Practical-1

Aim: Create a "Hello World" program using ES6 arrow function in react.

Code:

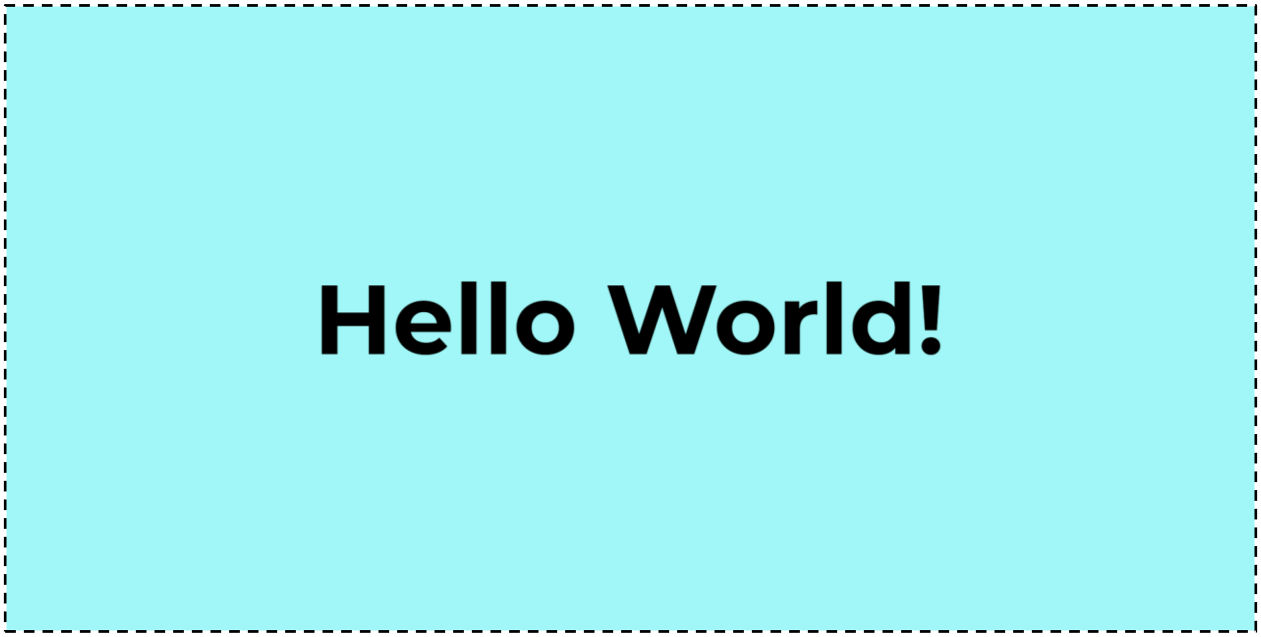
App.js

```
import './App.css';
const App = () => {
  return (
  <h1>Hello World!</h1>
  );
};
export default App;
```

Index.js

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import './index.css';
import App from './App';

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
  <React.StrictMode>
    <App />
  </React.StrictMode>
);
```

Output:



Hello World!

Practical-2

Aim: a. Create a functional component to display an alert message onclick event in Reactjs.

b. Create a functional component to display id and name from a list of persons in Reactjs.

Code:

A.

App.js

```
import Alert from './Components/Alert';
import './App.css';
function App() {
  return (
    <div className="App">
      <Alert>
        Hello
      </Alert>
    </div>
  );
}
export default App;
```

Alert.js

```
import React from 'react';
const Alert = ({ message }) => {
    const onClickHandler = () => {
        alert("Alert Triggered")
    }
    return(
        <button className='btn' onClick={onClickHandler}>
            Create Alert
        </button>
    )
 };
export default Alert;
```

Index.js

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import './index.css';
import App from './App';

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
  <React.StrictMode>
    <App />
  </React.StrictMode>
);
```

B.

App.js

```
import Genshin from './Components/Genshin';
import './App.css';
function App() {
  const data = [
    {id:1,name: "Wanderer"},
    {id:2,name: "Xiao"},
    {id:3,name: "Venti"},
    {id:4,name: "Kazuha"},
    {id:5,name: "Xianyun"},
  ]
  return (
    <div className="App">
      <Genshin items={data}></Genshin>
    </div>
  );
}
export default App;
```

Genshin.js

```
import React from 'react';
import './Genshin.css'
```

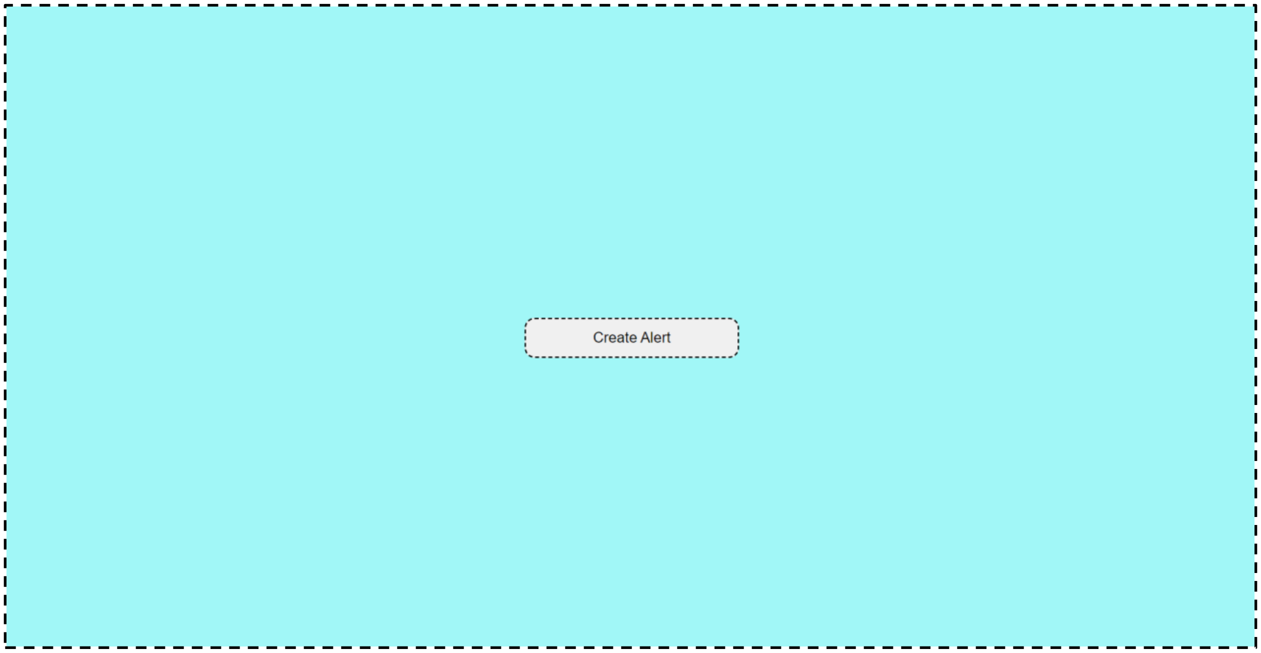
```
const Genshin = (props) => {
  return(
    <div>
      <h2>List of 5 Star Anemo User</h2>
      <table border={1}>

        <tr className='tbl'>
          <th className='data0'>Id</th>
          <th className='data0'>Name</th>
        </tr>

        <tr className='tbl'>
          <td>ID : {props.items[0].id}</td>
          <td>Name : {props.items[0].name}</td>
        </tr>

        <tr className='tbl'>
          <td>ID : {props.items[1].id}</td>
          <td>Name : {props.items[1].name}</td>
        </tr>

        <tr className='tbl'>
          <td>ID : {props.items[2].id}</td>
          <td>Name : {props.items[2].name}</td>
        </tr>

        <tr className='tbl'>
          <td>ID : {props.items[3].id}</td>
          <td>Name : {props.items[3].name}</td>
        </tr>

        <tr className='tbl'>
          <td>ID : {props.items[4].id}</td>
          <td>Name : {props.items[4].name}</td>
        </tr>

      </table>
    </div>
  );
}
export default Genshin;
```

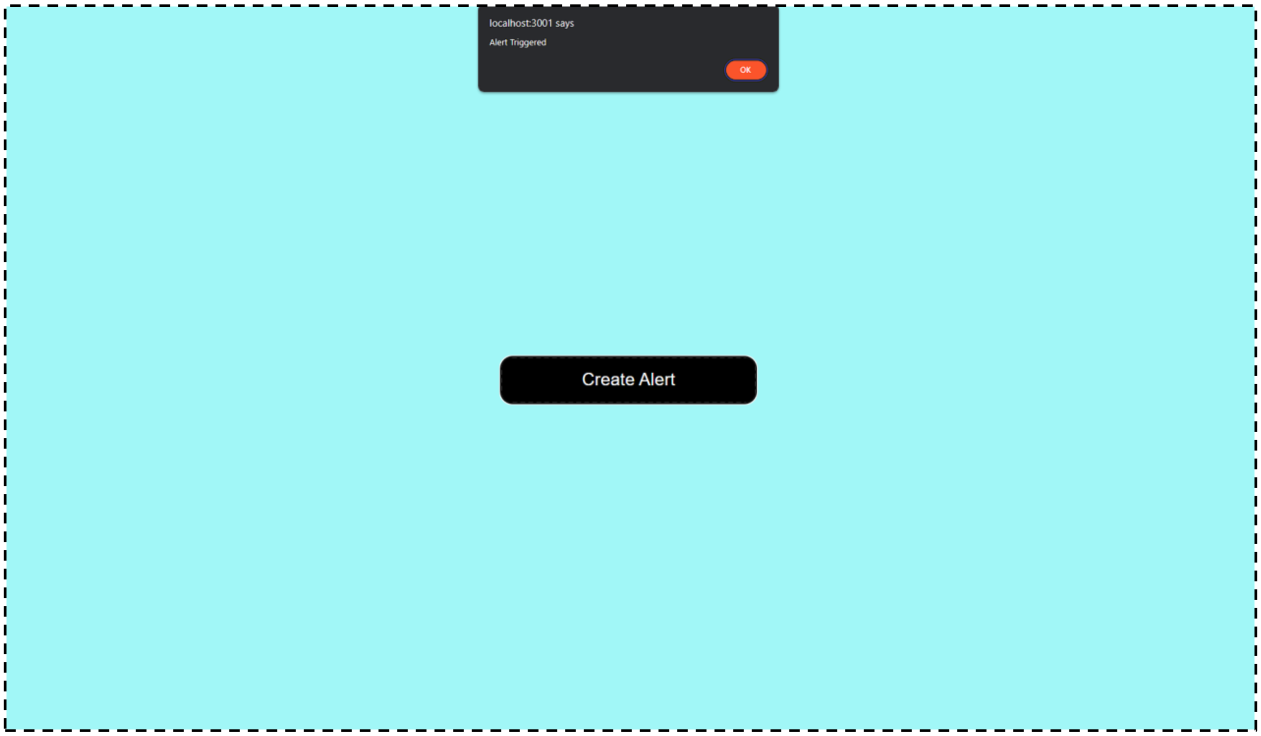
Index.js

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import './index.css';
import App from './App';

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
  <React.StrictMode>
    <App />
  </React.StrictMode>
);
```

Output:

A.

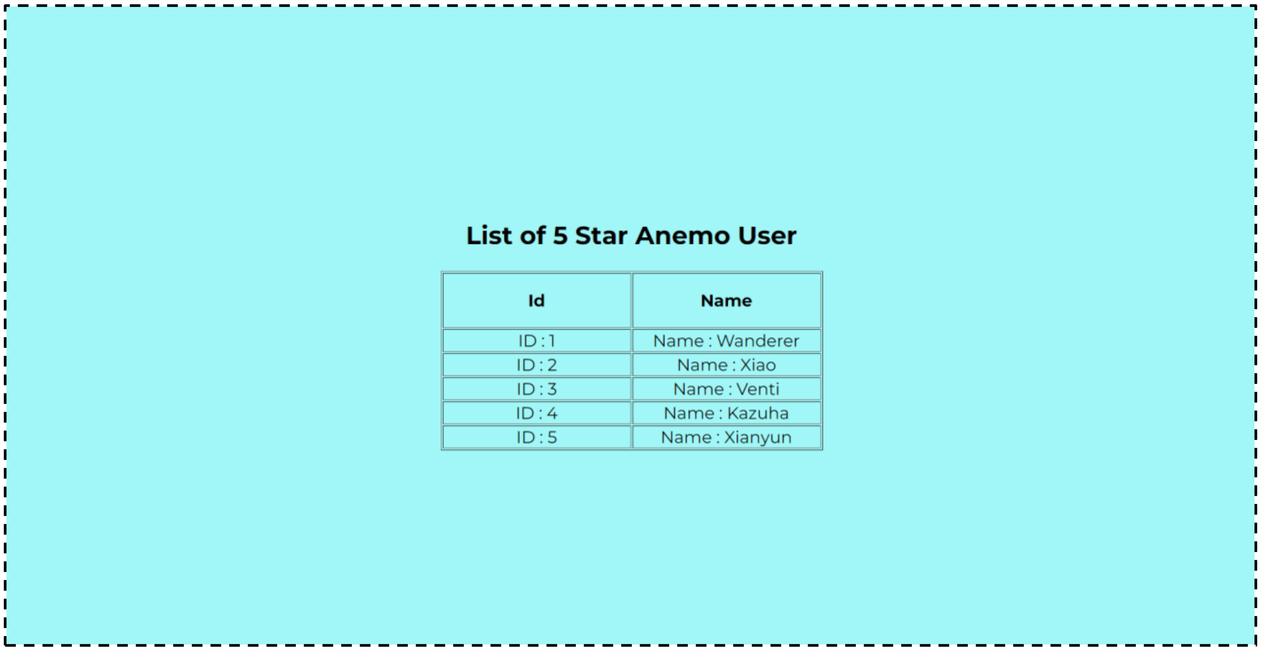




localhost:3001 says

Alert Triggered

B.



List of 5 Star Anemo User

Id	Name
ID : 1	Name : Wanderer
ID : 2	Name : Xiao
ID : 3	Name : Venti
ID : 4	Name : Kazuha
ID : 5	Name : Xianyun

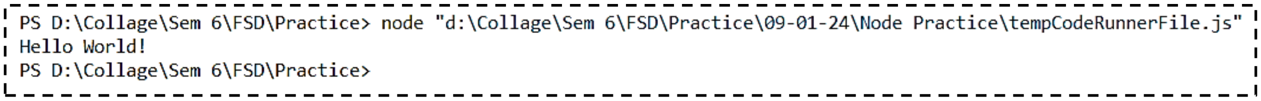
Practical-3

Aim: Implement “Hello World” program in Node.js.

Code:

```
const hello = "Hello World!";
console.log(hello);
```

Output:



```
PS D:\Collage\Sem 6\FSD\Practice> node "d:\Collage\Sem 6\FSD\Practice\09-01-24\Node Practice\tempCodeRunnerFile.js"
Hello World!
PS D:\Collage\Sem 6\FSD\Practice>
```

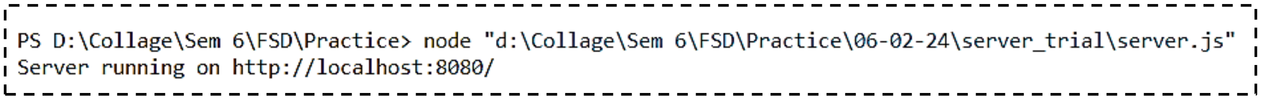
Practical-4

Aim: Create a simple web server using the HTTP module in Node.js.

Code:

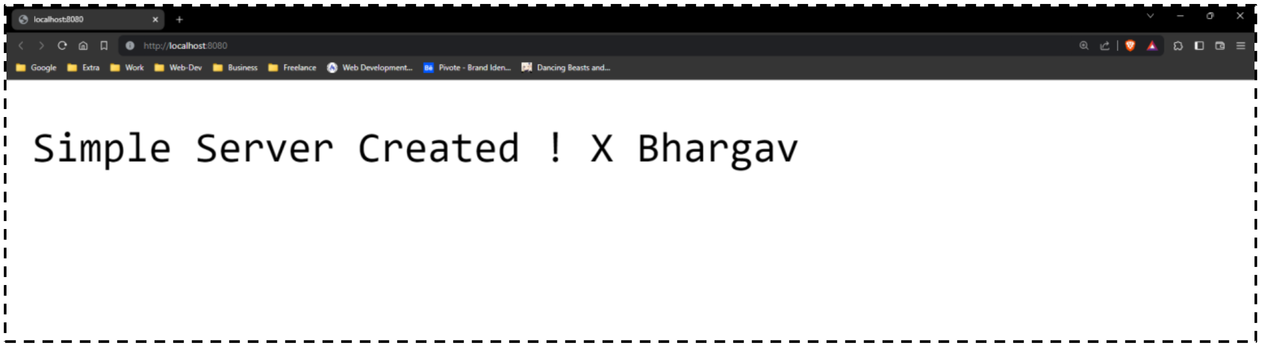
```
const http = require('http');

const server = http.createServer((req, res) => {
  res.writeHead(200, {'Content-Type': 'text/plain'});
  res.end('Simple Server Created ! X Bhargav);
});

const port = 8080;

// Start the server
server.listen(port, () => {
  console.log(`Server running on http://localhost:${port}/`);
});
```

Terminal :



```
PS D:\Collage\Sem 6\FSD\Practice> node "d:\Collage\Sem 6\FSD\Practice\06-02-24\server_trial\server.js"
Server running on http://localhost:8080/
```

Browser :



```
Simple Server Created ! X Bhargav
```

Practical-5

Aim:

A. Create a functional component to perform increment and decrement operation onclick event by using useState in Reactjs.

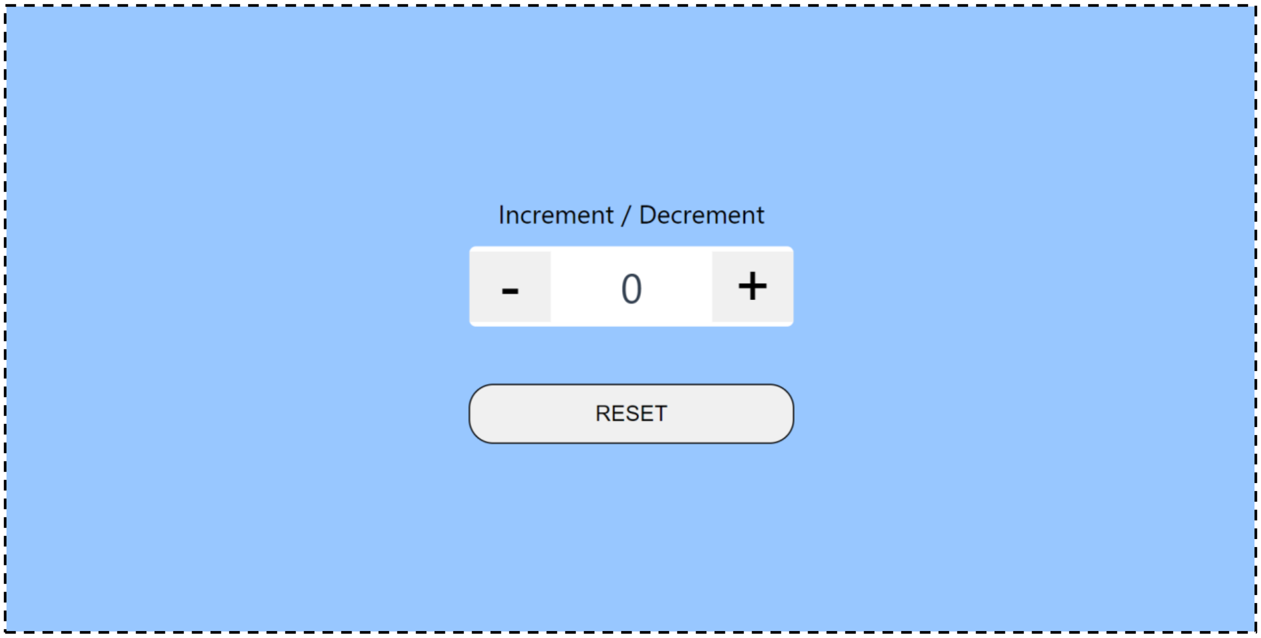
B.Create a class component to update state onclick event in Reactjs.

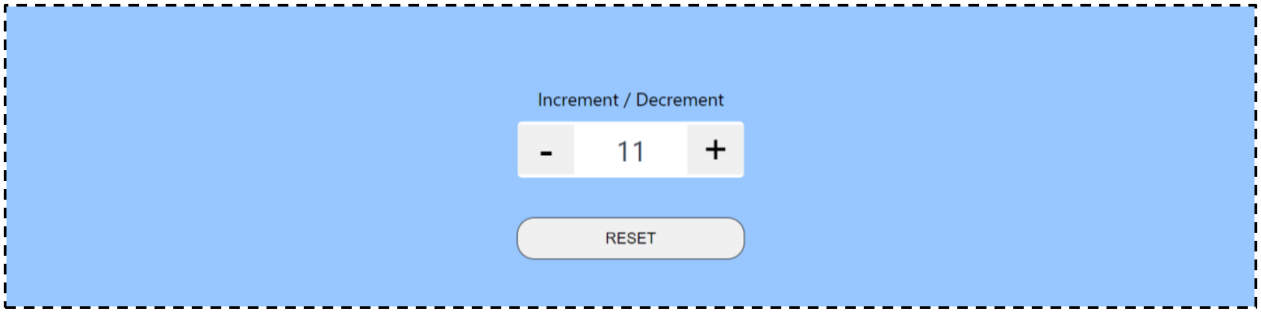
A.

Code:

```
import './App.css';
import React,{useState} from 'react';

function App() {
  const [count,setCount] = useState(0);

  function decreseHandler() {
    setCount(count-1)
  }
  function increseHandler() {
    setCount(count+1)
  }
  function resetHandler(){
    setCount(0);
  }
  return (
    <div className="container">
      <div className="inc_dec">Increment / Decrement </div>
      <div className="box">
          <button onClick={decreseHandler} className="dec">-</button>
          <div className="count_num" > {count} </div>
          <button onClick={increseHandler} className="inc" >+</button>
      </div>
      <button onClick={resetHandler} className='rst' > RESET </button>
    </div>
  );
}

export default App;
```

Output:



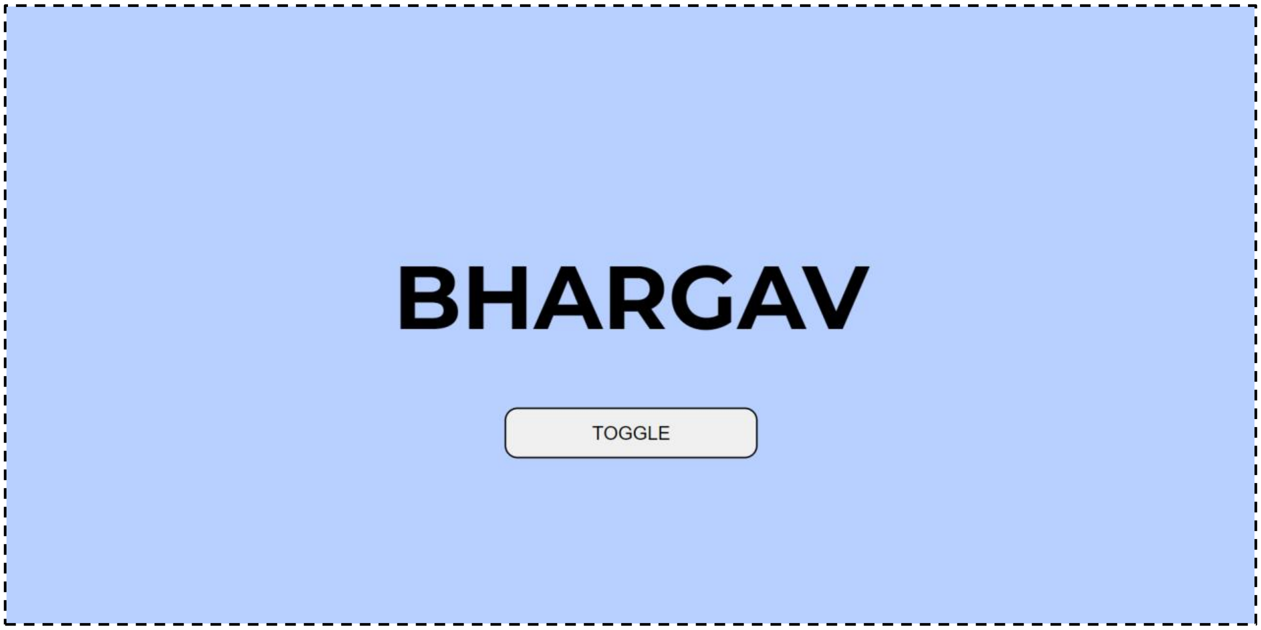


B.

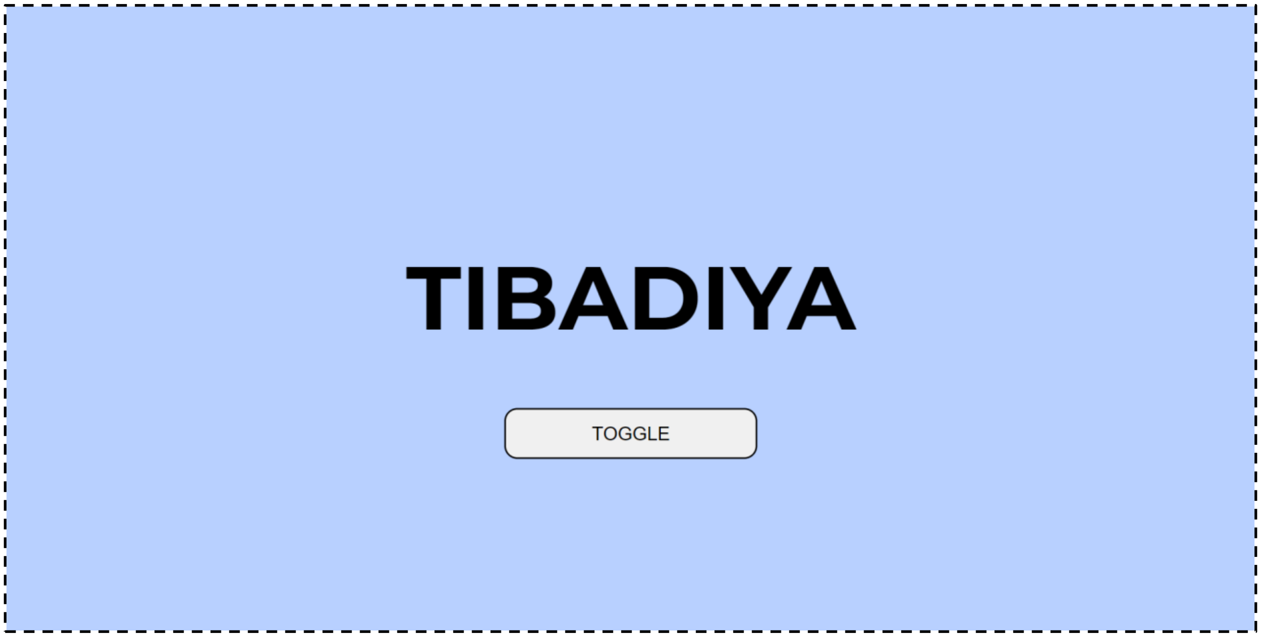
Code:

```
import { Component } from 'react';
import './App.css';

class App extends Component {
  constructor(props){
    super(props);
    this.state = {
      text : "BHARGAV",
    };
  }

  onClickHandler = () => {
    if(this.state.text === "BHARGAV"){
      this.setState({text:"TIBADIYA"});
    } else{
      this.setState({text:"BHARGAV"});
    }
  };

  render () {
    return(
      <div className='box'>
        <h1 className='txt'>{this.state.text}</h1>
        <button onClick={this.onClickHandler} className='btn'> TOGGLE </button>
      </div>
    )
  }

};

export default App;
```

Output:



BHARGAV



TIBADIYA

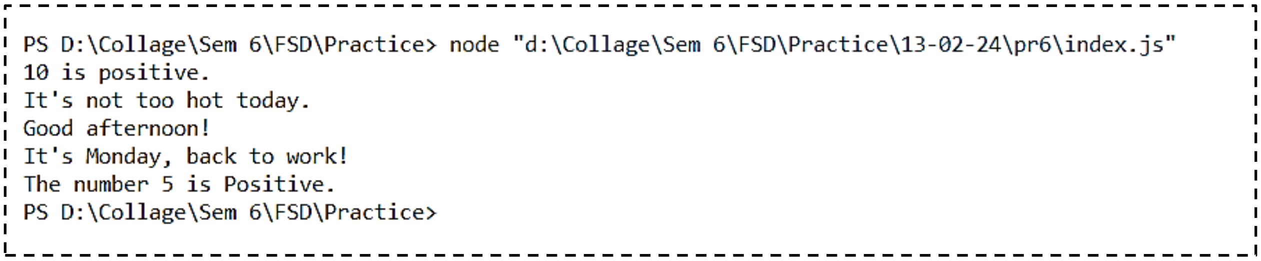
Practical-6

Aim: Write a Node.js program to demonstrate the use of various conditional statements.

Code:

```
// If statement
const num = 10;
if (num > 0) {
  console.log(`${num} is positive.`);
}

// If-else statement
const temperature = 25;
if (temperature >= 30) {
  console.log("It's a hot day!");
} else {
  console.log("It's not too hot today.");
}

// If-else if-else statement
const hour = 14;
if (hour < 12) {
  console.log("Good morning!");
} else if (hour >= 12 && hour < 18) {
  console.log("Good afternoon!");
} else {
  console.log("Good evening!");
}

// Switch statement
const day = 'Monday';
switch (day) {
  case 'Monday':
    console.log("It's Monday, back to work!");
    break;
  case 'Friday':
    console.log("It's Friday, time to relax!");
    break;
  default:
    console.log("It's a regular day.");
}

// Ternary operator
const x = 5;
const result = x > 0 ? 'Positive' : 'Negative or Zero';
console.log(`The number ${x} is ${result}.`);
```

Output:



```
PS D:\Collage\Sem 6\FSD\Practice> node "d:\Collage\Sem 6\FSD\Practice\13-02-24\pr6\index.js"
10 is positive.
It's not too hot today.
Good afternoon!
It's Monday, back to work!
The number 5 is Positive.
PS D:\Collage\Sem 6\FSD\Practice>
```

Practical-7

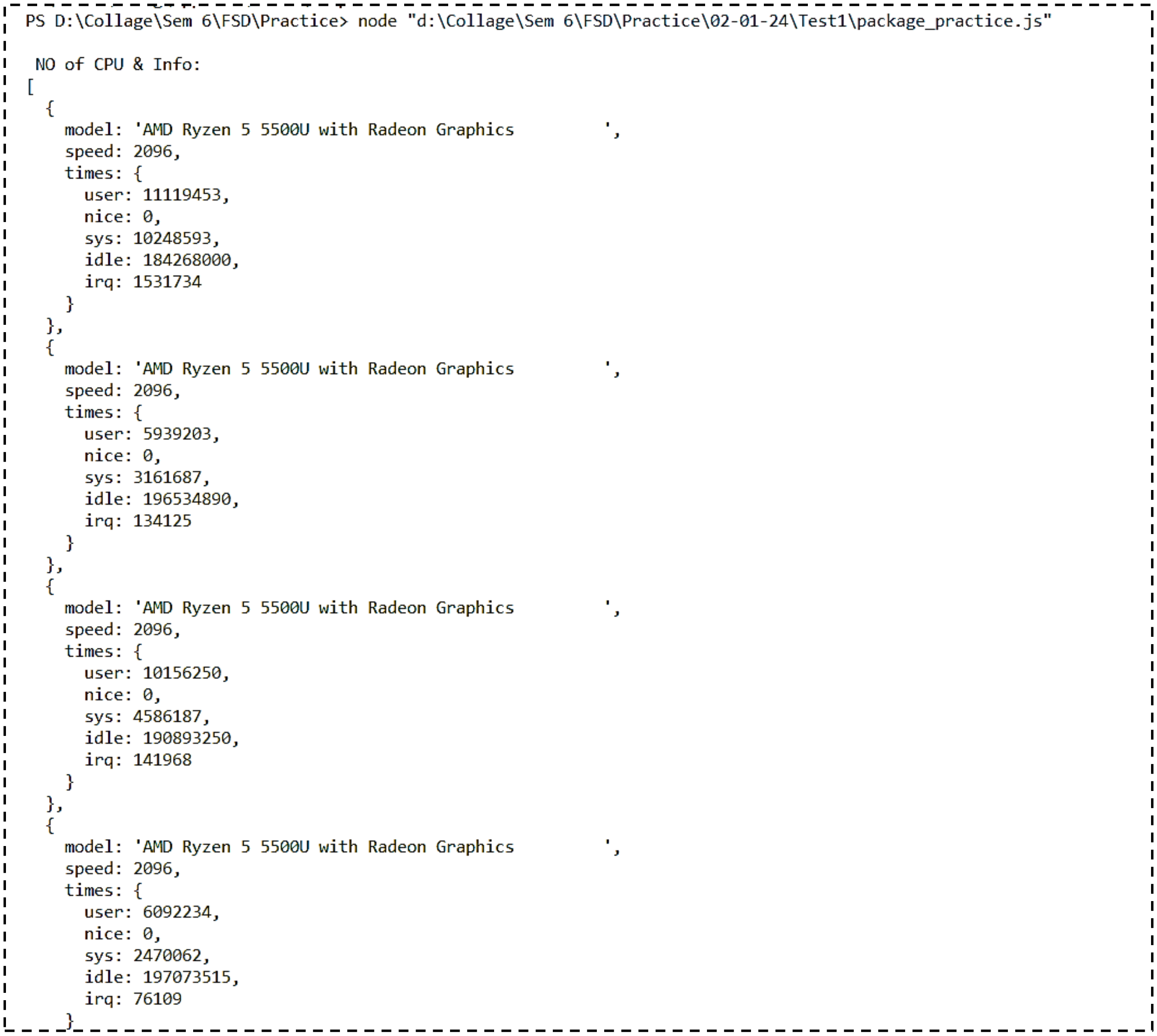
Aim: Write a Node.js program to demonstrate use of path, fs, os and util module.

OS

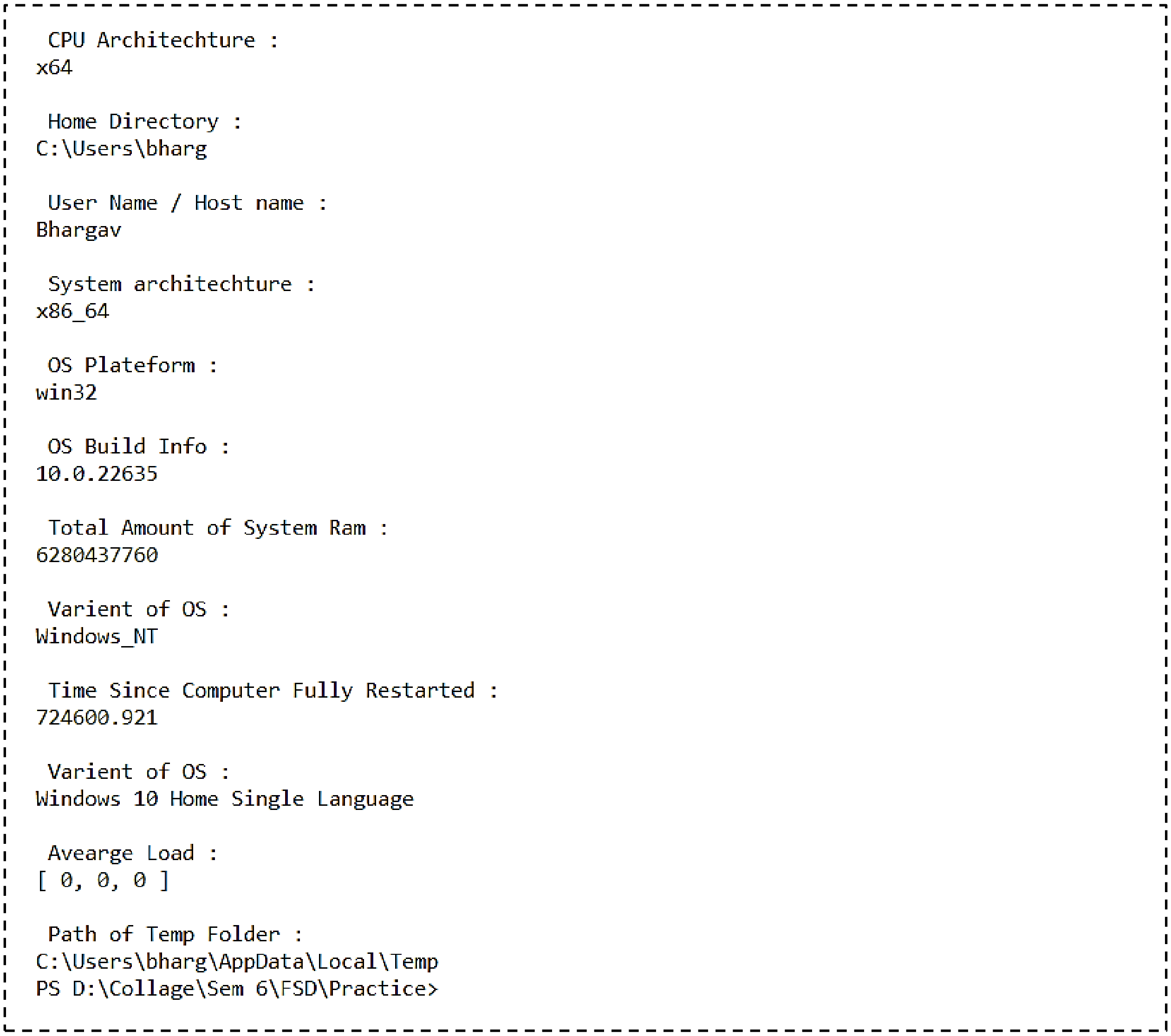
Code :

```
const a = require('node:os');

console.log("\n NO of CPU & Info:");
console.log(a.cpus());

console.log("\n CPU Architechture :");
console.log(a.arch());

console.log("\n Home Directory :");
console.log(a.homedir());

console.log("\n User Name / Host name :");
console.log(a.hostname());

console.log("\n System architechture :");
console.log(a.machine());

console.log("\n OS Plateform :");
console.log(a.platform());

console.log("\n OS Build Info :");
console.log(a.release());

console.log("\n Total Amount of System Ram :");
console.log(a.totalmem());

console.log("\n Varient of OS :");
console.log(a.type());

console.log("\n Time Since Computer Fully Restarted :");
console.log(a.uptime());

console.log("\n Varient of OS :");
console.log(a.version());

console.log("\n Avearge Load :");
console.log(a.loadavg());

console.log("\n Path of Temp Folder :");
console.log(a.tmpdir());
```

Output :



```
PS D:\Collage\Sem 6\FSD\Practice> node "d:\Collage\Sem 6\FSD\Practice\02-01-24\Test1\package_practice.js"

 NO of CPU & Info:
[
  {
    model: 'AMD Ryzen 5 5500U with Radeon Graphics         ',
    speed: 2096,
    times: {
      user: 11119453,
      nice: 0,
      sys: 10248593,
      idle: 184268000,
      irq: 1531734
    }
  },
  {
    model: 'AMD Ryzen 5 5500U with Radeon Graphics         ',
    speed: 2096,
    times: {
      user: 5939203,
      nice: 0,
      sys: 3161687,
      idle: 196534890,
      irq: 134125
    }
  },
  {
    model: 'AMD Ryzen 5 5500U with Radeon Graphics         ',
    speed: 2096,
    times: {
      user: 10156250,
      nice: 0,
      sys: 4586187,
      idle: 190893250,
      irq: 141968
    }
  },
  {
    model: 'AMD Ryzen 5 5500U with Radeon Graphics         ',
    speed: 2096,
    times: {
      user: 6092234,
      nice: 0,
      sys: 2470062,
      idle: 197073515,
      irq: 76109
    }
```



```
 CPU Architechture :
x64

 Home Directory :
C:\Users\bharg

 User Name / Host name :
Bhargav

 System architechture :
x86_64

 OS Plateform :
win32

 OS Build Info :
10.0.22635

 Total Amount of System Ram :
6280437760

 Varient of OS :
Windows_NT

 Time Since Computer Fully Restarted :
724600.921

 Varient of OS :
Windows 10 Home Single Language

 Avearge Load :
[ 0, 0, 0 ]

 Path of Temp Folder :
C:\Users\bharg\AppData\Local\Temp
PS D:\Collage\Sem 6\FSD\Practice>
```

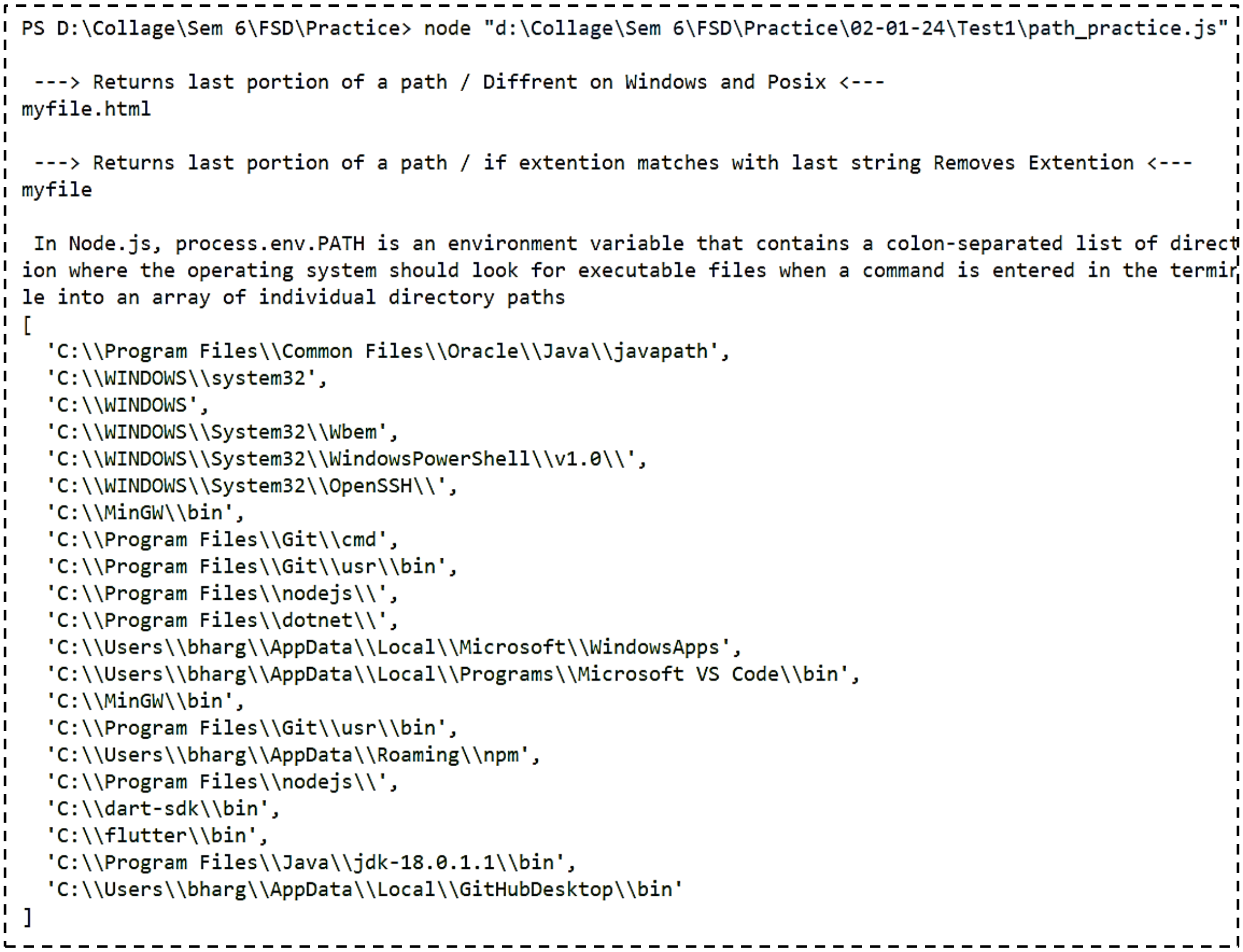
Path :

Code :

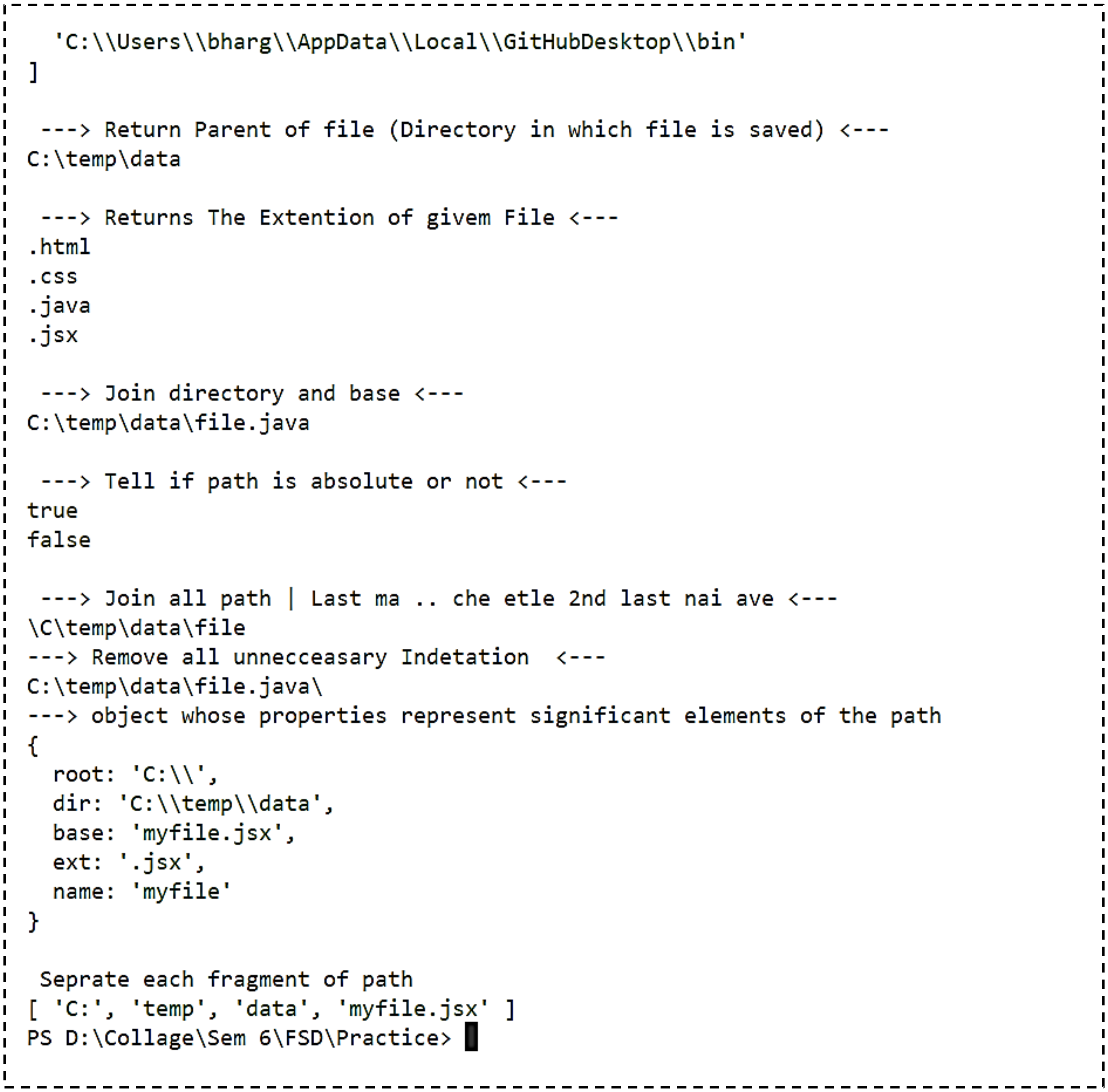
```
const path = require('node:path');

console.log("\n ---> Returns last portion of a path / Diffrent on Windows and Posix <---");
var result = path.basename('C:\\temp\\myfile.html');
console.log(result);

console.log("\n ---> Returns last portion of a path / if extention matches with last string Removes Extention <---");
result = path.basename('C:\\temp\\myfile.html','.html');
console.log(result);

console.log("\n In Node.js, process.env.PATH is an environment variable that contains a colon-separated list of directories. Each directory in this list represents a location where the operating system should look for executable files when a command is entered in the terminal. >>  code is splitting the PATH environment variable into an array of individual directory paths");
result = process.env.PATH.split(path.delimiter);
console.log(result);

console.log("\n ---> Return Parent of file (Directory in which file is saved) <---")
result = path.dirname('C:\\temp\\data\\myfile.html');
console.log(result)

console.log("\n ---> Returns The Extention of givem File <---")

console.log(path.extname('C:\\temp\\data\\myfile.html'))
console.log(path.extname('C:\\temp\\data\\myfile.css'))
console.log(path.extname('C:\\temp\\data\\myfile.java'))
console.log(path.extname('C:\\temp\\data\\myfile.jsx'))


console.log("\n ---> Join directory and base <---")
result = path.format({
  dir: 'C:\\temp\\data',
  base: 'file.java',
});
console.log(result);


console.log("\n ---> Tell if path is absolute or not <---  ")
result = path.isAbsolute('//server');
console.log(result);
result = path.isAbsolute('test1//arrow_function.js');
```

```
console.log(result);

console.log("\n ---> Join all path | Last ma .. che etle 2nd last nai ave <--- ")
result = path.join('/C', 'temp', 'data/file', 'java','..');
console.log(result);

console.log("---> Remove all unnecceasary Indetation  <---");
result = path.normalize('C:\\temp\\data\\\\\\\\file.java\\');
console.log(result);

console.log("---> object whose properties represent significant elements of the path")
result = path.parse('C:\\temp\\data\\myfile.jsx');
console.log(result);

console.log("\n Seprate each fragment of path")
result = 'C:\\temp\\data\\myfile.jsx'.split(path.sep);
console.log(result);
```

Output :



```
PS D:\Collage\Sem 6\FSD\Practice> node "d:\Collage\Sem 6\FSD\Practice\02-01-24\Test1\path_practice.js"

 ---> Returns last portion of a path / Diffrent on Windows and Posix <---
myfile.html

 ---> Returns last portion of a path / if extention matches with last string Removes Extention <---
myfile

 In Node.js, process.env.PATH is an environment variable that contains a colon-separated list of direct
ion where the operating system should look for executable files when a command is entered in the termin
le into an array of individual directory paths
[
  'C:\\Program Files\\Common Files\\Oracle\\Java\\javapath',
  'C:\\WINDOWS\\system32',
  'C:\\WINDOWS',
  'C:\\WINDOWS\\System32\\Wbem',
  'C:\\WINDOWS\\System32\\WindowsPowerShell\\v1.0\\',
  'C:\\WINDOWS\\System32\\OpenSSH\\',
  'C:\\MinGW\\bin',
  'C:\\Program Files\\Git\\cmd',
  'C:\\Program Files\\Git\\usr\\bin',
  'C:\\Program Files\\nodejs\\',
  'C:\\Program Files\\dotnet\\',
  'C:\\Users\\bharg\\AppData\\Local\\Microsoft\\WindowsApps',
  'C:\\Users\\bharg\\AppData\\Local\\Programs\\Microsoft VS Code\\bin',
  'C:\\MinGW\\bin',
  'C:\\Program Files\\Git\\usr\\bin',
  'C:\\Users\\bharg\\AppData\\Roaming\\npm',
  'C:\\Program Files\\nodejs\\',
  'C:\\dart-sdk\\bin',
  'C:\\flutter\\bin',
  'C:\\Program Files\\Java\\jdk-18.0.1.1\\bin',
  'C:\\Users\\bharg\\AppData\\Local\\GitHubDesktop\\bin'
]
```



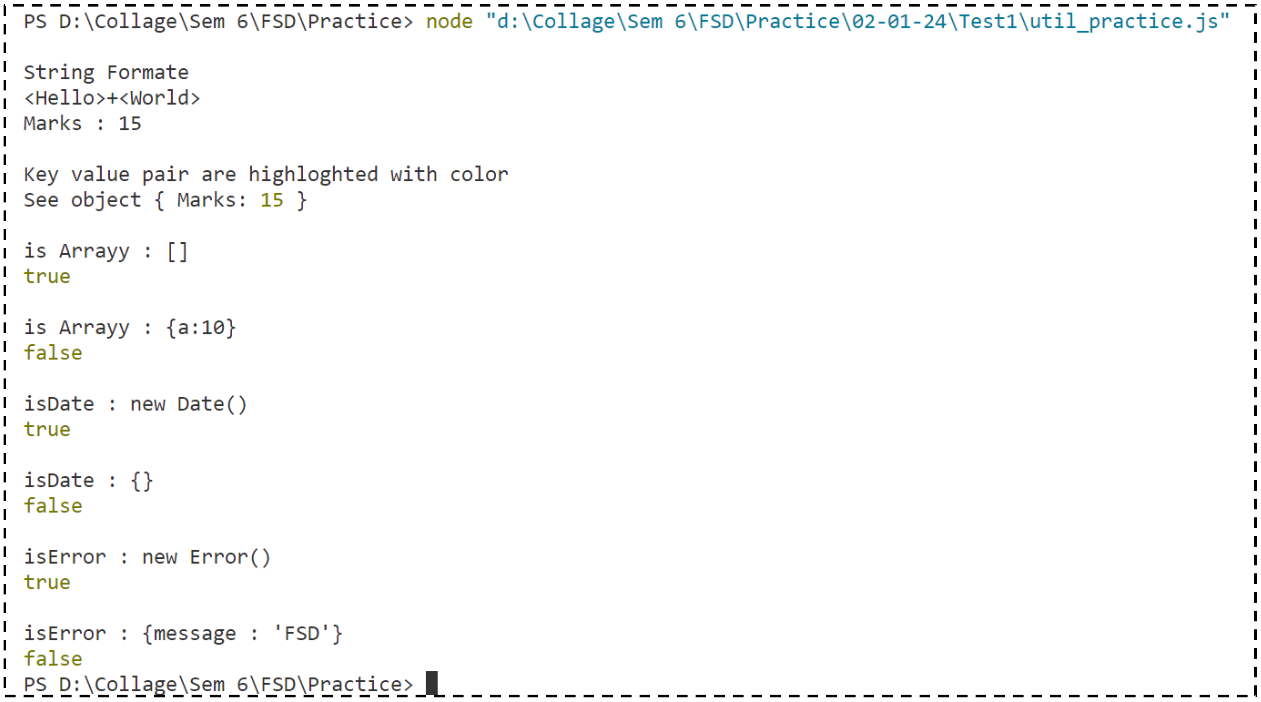
```
  'C:\\Users\\bharg\\AppData\\Local\\GitHubDesktop\\bin'
]

 ---> Return Parent of file (Directory in which file is saved) <---
C:\temp\data

 ---> Returns The Extention of givem File <---
.html
.css
.java
.jsx

 ---> Join directory and base <---
C:\temp\data\file.java

 ---> Tell if path is absolute or not <---
true
false

 ---> Join all path | Last ma .. che etle 2nd last nai ave <---
\C\temp\data\file
---> Remove all unnecceasary Indetation  <---
C:\temp\data\file.java\
---> object whose properties represent significant elements of the path
{
  root: 'C:\\',
  dir: 'C:\\temp\\data',
  base: 'myfile.jsx',
  ext: '.jsx',
  name: 'myfile'
}

 Seprate each fragment of path
[ 'C:', 'temp', 'data', 'myfile.jsx' ]
PS D:\Collage\Sem 6\FSD\Practice> █
```

util :

Code :

```
const util = require('node:util');

console.log("\nString Formate")

var result = util.format('<%s>+<%s>', 'Hello', 'World');
console.log(result);

result = util.format('%s : %d', 'Marks', '15');
console.log(result);

console.log("\nKey value pair are highloghted with color");
result = util.formatWithOptions({ colors: true }, 'See object %O', { Marks : 15 });
console.log(result);

console.log("\nis Arrayy : []");
console.log(util.isArray([]));

console.log("\nis Arrayy : {a:10}");
console.log(util.isArray({a:10}));

console.log("\nisDate : new Date()");
console.log(util.isDate(new Date()));

console.log("\nisDate : {} ");
console.log(util.isDate({}));

console.log("\nisError : new Error()");
console.log(util.isError(new Error("Error")));

console.log("\nisError : {message : 'FSD'}");
console.log(util.isError({message : 'FSD'}));
```

Output:



```
PS D:\Collage\Sem 6\FSD\Practice> node "d:\Collage\Sem 6\FSD\Practice\02-01-24\Test1\util_practice.js"

String Formate
<Hello>+<World>
Marks : 15

Key value pair are highloghted with color
See object { Marks: 15 }

is Arrayy : []
true

is Arrayy : {a:10}
false

isDate : new Date()
true

isDate : {}
false

isError : new Error()
true

isError : {message : 'FSD'}
false
PS D:\Collage\Sem 6\FSD\Practice>
```

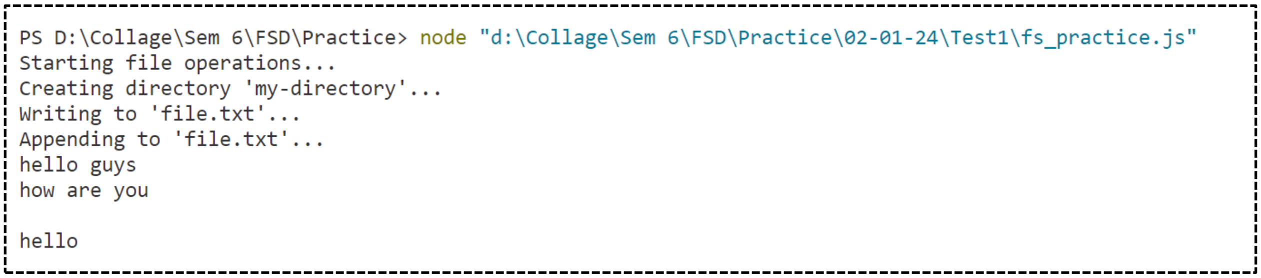
fs :

Code :

```
const fs = require("fs");
const path = require("path");

console.log("Starting file operations...");

// Create a directory
console.log("Creating directory 'my-directory'...");
fs.mkdirSync(path.join(__dirname, "my-directory"));

// Write to a file
console.log("Writing to 'file.txt'...");
fs.writeFileSync("file.txt", "hello guys\n");
fs.writeFileSync("file.txt", "how are you\n", { flag: "a" }); // Append mode

// Append to a file
console.log("Appending to 'file.txt'...");
fs.appendFileSync("file.txt", "\nhello\n");

// Read from a file and output to console
console.log("Reading 'file.txt'...");
const fileContent = fs.readFileSync("file.txt").toString();
console.log(fileContent);

// Rename a file
console.log("Renaming 'file.txt' to 'FsModule.txt'...");
fs.rename("file.txt", "FsModule.txt", () => {
  console.log("Renamed file to 'FsModule.txt'");
});

// Create a subdirectory
console.log("Creating subdirectory 'my-directory/sub-directory'...");
fs.mkdirSync(path.join(__dirname, "my-directory", "sub-directory"));

// Delete a file
console.log("Deleting 'file.txt'...");
fs.unlinkSync(path.join(__dirname, "file.txt"));
```

```
// Delete a directory
console.log("Deleting directory 'my-directory'...");
fs.rmdirSync(path.join(__dirname, "my-directory"));

console.log("Finished file operations.");
```

Output:



```
PS D:\Collage\Sem 6\FSD\Practice> node "d:\Collage\Sem 6\FSD\Practice\02-01-24\Test1\fs_practice.js"
Starting file operations...
Creating directory 'my-directory'...
Writing to 'file.txt'...
Appending to 'file.txt'...
hello guys
how are you

hello
```