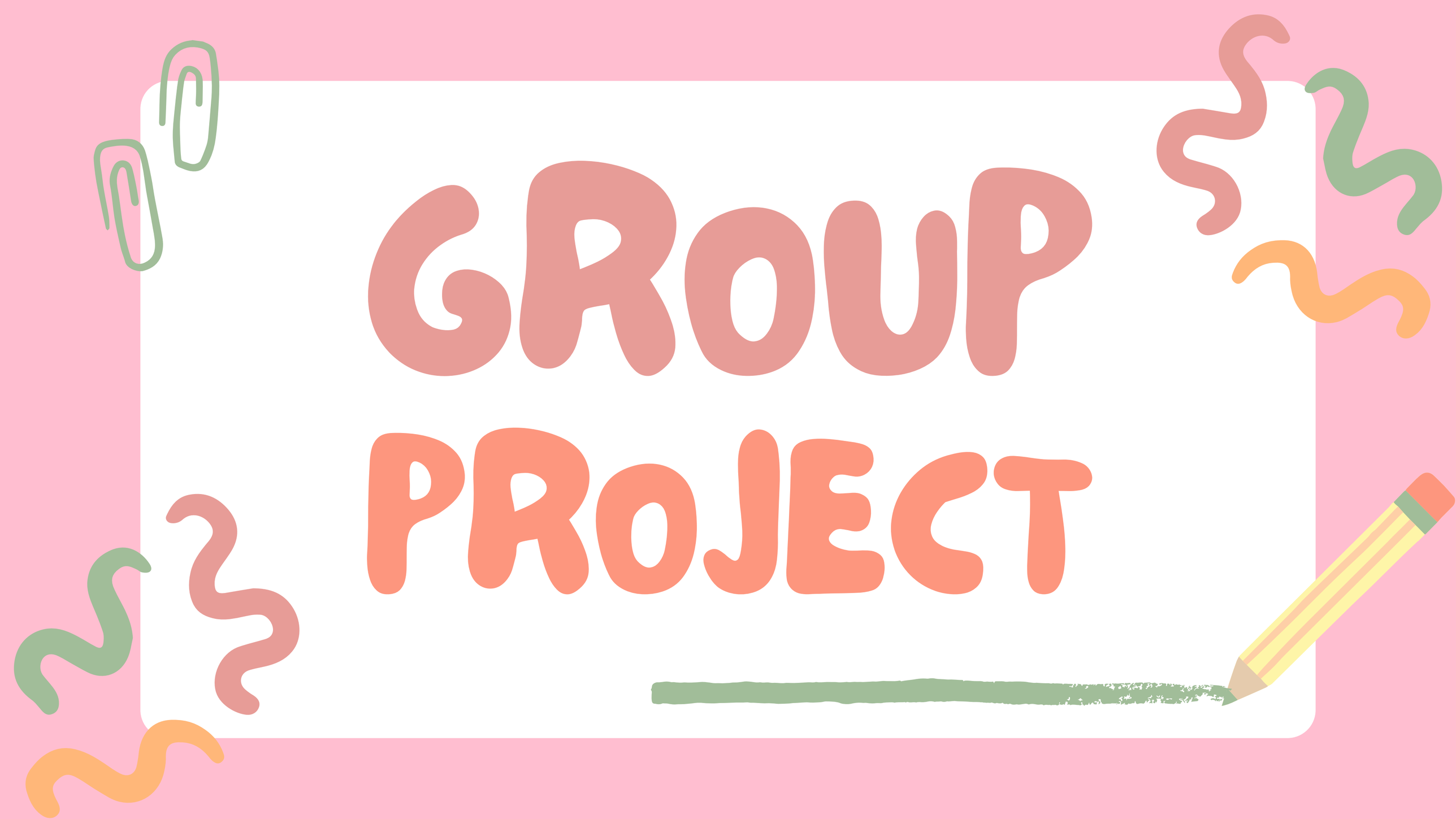
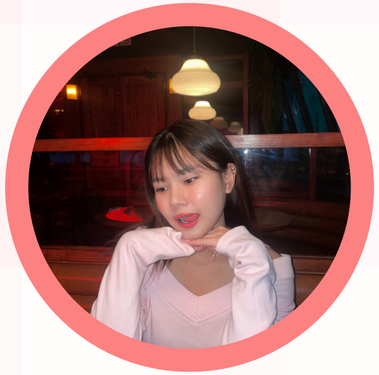
GROUP PROJECT



MEMBER OF GROUP



THANYANAN KHAMPOOL
65102010160

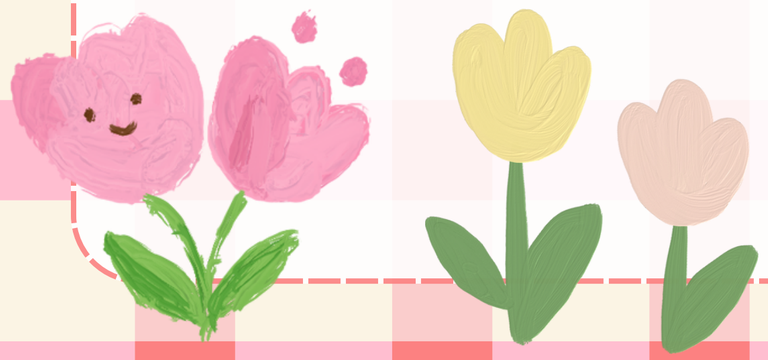
PHANTIPHA SRIPADUPWONG
65102010208

WERAPAT PHASIT
65102010210

THANIS KLEEBRAKSON
65102010688

PROJECT OVERVIEW NODE.JS

The project is a Node.js-based web application designed to provide [state the purpose or goal of the application]. It utilizes various technologies and frameworks, with Node.js serving as the primary runtime environment. The application follows a scalable architecture, consisting of server-side components, a database layer, and potentially a client-side interface



ARCHITECTURE

1. EVENT-DRIVEN ARCHITECTURE: NODE.JS EMPLOYS AN EVENT-DRIVEN ARCHITECTURE THAT UTILIZES A SINGLE-THREADED EVENT LOOP TO HANDLE MULTIPLE CONCURRENT OPERATIONS EFFICIENTLY.
2. NON-BLOCKING I/O: NODE.JS USES NON-BLOCKING I/O OPERATIONS, ALLOWING IT TO HANDLE MULTIPLE REQUESTS WITHOUT WAITING FOR EACH OPERATION TO COMPLETE, WHICH RESULTS IN IMPROVED PERFORMANCE AND RESPONSIVENESS.
3. V8 ENGINE INTEGRATION: BUILT ON GOOGLE'S V8 JAVASCRIPT ENGINE, NODE.JS BENEFITS FROM HIGH-SPEED EXECUTION OF JAVASCRIPT CODE, ENABLING FAST AND SCALABLE APPLICATIONS.
4. NPM ECOSYSTEM: NODE.JS IS SUPPORTED BY NPM, THE WORLD'S LARGEST PACKAGE ECOSYSTEM FOR JAVASCRIPT, PROVIDING ACCESS TO A VAST ARRAY OF MODULES AND TOOLS FOR BUILDING VARIOUS TYPES OF APPLICATIONS

PROS AND CONS

PROS

- Scalability: Node.js is known for its ability to handle a large number of concurrent connections efficiently, making it suitable for applications that require high scalability.
- Performance: Node.js uses non-blocking, event-driven architecture, which allows it to handle I/O operations asynchronously, resulting in faster performance compared to traditional server-side technologies.
- Large ecosystem: Node.js has a vast ecosystem of libraries and packages available through npm (Node Package Manager), which makes it easy for developers to find and use third-party modules to build their applications.
- Single language: Node.js allows developers to use JavaScript for both client-side and server-side development, enabling full-stack development with a single language.

CONS

- Callback hell: Managing asynchronous code in Node.js using callbacks can sometimes lead to callback hell, where code becomes nested and difficult to read and maintain.
- Unstable APIs: Node.js ecosystem evolves rapidly, which can lead to frequent changes in APIs and libraries, causing compatibility issues and requiring frequent updates to applications.
- Limited CPU-bound tasks: Node.js is not well-suited for CPU-bound tasks due to its single-threaded event loop model, which can lead to poor performance for tasks that require heavy CPU usage.
- Learning curve: For developers coming from a synchronous programming background, the asynchronous nature of Node.js can have a steep learning curve.

BEST PRACTICES

- ASYNCHRONOUS PROGRAMMING: EMBRACE NODE.JS'S ASYNCHRONOUS, EVENT-DRIVEN NATURE BY UTILIZING CALLBACKS, PROMISES, OR ASYNC/AWAIT SYNTAX TO HANDLE ASYNCHRONOUS OPERATIONS SUCH AS FILE I/O, NETWORK REQUESTS, AND DATABASE QUERIES.
- MODULARIZATION: ORGANIZE YOUR NODE.JS APPLICATIONS INTO MODULES TO PROMOTE CODE REUSABILITY, MAINTAINABILITY, AND SCALABILITY. USE THE COMMONJS OR ES MODULES SYNTAX TO DEFINE MODULES AND ENCAPSULATE RELATED FUNCTIONALITY
- ERROR HANDLING: IMPLEMENT ROBUST ERROR HANDLING MECHANISMS TO GRACEFULLY HANDLE ERRORS AND PREVENT APPLICATION CRASHES. USE TRY-CATCH BLOCKS OR ERROR-FIRST CALLBACKS TO HANDLE SYNCHRONOUS AND ASYNCHRONOUS ERRORS EFFECTIVELY.

COMMON USE CASES

- Web Applications: Node.js is well-suited for building scalable and high-performance web applications. Its event-driven architecture and non-blocking I/O model make it deal for handling a large number of concurrent requests, making it popular for real-time applications like chat applications, collaboration tools, social media platforms, and content management systems.
- API Development: Node.js is widely used for building RESTful APIs to power web and mobile applications. Its lightweight and fast nature, along with frameworks like Express.js, makes it easy to create APIs for various purposes such as microservices, backend services for single-page applications, and integrations with third-party services.
- Real-time Applications: Node.js excels in building real-time applications that require bi-directional communication between the client and server.

FEATURES

- Cross-Platform: Node.js runs on operating systems, including Windows, macOS, and Linux, ensuring compatibility across different environments. This allows developers to write code once and deploy it seamlessly across different platforms.
- Built-in HTTP Module: Node.js includes a built-in HTTP module that simplifies the process of creating HTTP servers and handling HTTP requests and responses. This makes it easy to build web applications and APIs using Node.js.
- Streams and Buffers: Node.js provides native support for streams and buffers, which are essential for handling large amounts of data efficiently. Streams allow developers to process data in chunks, while buffers are used for handling binary data.

WORKING WITH THE FRAMEWORK. DEMO HOW TO GET STARTED AND SHOW SOME EXAMPLES

index.html | script.js | style.css

index.html > html

```html
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="style.css">
    <title>Seiya Shop</title>
</head>

<body>
    <header>
        <h1>Seiya Bloxfruit</h1>
    </header>

    <main id="product-list">
        <!-- Product listing goes here -->
    </main>

    <script src="script.js"></script>
</body>

</html>
```

index.html | script.js | style.css

style.css > ...

```css
body {
    font-family: 'Arial', sans-serif;
    margin: 0;
    padding: 0;
    background-color: #f4f4f4;
}

header {
    background-color: #333;
    color: white;
    text-align: center;
    padding: 1em;
}

main {
    display: flex;
    flex-wrap: wrap;
    justify-content: space-around;
    padding: 2em;
}

.product {
    border: 1px solid #ddd;
    padding: 1em;
    margin: 1em;
    width: 200px;
    text-align: center;
    background-color: white;
}
```

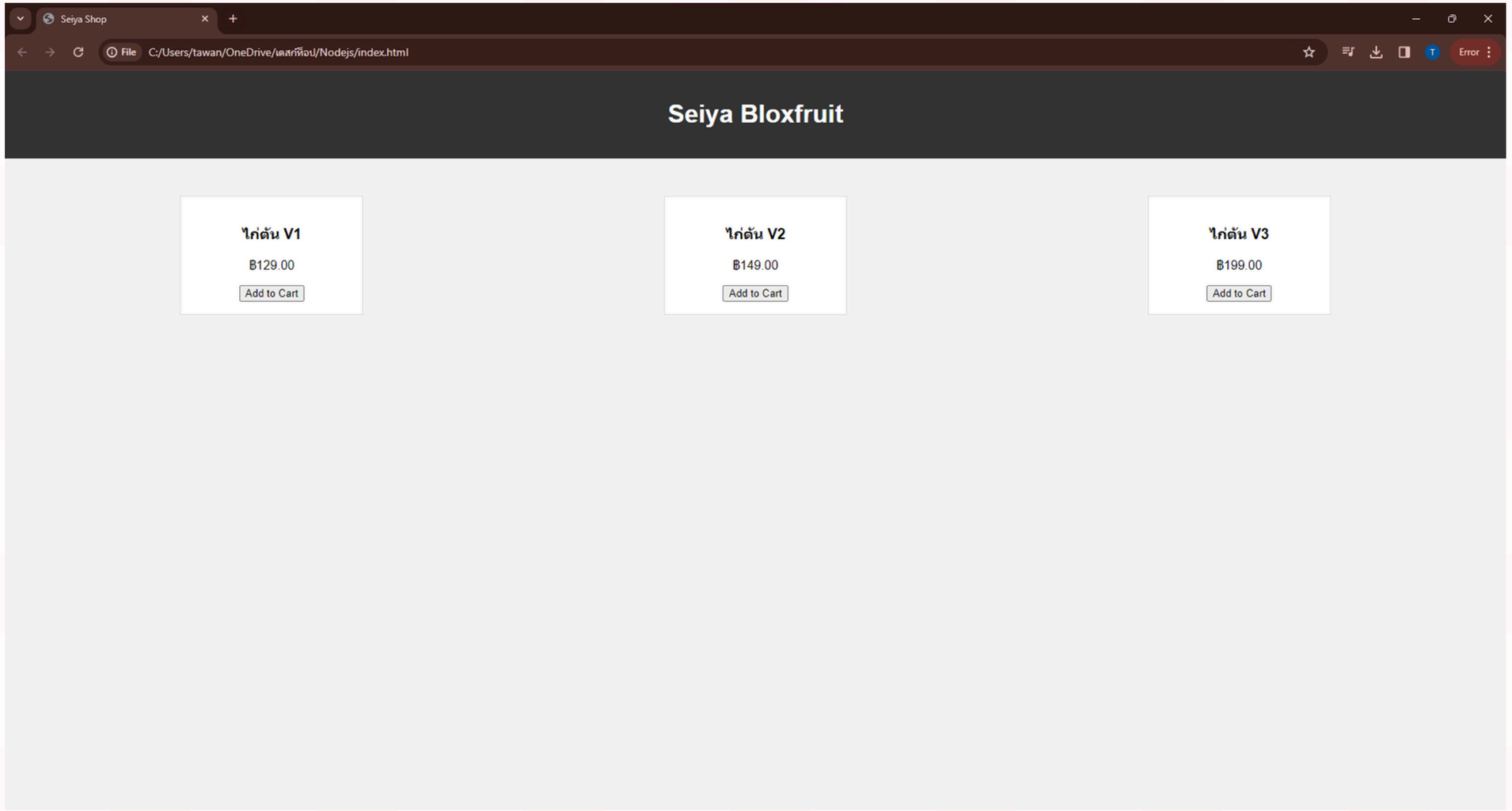
index.html | script.js | style.css

script.js > ...

```js
document.addEventListener('DOMContentLoaded', function () {
    const productList = document.getElementById('product-list');

    const products = [
        { id: 1, name: 'ไก่ตัน V1', price: 129 },
        { id: 2, name: 'ไก่ตัน V2', price: 149 },
        { id: 3, name: 'ไก่ตัน V3', price: 199 },
    ];

    products.forEach(product => {
        const productElement = document.createElement('div');
        productElement.classList.add('product');
        productElement.innerHTML = `
            <h3>${product.name}</h3>
            <p>฿${product.price.toFixed(2)}</p>
            <button>Add to Cart</button>
        `;
        productList.appendChild(productElement);
    });
});
```


Seiya Shop
File
C:/Users/tawan/OneDrive/เดสก์ท็อป/Nodejs/index.html
Error
Seiya Bloxfruit
ไก่ต้น V1
฿129.00
Add to Cart
ไก่ต้น V2
฿149.00
Add to Cart
ไก่ต้น V3
฿199.00
Add to Cart

THANK YOU!

OUR Group