

# Vue.js Beginner's Guide

**Osanda Abeysinghe**

## **Tabel of Contents**

- 1. Introduction to Vue.js**
- 2. Setup Guide & Installation**
- 3. Core Concepts**
- 4. Styling with Vuetify**
- 5. Routing in Vue.js**
- 6. Hands on Example**

# 1. Introduction to Vue.js

## What is Vue.js?

Vue.js is a modern JavaScript framework used to build user interfaces and web applications. Think of it as a tool that helps you create interactive websites more easily. Vue.js makes it simple to display data on web pages and respond to user actions like clicks and form submissions. The name "Vue" comes from the French word meaning "view," which makes sense because Vue.js focuses on the view layer of web applications what users see and interact with.

Vue.js is,

- **Easy to learn** - Perfect for beginners
- **Fast** - Creates responsive applications
- **Flexible** - Works well for small and large projects
- **Well documented** - Has excellent guides and tutorials

## History of Vue.js

Vue.js was created by Evan You in 2013 while he was working at Google. The first official release was in 2014. Vue.js 1.0 was released in 2015 Since then, Vue has become very popular among developers because its easy to learn and use.

## Purpose

Vue.js was designed to make building interactive web interfaces easier and more enjoyable. Its main purposes are

- **Create Dynamic Web Pages** - Make websites that can change content without refreshing
- **Manage Complex Data** - Handle information that changes over time
- **Build Single Page Applications** - Websites that feel like desktop applications
- **Component Based Development** - Build reusable pieces of code

## Why Developers use Vue.js

Advantages

- Vue.js is designed to be approachable for beginners while remaining powerful for experts. Can start learning Vue.js with basic HTML, CSS, and JavaScript knowledge.
- Vue.js has some of the best documentation in the JavaScript ecosystem.
- Vue.js can be adopted incrementally.
- Flexible and Versatile.
- Vue.js is optimized for speed and efficiency.

Comparisons

Features	Vue.js	React	Angular
Performance	Automatic performance optimizations	Excellent runtime performance	Excellent performance for large applications
Development	Fastest development	More Setup Required	Most Complex Setup
Tools	Vue DevTools, Vite, Nuxt.js	Create React App, Next.js, Gatsby	Angular CLI, Angular Universal
Learning	Template syntax like HTML	Functional components and hooks concept	Functional components and hooks concept

## 2. Setup Guide & Installation

### Using CDN

Step 1: Create HTML file

Add this line - `<script src="https://unpkg.com/vue@3/dist/vue.global.js"></script>`

Step 2: Run application

### Vue CLI

Prerequisites

`node --version`

`npm --version`

Step 1: Install Vue CLI

`npm install -g @vue/cli`

Step 2: Create new project

`vue create my-vue-project`

Step 3: Choose project configuration

Step 4: Installation

Step 5: Navigate and Start Server

`cd my-vue-project`

`npm run server`

Step 6: <http://localhost> Open Browser

### Vite

Step 1: Create Vue 3 project with Vite

# Method 1: Using npm

`npm create vue@latest my-vue-app`

# Method 2: Using yarn

`yarn create vue my-vue-app`

# Method 3: Using pnpm

`pnpm create vue my-vue-app`

Step 2: Interactive Project Setup

Step 3: Install dependencies and start development

```
cd my-vue-app
```

```
npm install
```

```
npm run dev
```

Step 4: Open application

```
C:\Users\LapMaster\my-vue-app>npm run dev

> my-vue-app@0.0.0 dev
> vite

VITE v7.1.4  ready in 814 ms

→ Local:   http://localhost:5173/
→ Network: use --host to expose
→ Vue DevTools: Open http://localhost:5173/__devtools__/ as a separate window
→ Vue DevTools: Press Alt(⌘)+Shift(⇧)+D in App to toggle the Vue DevTools
→ press h + enter to show help
```

## Folder structure

```

✓ MY-VUE-APP
├── > .vscode
├── > node_modules
├── > public
├── > src
├── .gitignore
├── index.html
├── ✓ package.json
├──   package-lock.json
├── i README.md
├── ✓ ⚡ vite.config.js
├──   jsconfig.json
├──   ✓ vitest.config.js
```

### 3. Core Concepts

#### Vue Instance and Reactivity

A Vue instance is an object that contains app data, methods, and configuration. Think of it as the brain that control application.

Reactivity is feature that automatically update user interface when data changes. When modify data, Vue automatically re-renders the parts of page that use that data.

#### Templates & Directives

Templates are HTML with Vue syntax that describe how UI should look. Vue compiles templates into JavaScript functions that efficiently update the DOM.

#### Components

Components are reusable pieces of code that encapsulate HTML, CSS, and JavaScript functionality. They are the building blocks of Vue applications.

Components are,

- Reusability
- Organization
- Maintainability

Props - Are custom attributes can pass data from parent to child components.

Events - Children can communicate with parents by emitting custom events.

#### Lifecycle Hooks

Lifecycle hooks are special methods that Vue calls automatically at different stages of a components life. They allow to run code at specific times.

## 4. Styling with Vuetify

Vuetify is a Vue.js component library that provides beautiful, ready-made components following Google Material Design guidelines. It is like having a box of pre-designed LEGO blocks for building website.

Benefits:

- **Ready-made components** - Buttons, cards, navigation, forms
- **Beautiful design** - Follows Material Design principles
- **Responsive** - Works on all screen sizes
- **Consistent** - All components look great together

### Installing Vuetify

**New project:**

```
vue create my-app
```

```
cd my-app
```

```
vue add vuetify
```

**Existing project:**

```
npm install vuetify@^3.0.0
```



## 5. Routing in Vue.js

Routing lets show different pages or views inside your app without refreshing the browser.

### Install Vue Router:

```
npm install vue-router@4
```

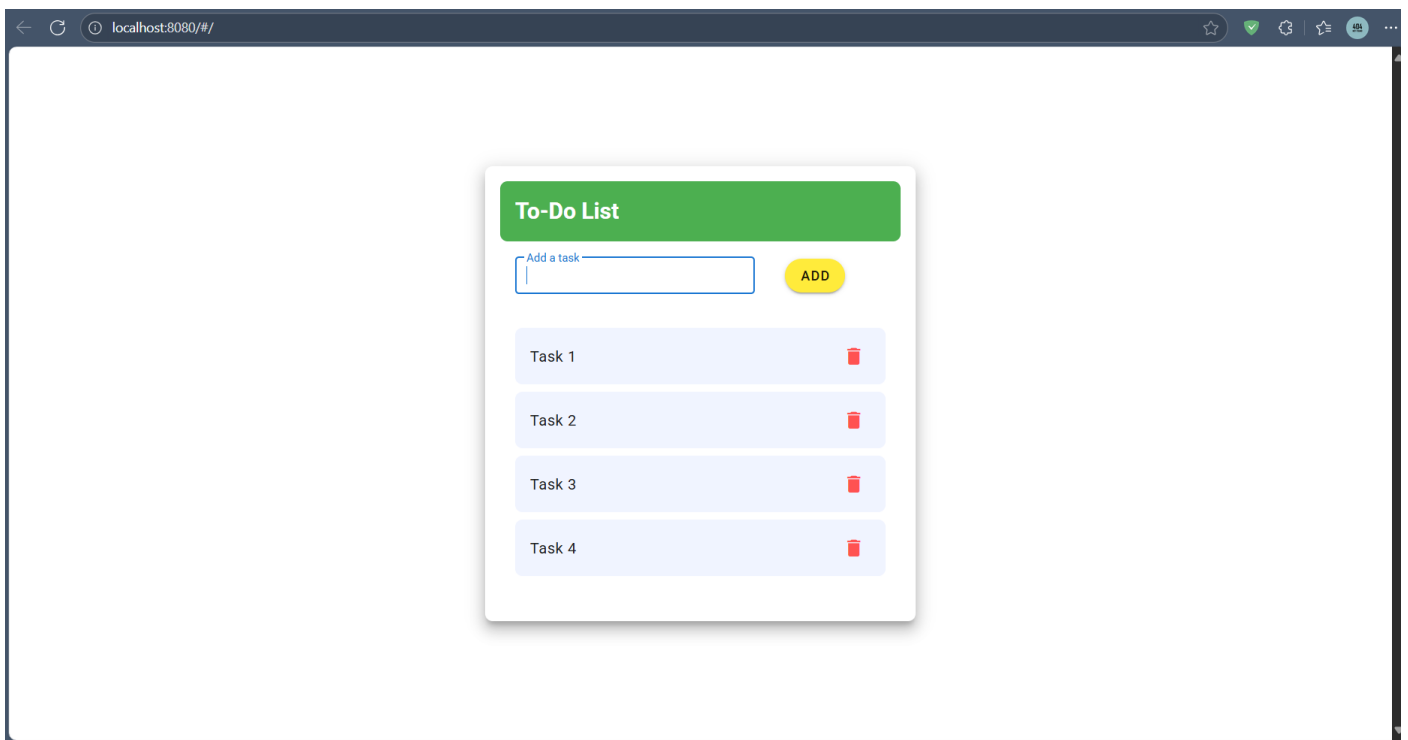
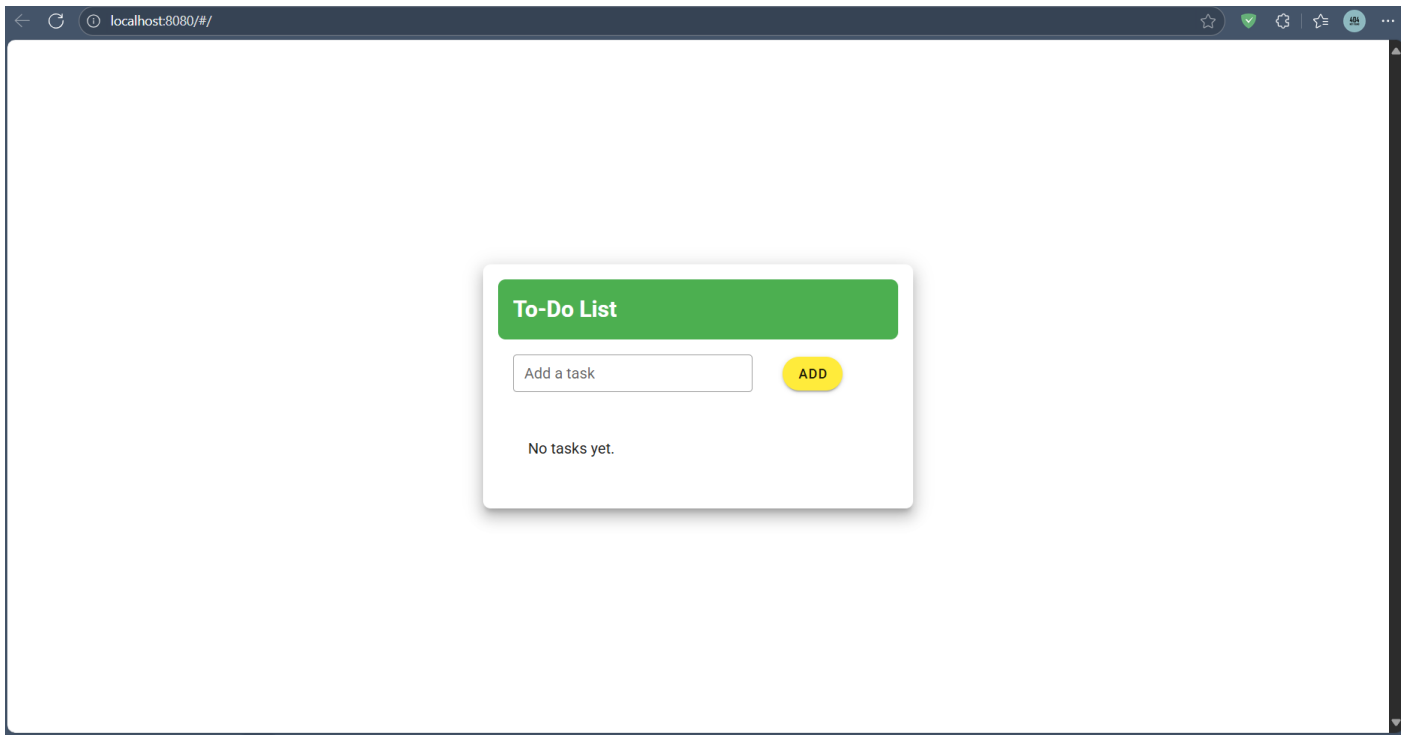
Vue help keep track of changing data in app. Two basic tools are `reactive()` and `ref()`:

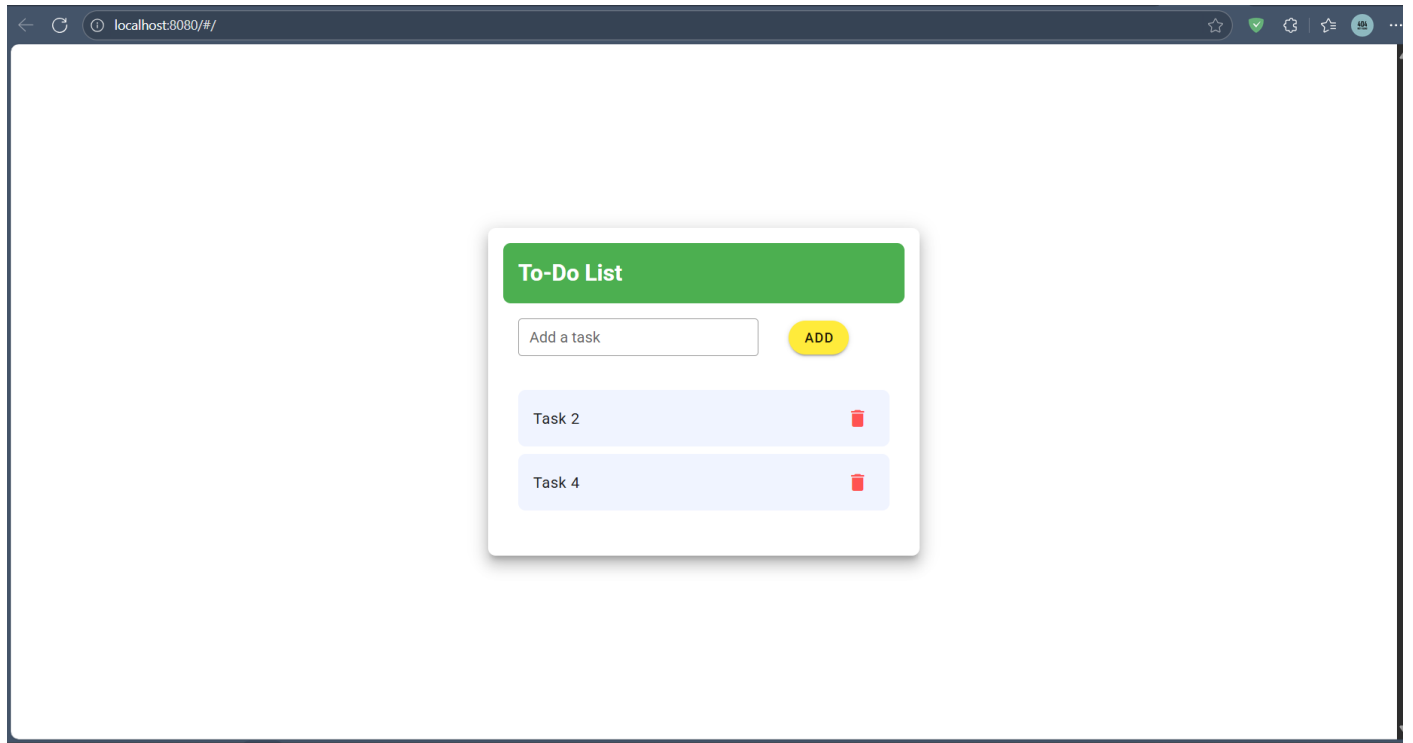
- **ref()** makes a value reactive.
- **reactive()** for objects and arrays.

### Brief Introduction to Pinia and Vuex

- **Pinia:** The new recommended way for state management in Vue 3. It is simple and easy to integrate. Provides a central “store” for all shared data and logic.
- **Vuex:** The older solution for state management. Still works well but Pinia is now preferred since it is simpler and works best with Vue 3.

## 6. Hands on Example





GitHub Repo - <https://github.com/osanda-r/Vue-To-Do-List>