

**React 17**

Vivekanand P V

**Aim:**

* To gain working understanding of the modern component-based front-end design using React.js ecosystem
* To learn the best-practices of front-end design and JavaScript development
* To learn the industry-standard development practices
* To learn state-management using Redux

**Duration:**

5 days (standard content)

**Prerequisites:**

* Hands-on experience in HTML, CSS, and JavaScript
* Understanding of software engineering principles
* Knowledge of standard design patterns is a bonus

**Essential Reading:**

The aim of essential reading is to bring participants quickly to the pace. Participants are required to go through these quick guides before attending the training.

1. JavaScript basics: <https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/JavaScript_basics>
2. DOM (Document Object Model) basics: <https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model/Introduction>
3. Arrow functions: <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Functions/Arrow_functions>
4. Array API: <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array>
5. Promise API: <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Promise>

**Lab Setup:**

|  |  |
| --- | --- |
| **Classroom Training** | **Virtual Training** |
| * HD-quality projector or screen with HDMI connectivity * Whiteboard and markers | Access to (one of):   * Zoom Meeting * Skype * Google Meet (Hangout) * Microsoft Teams |

**Environment Setup:**

* Access to internet and Npm repository
* Please refer to the developer environment setup guide
* Access to GitHub so that participants can clone/download/refer to sample and reference code
* Access to React documentation and other online resources

**Content**

**Day 1**

**Session 1**

JavaScript fast fundamentals

Functions, scopes, hoisting, let, and const, Arrow functions

Objects and encapsulation and this keyword

Destructuring assignment and spread syntax

ES6 classes

TypeScript as a superset of JavaScript

Type annotations and type assertions

Type safe programming practice

TypeScript classes and OO programming

Introduction to generics

Interfaces, type, and enums

ES6 module import system

**Session 2**

Introduction to component-based front-end development

Project anatomy

React components and composition

Data binding

Embedded expressions

JSX

Component state

Hands-on exercises

**Day 2**

**Session 1**

Conditionals

List-rendering

DOM-events

Custom events

Props

One-way binding

Component composition

Content projection

Hands-on exercises

**Session 2**

Virtual DOM and rendering

Life-cycle methods

Class components

Working with styles

React hooks

Advanced hooks

Render props

Exercises

**Day 3**

**Session 1**

Routing

Route props

Route definitions

Parameters in routing

Redirection and navigation

Nested routes

Programmatic navigation

Route organization

Lazy loading

Hands-on exercises

**Session 2**

Introduction to Promise API

Http communication using Axios

Working with Http headers

Consuming RESTful webservices

Working with Forms

Controlled components

Refs and native elements

Component state-management techniques

Exercises

**Day 4**

**Session 1**

Introduction to state-management

Flux architecture

Pure functions and currying technique

Reducers

Actions and action creators

Immutable structures from Immutable.js

Why immutability in state-management?

Store

Dispatching actions from components

Subscription to state in components

**Session 2**

Introduction to Redux workflow

Redux Toolkit

Slices

Working with redux in function components

Working with redux in class components

Redux middleware

Asynchronous actions with Redux thunk

Creating custom middleware

Hands-on lab

**Day 5**

**Session 1**

Live project (2 hours)

Presentations and Discussions

Building React apps

Optimizing the build process

Dockerizing the React apps

Hosting React apps in Nginx

**Session 2**

Introduction to unit testing

Testing components

Jest api and matchers

Testing services and stubbing

Emulating user interaction

Test harness and automating tests

Testing asynchronous code

Organizing tests

Coverage

Practical tips on component testing

Q & A

Further reading

Conclusion