Vue.js Training Program (2 Weeks)

Week 1 – Foundations

Day 1 – Introduction to Vue.js

Why Learn: Understand Vue’s reactivity system and its role in modern front-end development.

Tasks:

- Set up Vue project (via CDN & Vite).
- Explore folder structure.
- Create first Vue component.

Day 2 – Vue Instance & Data Binding

Why Learn: Learn how Vue binds data between UI and logic.

Tasks:

- Use data(), methods, and computed.
- Demonstrate interpolation and binding.
- Build a simple counter app.

Day 3 – Directives & Event Handling

Why Learn: Directives (v-if, v-for, v-model) power Vue’s dynamic behavior.

Tasks:

- Use conditional rendering (v-if, v-show).
- Loop over arrays with v-for.
- Add form handling with v-model.

Day 4 – Components (Part 1)

Why Learn: Components make code reusable and modular.

Tasks:

- Create and register components.
- Pass data with props.
- Emit custom events to parent.

Day 5 – Components (Part 2)

Why Learn: Learn advanced component communication.

Tasks:

- Use slots for content distribution.
- Build parent-child component communication.
- Create a reusable button component.

Day 6 – Vue Router (Part 1)

Why Learn: Routing enables SPA navigation.

Tasks:

- Install Vue Router.
- Define routes & navigation links.
- Create Home and About pages.

Day 7 – Vue Router (Part 2)

Why Learn: Handle dynamic routes and navigation guards.

Tasks:

- Implement route params (/user/:id).
- Add navigation guards for protected routes.

⍰ Week 2 – Advanced Features

Day 8 – State Management with Pinia

Why Learn: Manage state across components.

Tasks:

- Install and configure Pinia.
- Create global store for user/session.
- Use getters and actions.

Day 9 – API Integration (Axios)

Why Learn: Connect Vue apps with real backend APIs.

Tasks:

- Install Axios.
- Fetch data from public API.
- Display data in component.

Day 10 – Forms & Validation

Why Learn: Handle real-world user inputs.

Tasks:

- Build form with v-model.
- Add validation with VeeValidate/Yup.

Day 11 – Lifecycle Hooks + Composition API

Why Learn: Understand component lifecycle and new Vue 3 features.

Tasks:

- Explore onMounted, onUpdated, onUnmounted.
- Rewrite a component using Composition API.

Day 12 – Error Handling + Best Practices

Why Learn: Ensure stability and clean code.

Tasks:

- Handle API errors with try/catch.
- Add error boundary component.
- Apply folder structure best practices.

Day 13 – Testing with Vitest/Jest

Why Learn: Reliable apps need automated testing.

Tasks:

- Write unit test for a Vue component.
- Mock API call in test.

Day 14 – Deployment + Capstone Project

Why Learn: Deploy a Vue app and finalize training.

Tasks:

- Deploy to Netlify/Vercel.
- Capstone Project: Build a Task Manager / Blog App with Vue Router, Pinia, Axios (CRUD), auth mock, and deployment.