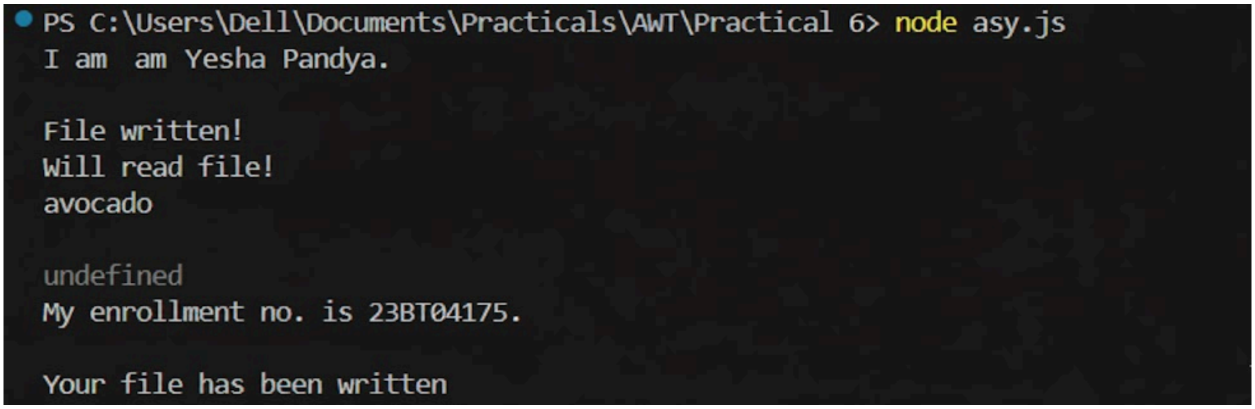
PRACTICAL 6

Creating a custom web component that can be reused across multiple projects

NodeJS File Operations

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

//blocking (Synchronous) Way
const textIn = fs.readFileSync('input.txt', 'utf-8');
console.log(textIn);

const textOut = `This is what we know about the avocado: ${textIn}.`;
fs.writeFileSync('output.txt', textOut);
console.log('File written (synchronously)!');

//Non-Blocking (Asynchronous) Way
fs.readFile('start.txt', 'utf-8', (err, data1) => {
  if (err) return console.log('ERROR reading start.txt!');
  console.log('start.txt contents:', data1);

  //read the file whose name is inside start.txt
  fs.readFile(`${data1}.txt`, 'utf-8', (err, data2) => {
    if (err) return console.log(`ERROR reading ${data1}.txt!`);
    console.log(`${data1}.txt contents:`, data2);

    //read append.txt
    fs.readFile('append.txt', 'utf-8', (err, data3) => {
      if (err) return console.log('ERROR reading append.txt!');
      console.log('append.txt contents:', data3);

      //write the final output
      fs.writeFile('final.txt', `${data2}\n${data3}`, 'utf-8', err => {
        if (err) return console.log('ERROR writing final.txt!');
        console.log('Your file has been written successfully!');
      });
    });
  });
});

console.log('Will read file asynchronously...');
```

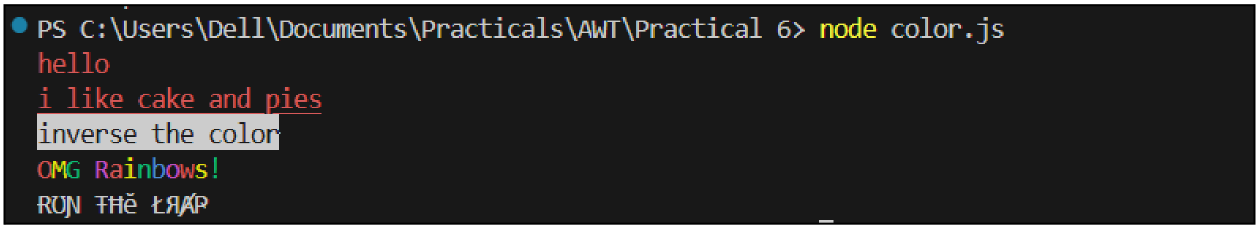


```
PS C:\Users\Dell\Documents\Practicals\AWT\Practical 6> node asy.js
I am  am Yesha Pandya.

File written!
Will read file!
avocado

undefined
My enrollment no. is 23BT04175.

Your file has been written
```

NodeJS 'colors.js'

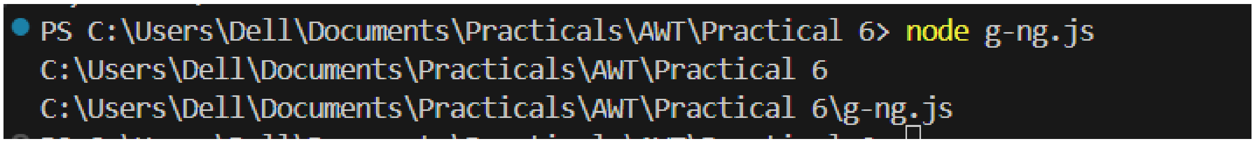
(Run `npm install colors` before running following code to install the colours package)

```
//import the 'colors' package
var colors = require('colors');

//basic color formatting
console.log('hello'.red);

//you can chain multiple styles
console.log('i like cake and pies'.underline.red);

//inverse (swap background and foreground)
console.log('inverse the color'.inverse);

//fun rainbow text
console.log('OMG Rainbows!'.rainbow);

//"Trap" style animation (works only in some terminals)
console.log('Run the trap'.trap);
```



```
PS C:\Users\Dell\Documents\Practicals\AWT\Practical 6> node color.js
hello
i like cake and pies
inverse the color
OMG Rainbows!
RUN ŦĦĕ ŁЯȺP
```

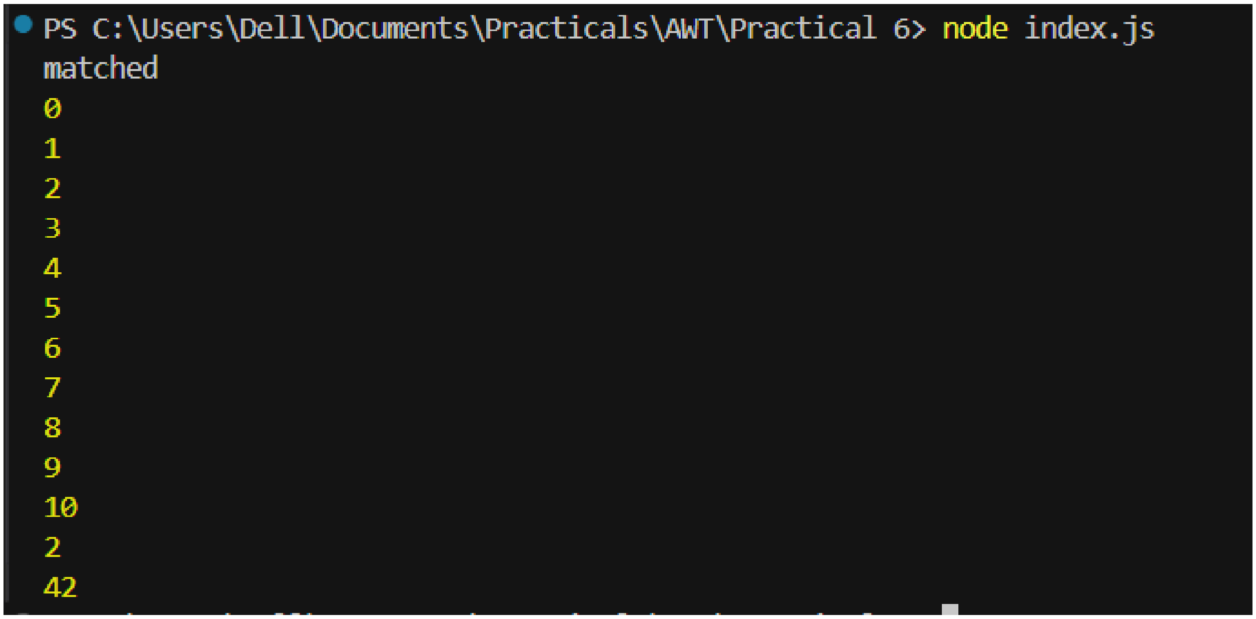
```
console.log(__dirname);
console.log(__filename)
```



```
PS C:\Users\Dell\Documents\Practicals\AWT\Practical 6> node g-ng.js
C:\Users\Dell\Documents\Practicals\AWT\Practical 6
C:\Users\Dell\Documents\Practicals\AWT\Practical 6\g-ng.js
```

```
//index.js
var x = 20;

if (x == 20) {
  console.log("matched");
}
for (let i = 0; i <= 10; i++) {
  console.log(i);
}
//importing module
const app = require('./app');

const arr = [2, 4, 7, 6, 8, 9, 10];
```

```
console.log(arr[0]);      // prints 2
console.log(app.x);       // prints 42 (from app.js)

//app.js
module.exports = {
  x: 42
};
```



```
PS C:\Users\Dell\Documents\Practicals\AWT\Practical 6> node index.js
matched
0
1
2
3
4
5
6
7
8
9
10
2
42
```

👉Node.js mini project that serves a dynamic product website with routing, templates, and images

Project Folder Structure:

```
project-folder/
│
├── index2.js
├── data.json
├── modules/
│   └── replaceTemplate.js
├── templates/
│   ├── template-overview.html
│   ├── template-card.html
│   └── template-product.html
└── images/
    ├── apple.jpg
    ├── banana.jpg
    └── orange.jpg
```

index2.js

```
const fs = require('fs');
const http = require('http');
const url = require('url');
const path = require('path');
const slugify = require('slugify');
const replaceTemplate = require('./modules/replaceTemplate');

//Reading Templates
const tempOverview =
fs.readFileSync(`${__dirname}/templates/template-overview.html`, 'utf-8');
const tempCard = fs.readFileSync(`${__dirname}/templates/template-card.html`,
'utf-8');
const tempProduct =
fs.readFileSync(`${__dirname}/templates/template-product.html`, 'utf-8');

//Reading Data
const data = fs.readFileSync(`${__dirname}/data.json`, 'utf-8');
const dataObj = JSON.parse(data);

//Creating Slugs (for clean URLs)
const slugs = dataObj.map(el => slugify(el.productName, { lower: true }));
dataObj.forEach((el, i) => el.slug = slugs[i]); // attach slug to each object
console.log('Generated slugs:', slugs);

//Create Server
const server = http.createServer((req, res) => {
  const { query, pathname } = url.parse(req.url, true);

  //Serve Images
  if (pathname.startsWith('/images/')) {
    const imagePath = path.join(__dirname, pathname);
    const extname = path.extname(imagePath).toLowerCase();
    let contentType = 'text/html';

    if (extname === '.jpg' || extname === '.jpeg') contentType = 'image/jpeg';
    else if (extname === '.png') contentType = 'image/png';
    else if (extname === '.gif') contentType = 'image/gif';

    fs.readFile(imagePath, (err, data) => {
      if (err) {
        res.writeHead(404, { 'Content-type': 'text/html' });
        return res.end('<h1>Image not found</h1>');
      }
      res.writeHead(200, { 'Content-type': contentType });
      res.end(data);
    });
    return;
  }

  //Overview Page
  if (pathname === '/' || pathname === '/overview') {
    res.writeHead(200, { 'Content-type': 'text/html' });
    const cardsHtml = dataObj.map(el => replaceTemplate(tempCard,
el)).join('');
    const output = tempOverview.replace('{%PRODUCT_CARDS%}', cardsHtml);
    res.end(output);

  //Product Page
  } else if (pathname === '/product') {
    res.writeHead(200, { 'Content-type': 'text/html' });
    const product = dataObj.find(p => p.slug === query.id || p.id ==
query.id);
    if (!product) {
      res.writeHead(404, { 'Content-type': 'text/html' });
      return res.end('<h1>Product not found!</h1>');
    }
    const output = replaceTemplate(tempProduct, product);
    res.end(output);

  //API Route
  } else if (pathname === '/api') {
    res.writeHead(200, { 'Content-type': 'application/json' });
    res.end(data);

  //Not Found
  } else {
    res.writeHead(404, {
      'Content-type': 'text/html',
      'my-own-header': 'hello-world'
    });
    res.end('<h1>Page not found!</h1>');
  }
});

//Start Server
server.listen(8000, '127.0.0.1', () => {
  console.log('Server running at http://127.0.0.1:8000');
});
```

modules/replaceTemplate.js

```
module.exports = (template, product) => {
  let output = template;
  output = output.replace(/{%PRODUCT_NAME%}/g, product.productName);
  output = output.replace(/{%PRODUCT_DESCRIPTION%}/g,
product.productDescription);
  output = output.replace(/{%PRODUCT_IMAGE%}/g, product.productImage);
  output = output.replace(/{%SLUG%}/g, product.slug);
  return output;
};
```

templates/template-overview.html

```
<html>
  <body>
    <h1>Product Overview</h1>
    <div>
      {%PRODUCT_CARDS%}
    </div>
  </body>
</html>
```

templates/template-card.html

```
<div class="card">
  <img src="/images/{%PRODUCT_IMAGE%}" alt="{%PRODUCT_NAME%}">
  <h2>{%PRODUCT_NAME%}</h2>
  <p>{%PRODUCT_DESCRIPTION%}</p>
  <a href="/product?id={%SLUG%}">View Details</a>
</div>
```

templates/template-product.html

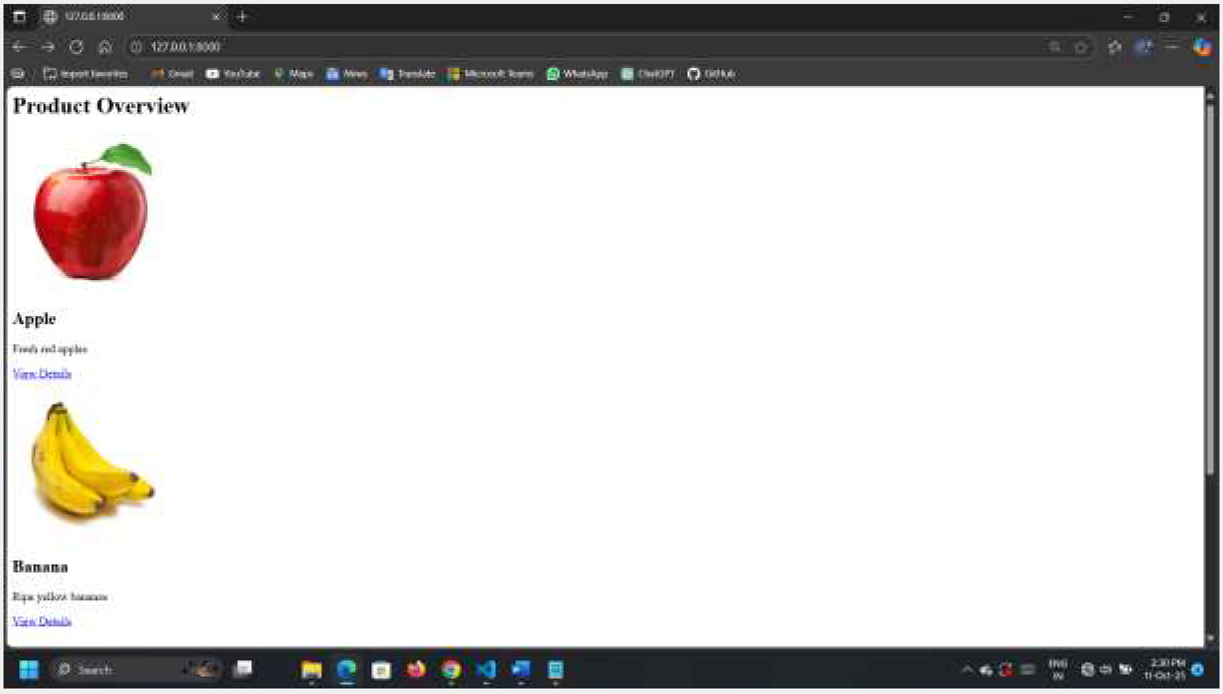
```
<html>
  <body>
    <h1>{%PRODUCT_NAME%}</h1>
    <img src="/images/{%PRODUCT_IMAGE%}" alt="{%PRODUCT_NAME%}">
    <p>{%PRODUCT_DESCRIPTION%}</p>
  </body>
</html>
```

data.json

```
[
  {
    "id": 0,
    "productName": "Apple",
    "productDescription": "Fresh red apples",
    "productImage": "apple.jpg"
  },
  {
    "id": 1,
    "productName": "Banana",
    "productDescription": "Ripe yellow bananas",
    "productImage": "banana.jpg"
  },
  {
    "id": 2,
    "productName": "Orange",
    "productDescription": "Juicy oranges",
    "productImage": "orange.jpg"
  }
]
```

To run:

1. Install dependencies `npm install slugify`
2. Run server `node index2.js`



Product Overview

Apple

Fresh red apples

View Details

Banana

Ripe yellow bananas

View Details

****