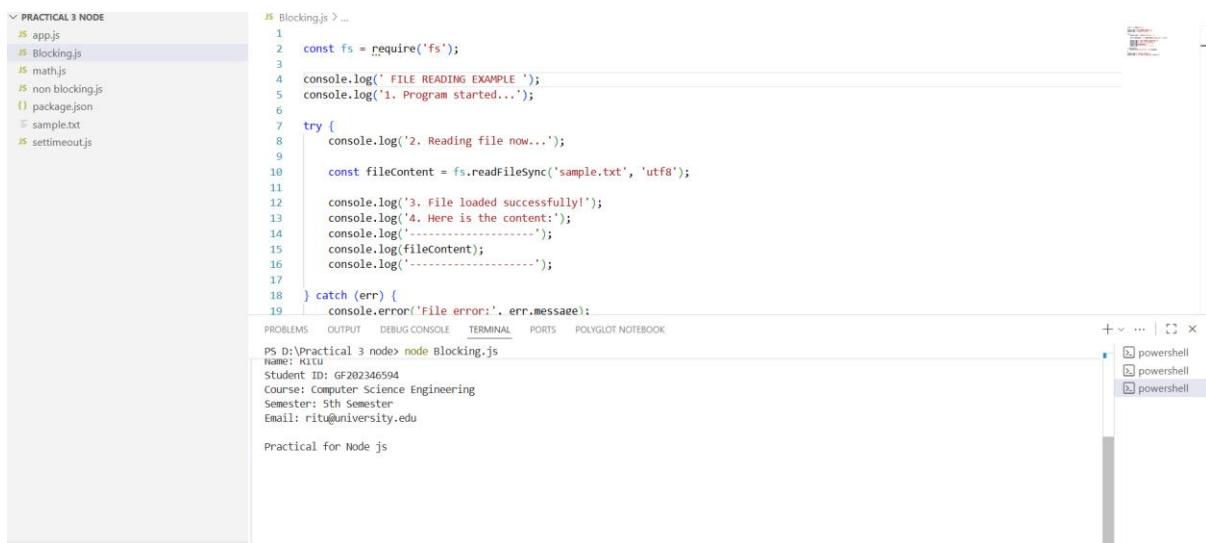
## Practical 3-



The screenshot shows the VS Code interface with the Explorer sidebar open, displaying files like app.js, blocking-read.js, math.js, non-blocking-read.js, package.json, sample.txt, and setTimeout.js. The terminal tab is active, showing the execution of app.js and its output:

```
PS D:\Practical 3 node> node app.js
== Math Operations Demo ==
10 + 5 = 15
10 - 5 = 5
Additional examples:
20 + 15 = 35
50 - 25 = 25
PS D:\Practical 3 node> fwd-i-search: _
```

The bottom left of the interface shows the Outline, Timeline, and CS-SCRIPT - ACTIVE sections.



The screenshot shows the VS Code interface with the Explorer sidebar open, displaying files like app.js, Blocking.js, math.js, non-blocking.js, package.json, sample.txt, and setTimeout.js. The terminal tab is active, showing the execution of Blocking.js and its output:

```
PS D:\Practical 3 node> node Blocking.js
FILE READING EXAMPLE
1. Program started...
2. Reading file now...
3. File loaded successfully!
4. Here is the content:
-----
console.log('-----');
console.log('-----');
```

The bottom right of the interface shows a list of open terminals: powershell, powershell, and powershell.

**PRACTICAL 3 NODE**

```

JS nonblocking.js
JS app.js
JS Blocking.js
JS math.js
JS nonblocking.js
{} package.json
sample.txt
JS setTimeout.js

```

```

JS nonblocking.js > ...
1 const fs = require('fs');
2
3 console.log(' ASYNC FILE READ EXAMPLE');
4 console.log('1. Program initialization...');

5
6 console.log('2. Starting file read operation (ASYNC)...');

7
8 fs.readFile(' sample.txt', 'utf8', (err, fileContent) => {
9     if (err) {
10         console.error('File reading failed:', err.message);
11         return;
12     }
13
14     console.log('4. File loaded successfully! (Callback executed)');
15     console.log('5. Content below:');
16     console.log('=====');
17     console.log(fileContent);
18     console.log('=====');
19     console.log('6. Async operation completed!');

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS POLYGLOT NOTEBOOK

PS D:\Practical 3 node> node nonblocking.js

- Starting file read operation (ASYNC)...
- This executes RIGHT AWAY after file read starts
- Main thread execution done
- Note: File content will appear after this line
- File loaded successfully! (Callback executed)
- Content below:

```

=====
Student Information System
Name: Ritu
Student ID: GF202346594
Course: Computer Science Engineering
Semester: 5th Semester
Email: ritu@university.edu

```

Practical for Node is

**PRACTICAL 3 NODE**

```

JS app.js
JS Blocking.js
JS math.js
JS nonblocking.js
JS p3.js
{} package.json
sample.txt

```

```

JS p3.js > loadPersonInfo > setTimeout() callback > personInfo > emailAddress
1
2 function loadPersonInfo() {
3     console.log('Loading person information...');
4
5     setTimeout(() => {
6         const personInfo = {
7             fullName: 'Ritu Singh',
8             emailAddress: 'Ritu99@gmail.com'
9         };
10
11         console.log('Information loaded successfully!');
12         console.log('Person Details:', personInfo);
13     }, 2000);
14 }
15
16 loadPersonInfo();
17
18

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS POLYGLOT NOTEBOOK

PS D:\Practical 3 node> node nonblocking.js

Practical for Node js

```

=====
6. Async operation completed!
9. Background tasks can run while file loads!
● PS D:\Practical 3 node> node p3.js
Loading person information...
Information loaded successfully!
Person Details: { fullName: 'Ritu Singh', emailAddress: 'Ritu99@gmail.com' }

```

> OUTLINE  
> TIMELINE  
CS-SCRIPT - ACTIVE