

# Vue.JS, Vuetify and Vite, Nuxt JS, Testing Library BOOTCAMP

## VUE-201

### OVERVIEW

20 half days for freshers and developer with less than 2 years of industry experience

1 half day = 4 hours

Vue.js (Vue for short) is a lightweight frontend framework, offering easy two-way data binding, and a reactive UI, all from a tiny code footprint. Vuex state management tool used in Vue applications. The version of Vue covered is Vue 2. An introduction to composition API of Vue 3 is also provided at the end. An introduction to Nuxt JS is also provided.

In this bootcamp you will learn about the Vue Single Page Application (SPA) framework, and how to build a fast, flowing web UI with it.

This training covers JavaScript and TypeScript and does not have JavaScript / TypeScript as a pre-requisite.

### PREREQUISITES

- Basic HTML, CSS knowledge
- Bootstrap knowledge is a plus, but not necessary

### CAPSTONE PROJECT

At the end of this bootcamp, participants will build a workshops application using Vue and Vuetify. A backend server shall be provided. The app will list workshops, details of workshops, their list sessions, and allow authenticated users to post sessions, and also upvote/downvote on sessions.

## LIST OF SOFTWARE TO BE INSTALLED BEFORE TRAINING BEGINS

1. Git CLI on participant systems and GitHub account should be created for every participant (to be created individually by participant). The **GitHub account should be a personal one** and not one associated with the company's GitHub account (I will not be able to add a company account as collaborator on my repositories, and hence shall not be able to share code).

Git CLI download: <https://git-scm.com/downloads>

GitHub link for account creation: <https://github.com/join?source=header-home>

**Once an account is created by everyone, the list of GitHub user names needs to be shared with me - I will add them to the GitHub repository before start of training.**

2. Node.js needs to be installed on all systems – Mac OSX, Linux and Windows is supported. The 20.x.x (LTS version) may be installed. This will also install npm.

Node.js <https://nodejs.org/en/download/>

3. Download and install Visual Studio Code (VSCode)

<https://code.visualstudio.com/download>

It is available for Windows, Mac OSX and popular Linux distributions.

4. For browser – latest version of one of Chrome or Firefox, **preferably Chrome**. Internet Explorer is not acceptable.

Chrome: <https://www.google.com/chrome/browser/desktop/index.html>

Firefox: <https://www.mozilla.org/en-US/firefox/new/>

5. **Additionally, it would be great if participants have as little restrictions (as permissible) on internet access during the session**

# CHAPTERS AND TOPICS

## Day 1

### Quick Introduction to JavaScript Language Fundamentals

- Variables and Primitive Data Types
- Strict Mode Execution
- Variable Scopes, Scope chain
- Using Arrays
- Expressions, Operators and Operator Precedence
- Control flow - Branching and Looping

### Introduction to Objects

- Object Declaration using Literal Syntax
- Accessing Properties and Methods
- Adding and Deleting Properties

## Day 2

### Introduction to Functions

- Function Declaration and Usage
- Function call context (the *this* keyword)
- Handling Variable Number of Arguments
- Inner Functions
- Callbacks - Passing Functions as Arguments
- Returning Functions
- Higher-order Functions
- Closures

## Day 3

### Basic Introduction to Built-in Classes and Objects in the JavaScript Language

- Function - Functions as Objects, `call()`, `apply()` and `bind()` as Methods of Functions
- Array methods

The Functional Programming paradigm - Array iterator methods

Date methods

JSON

### **Features of ES2015+**

Block-level scoping and the use of let, const

Strings and Templating

Default Parameters

### **Day 4**

### **Features of ES2015+ (continued)**

Rest and spread operators

Object and Array Destructuring

Arrow Functions

Classes, Class Inheritance

Modules

Promises

Async..await

### **Day 5**

### **Introduction to TypeScript**

Why Typescript?

Features of TypeScript

Installation and getting started

The tsc compiler options and configuration using tsconfig.json file

Support in popular IDEs

DefinitelyTyped and using type definition files

Using the TypeScript Playground

## Basic types

- Primitive types and the any type
- Static type checking and type inference
- Type assertion (type-casting)
- Arrays
- Tuples
- Type Aliases
- The enum type
- Union types
- Intersection types

## Functions

- Defining function argument and return types
- Function signatures involving callback functions
- Using interfaces to define function signatures
- Overloaded function types

## Day 6

### Interfaces

- Using interface to define structure for an object (properties and methods)
- Extending an interface
- Implementing interfaces in classes

### Classes

- Access modifiers – public, protected, private and readonly
- Shortcut syntax for creating properties via constructor

### Generics

- Creating a generic class
- Using Generics

Generic type aliases  
Generic type defaults

## **Miscellaneous topics**

Creating and using Decorators  
Introduction to creating type definition files  
Using libraries that don't have type definition files

## **Day 7**

### **Introduction to Vue**

What and Why  
Installation - Gearing up for Vue application development  
Scaffolding an application using Vue CLI  
Understanding the project structure  
Understanding the build process  
More about the Vue CLI  
Linting using ESLint  
Configuring ESLint rules  
How it works internally - basic introduction to Webpack  
Entrypoint, setting up bundles  
Webpack development server  
Module loaders for TS, Babel, Sass  
Using Webpack plugins

### **Vue Basics**

How the Vue compiler and reactivity works  
Lifecycle hooks  
Introduction to the running example: Workshops Application  
Displaying a product

## Day 8

### Vue Basics (continued)

Writing filter functions

### Adding Interactivity

Binding to DOM events

Using computed properties

Directives: v-show, v-if, v-else, v-else-if

### Forms and Inputs

Using v-model binding

Binding values to check boxes and radio buttons

The v-for directive

Modifiers: .number, .lazy and more

## Day 9

### Conditionals, Looping and Lists

Showing list of products

Adding star rating with v-for range

Binding classes

Sorting records

### Working with components

Creating components

Global registration

Local registration

Relationships between components

Using props to pass data

Prop validation

## Day 10

Defining a template component - inline strings, template in script tag, and using single file components

Working with custom events – Listening to events, modifying child props using .sync

## Advanced Components and Routing

Working with slots – named slots, scoped slots

Async components

Adding Axios

Setting up routes

Router links

Adding routes with parameters

Using redirection and wildcards

## Day 11

**Case study:** Workshops Application development (Vue, Vue router, Vuetify, Authentication and Authorization)

## Day 12

### Extending Vue

Reusing functionality with mixins

Custom directives – modifiers, values and args

Render functions and JSX

**Case study:** Workshops Application development (Vue, Vue router, Vuetify)

## Day 13



## State Management

When is Vuex helpful?

Getting started and injecting the store into the app

Vuex state

Getters and Dynamic getters

Mutations

Actions

Vuex helpers

Vuex modules and namespaces

## Day 14

**Case study:** Workshops Application development (Vuex)

## Day 15

## Debugging

How to go about looking for errors

The Vue developer tools

Using the developer tools for Vuex

## Deployment

Configuring the app for different environments

Deployment options

Deployment on AWS (S3/EC2)

**Case study:** Workshops Application development (Vuex, Deployment on AWS)

## Day 16

## Unit Testing

What is unit testing and why it is required

Configure Jest in application

Writing a test – describe(), it()/test(), beforeEach()/afterEach(), beforeAll(), afterAll()

Test coverage

Vue Test Utils

How Vue Testing Library helps

Component tests

render, screen, getBy\*(), findBy\*(), queryBy\*() methods – when to use which methods

Jest DOM for DOM matchers, User event library

Testing services – using MSW for mocking API calls

**Case study:** Workshops Application development (Unit Testing)

## Day 17

### Miscellaneous topics

Problem with mixins

Vue 3 composition API – setup(), ref(), reactive(), computed(), watch(), lifecycle related methods

How composition API is to be used for code reuse between components

## Day 18

**Case study:** Workshops App using Unit Testing and Vue 3

## Day 19

### Nuxt JS

What is Nuxt.js

Features of Nuxt.js

Creating a Nuxt app

Directory structure

Setting up route

Setting up dynamic routes

Setting up meta tags in the document head

Overview of server-side rendering (SSR) works and how it works in Nuxt

Overview of static site generation (SSG) works and how it works in Nuxt

When to use SSR vs SSG

## Day 20

**Case study:** Workshops Application development (Nuxt JS)