

3.3 Spread and Rest Operators (Finalized)

3.3.1 What is the Spread Operator (...)

The spread operator allows an iterable (like an array or object) to be expanded in places like function arguments or array literals.

```
js
-----
const arr = [1, 2, 3];
const newArr = [...arr, 4, 5]; // [1, 2, 3, 4, 5]
```

3.3.2 What is the Rest Operator (...)

The rest operator collects all remaining elements into a single array.

```
js
-----
function sum(...args) {
  console.log(args); // All arguments as an array
}
sum(1, 2, 3); // Output: [1, 2, 3]
```

3.3.3 Rest in Function Parameters (taught in class)

```
js
-----
function sum(...numbers) {
  let total = 0;
  for (const num of numbers)
    total += num;
  console.log(total);
}
sum(3, 5, 7, 4, 3, 4, 6, 8, 0); // Output: 40
```

3.3.4 Rest in Destructuring (taught in class)

```
js
-----
const [a, b, ...rest] = [4, 7, 3, 2, 8, 7, 5, 3, 2];
console.log(a); // 4
console.log(b); // 7
console.log(rest); // [3, 2, 8, 7, 5, 3, 2]
```

3.3.5 Difference Between Spread and Rest

Feature	Spread	Rest
Usage	Expands elements	Gathers elements
Use in Arrays	<code>[...arr]</code>	<code>const [a, ...rest] = arr</code>
Use in Objects	<code>{...obj}</code>	<code>function func(...args)</code>

3.3.6 Common Use Cases

- Copying arrays
 - Merging arrays
 - Passing arguments
 - Collecting function arguments
 - Partial destructuring
-

3.3.7 Common Errors / Gotchas

- Can only use one rest element per function
 - Rest must be the last parameter
 - Spread creates shallow copies (not deep)
-



PROJECT SECTION: React App Setup with Vite



React Project Initialization (Vite-based)

1. Run Vite Project Creation Command:

```
bash
-----
npm create vite@latest my-react-app
```

2. Respond to Prompts:

```
sql
-----
Need to install the following packages:
create-vite@7.0.3
Ok to proceed? (y) y
```

3. Select Variant:

- Choose: JavaScript

```
cpp
-----
◇ Select a variant:
| JavaScript
```

4. Vite scaffolds the project:

```
arduino
-----
◇ Scaffolding project in ...
└─ Done. Now run:
  cd my-react-app
  npm install
  npm run dev
```

5. Navigate into the project:

```
bash
-----
cd my-react-app
```

6. Install dependencies:

```
bash
-----
npm install
```

Output sample:

```
rust
-----
added 152 packages, and audited 153 packages in 18s

33 packages are looking for funding
run `npm fund` for details

found 0 vulnerabilities
```

7. Run development server:

```
bash
-----
npm run dev
```

Output:

```
css
-----
> my-react-app@0.0.0 dev
> vite

VITE v7.0.6  ready in 677 ms


→ Local:   http://localhost:5173/
→ Network: use --host to expose
→ press h + enter to show help
```

Why React Applications Are SPA (Single Page Applications)

- **SPA** means the browser loads a **single HTML file** (`index.html`), and dynamically updates the UI using JavaScript.
- In React:
 - Page changes don't reload the entire page.
 - Instead, components are shown/hidden or updated via **JavaScript + Virtual DOM**.
- This leads to:
 - Faster user experience (no full reloads).
 - Smooth transitions between pages.
 - Easy state management with tools like Redux/Context.

Client-Server Interaction in React (with Vite)

- On initial load:
 1. Browser requests the site → server sends `index.html`.
 2. `<script type="module" src="/src/main.jsx">` boots the React app.
 3. All content changes happen inside the `<div id="root">` via JavaScript.
 4. Subsequent navigation doesn't re-request `index.html`.

 **Only the first time** the full HTML is fetched; after that, React handles UI changes internally.

`App.jsx` Initial Setup (as taught in class)

In Vite's default structure, the `App.jsx` file is where the core logic of your React component resides.

File: `App.jsx`


```
jsx
-----
import { useState } from 'react'
import reactLogo from './assets/nik.jpg'
import viteLogo from '/vite.svg'
import './App.css'

function App() {
  return (
    <div>
      <h1>Hellooo</h1>
      <h1>maii Namee in Nikhil</h1>
    </div>
  );
}

export default App;
```

Breakdown:

Line	Explanation
<code>useState import</code>	Prepares for using React state (not used yet here).
<code>nik.jpg</code>	Image file imported from <code>assets</code> folder.
<code>vite.svg</code>	Default Vite logo used in most starter templates.
<code>App.css</code>	Styling file applied to this component.
<code>function App()</code>	Defines a functional React component.
<code>return (...)</code>	Returns JSX to render UI.
<code>export default App;</code>	Makes <code>App</code> available for use in <code>main.jsx</code> .

 Your React app will render this component into the DOM (inside `#root`) via `main.jsx`.

File Relationship (so far):

- `index.html` → **Loads** `main.jsx`
- `main.jsx` → **Loads** `App.jsx`
- `App.jsx` → **Displays JSX content and uses CSS/assets**