# JAVASCRIPT FRAMEWORK AND APACHE SCALA TEST DRIVEN DEVELOPMENT

By Dr. Vishwanath Rao

Day 1

JEST AND ENZYME

Class Architecture

React with create-react-app
Enzyme
Jest
Node
VSCode
Other tools

Test Application
Front-end with React
Simple back-end
Focusing on unit testing components
Starting with Jest

Using Jest/Enzyme to validate accessibility requirements
Running tests under Jest
Jest Testing Outline
Discover()
It() / test()

Running code before and after tests
Matchers
Reporters
Code Coverage
What is code coverage?
Snapshots

Test Mocking
Unit testing vs integration testing
Mocks help maintain boundaries
Mocking components
Mocking functions

Asynchronous functions
To mock, or not to mock?
Enzyme and shallow rendering

Role of Enzyme
Enzyme vs Jest, Enzyme with Jest
Enzyme Configuration

Shallow rendering of a component
Full rendering
Unit vs integration testing re-visited

When should I use full rendering?
The full API vs the shallow API
Accessing subcomponents in a full render

Day 2

Styling in JavaScript with Styled-Components
Packaging an App with webpack
Getting Started with Styled-Components
Testing Styled Components

Refactoring with Higher-Order Components
Making Higher-Order Components
The Controllable Pattern
Stacking Higher-Order Components
Inspecting Components with React Devtools
Mantra: Keep the Unit Small

CHAI AND MOCHA

Introduction to Chai and Mocha Frameworks

install the Chai library on your local machine

use assertions to test boolean values

compare numeric values using assert

Day 3

performing string comparisons using assert

describe the assertions that can be used to test objects and their properties

use arrays and elements in arrays using assert

compare and contrast the should and expect APIs in Chai

test different data types using should

test different data types using expect

perform asynchronous function testing using expect

Introduction to Sinon

What is Sinon?
Sinon - Spies
Sinon - Stubs
Sinon - Mocks

Day 4

SCALA TEST

Setting up the Project

Why Testing?
Why ScalaTest?
What You Will Learn
Course Prerequisites
How This Course Is Structured
How to Access Project Source Code
Overview of Course Project

Writing the First Test

What Is a Test?
Writing the First Test
Running the Test
Running the Tests Using Scala Command Line
Running the Test Using Sbt Command Line
Running the Test Using Sbt Console
Understanding Testing Styles
Understanding FlatSpec Style
Understanding the Test Lifecycle

Working with Assertions

Understanding Assert in ScalaTest
Writing Tests with Assert
Writing Tests with AssertResult
Writing Tests with AssertThrows

Failing the Tests Unconditionally
Canceling a Test if PreCondition Not Met
Adding Information to Test Failures

Performing Expressive Testing Using Matchers

Understanding Matchers
Using Matchers to Test Equality
Using Matchers to Test Strings
Using Matchers to greaterThan or lessThan
Creating a Base Test for Unit Tests
Using Matchers to Test for Length and Size
Using Matchers to Test Container Elements
Using Matchers to Test Emptiness Property

Day 5

Using Matchers to Test Exceptions
Using Matchers to Test Logical Operations
Using Matchers to Test Negative Statement Structure
Using Matchers to Test Object Identity
Using Matchers to Test Pattern Matching

Generating Data through Fixtures and Asynchronous Testing

Understanding Fixtures
Working with Fixture Object
Working with Fixture Context Object
Working with Loan Fixture Method
Working with withFixture Method
Working with beforeAndAfter
Running Your Tests Asynchronous

# Mocking and Tagging Your Tests