🛠 1. Setting Up Development Environment

✅ Required Tools:

- **Node.js (v18+)**
- **VS Code**
- **Vue.js Devtools (Chrome Extension)**

🧩 Recommended VS Code Extensions:

- Vue - Official — for Vue 3 [https://marketplace.visualstudio.com/items?itemName=Vue.volar]
- ESLint — linting support
- Prettier — code formatting

🔍 Chrome Extension:

- **Vue.js Devtools**: Helps inspect Vue component structure, props, states, and events

⚙️ 2. Scaffolding Your First Vue.js Project with Vite

✨ Why Vite?

- Super fast development server
- Native ES Module support
- Designed for Vue 3 out of the box

📦 Create Project:

```
npm create vue@latest my-vue-app
cd my-vue-app
npm install
npm run dev
```

✅ Folder Structure:

```
my-vue-app/
│
├── index.html          # Entry HTML file
├── package.json        # Project metadata
├── public/             # Static assets
├── src/
```

```
│   ├── assets/          # Images, styles
│   ├── components/      # Vue components
│   ├── App.vue          # Main App component
│   └── main.js          # App entry script
```

🧰 3. Preparing Our Project for Work

✅ Clean `App.vue` for fresh start:

```
<!-- src/App.vue -->
<script setup>
// Composition API script block
</script>

<template>

    <h1>Welcome to Vue 3 with Vite</h1>

</template>

<style scoped>

</style>
```

📊 4. Using Data in Vue 3 (Composition API)

Example: Display String

```
<script setup>
import { ref } from 'vue'

const name = ref('John Doe')
const role = ref('Frontend Developer')
</script>

<template>

    <h2>{{ name }}</h2>
    <p>{{ role }}</p>

</template>
```

Example: Display Single Object

```
<script setup>
import { reactive } from 'vue'

const user = reactive({
  name: 'Alice',
  email: 'alice@example.com'
})
</script>

<template>
  <div>
    <h2>{{ user.name }}</h2>
    <p>{{ user.email }}</p>
  </div>
</template>
```

Example: Display List from Array

```
<script setup>
import { ref } from 'vue'

const fruits = ref(['Apple', 'Banana', 'Mango'])
</script>

<template>
  <ul>
    <li v-for="(item, index) in fruits" :key="index">{{ item }}</li>
  </ul>
</template>
```