

why learn fullstack?

A fullstack web developer is someone who is able to work on both front-end and back-end. Fullstack developer is the highly demanded job in the market. This course helps to learn all the top notch development technologies. AngularJS is the top most framework used for front-end and nodeJS is always a best solution for server side programming, thus learning these technologies always make you stay ahead.

What you will learn in the program ::

As part of the course you will have a hands on experience over Full stack Concepts, In-depth knowledge about developing Front-end and Back-end, confidence to develop your own project and enhance your coding skills, vital development skills such as setting up server in Node js and working with ExpressJS. In the Front End, you can opt for Angular JS instead of React JS, and in the back end, you can opt for Python instead of Node JS.

What you can do after this course:.(future)

Large and small organizations depend on Full Stack Web Developers to build, maintain, and support their web applications. As a Full Stack developer, you will build superior applications using Full stack. This course is specially designed by our expert faculty to give you the edge to build fine quality applications. You will gain expertise in MongoDB, Express.js, Angular, Node.js, JavaScript, HTML, and CSS.

What you need to know??

- No prerequisites required, HTML, CSS, JS will be added advantage.

before you start:

Are you passionate about developing applications using various emerging technologies. Are you curious to know how these applications work under the hood. If you are interested to learn programming and scripting languages to develop front-end and back-end then full stack is the right choice.

HTML and CSS

Introduction to html and css
inline css,internal css,external css,
basic structure of html
attributes
elements
paragraphs
headings
css selectors, combinators
links
comments
basic template using 3types of css
box-model
outline
floats
border
text formatting
text,fonts,colors
images
backgrounds
block level elements
inline elements
Units px , em , percentage
px to percent (%)
display,inline block
Tables
PSD to HTML , CSS
Assign a psd template
opacity
iframes

week 2:

positions
pseudo-classes
pseudo-elements
responsiveness
layout
forms
border-radius
design a login and registration forms
migration
new elements
new form elements, and attributes

html5 validations
box-shadows
audio tag
video tag
media queries
box-sizing
2d-transforms
3d-transforms
transitions
task on transforms and transitions
animations
Gradients
Assign a psd template 2

JAVASCRIPT

Week 1:

Introduction to Javascript
How to execute Javascript code
Javascript Versions ES5 Vs ES6
Javascript Execution Engine
How JS code gets Executed
Javascript Future Scope
When and where to use Javascript

Language Basics
Variables
Variable Hoisting
Null, Undefined and NaN
Variable - Objects
Functions
Creating and calling them
Function arguments and return data

Types and Scopes
Primitive and reference Types
Global and local scopes

Arrays
Basics and managing Elements
forEach() Method

working with array methods like push, pop, shift and unshift
working with parts of arrays like indexOf, slice and splice method
Filtering, mapping and reversing array elements
difference between concat and join

Objects

Object basics and Literal Notations

Property 'this'

alternative ways of creating object

Object.create();

prototypes

prototype in action

constructor function with and without args

handling 'this' with bind, call and apply

Functions

Closures

IIFE's

Built-in methods and properties

Week 2:

Built-In function

String Functions

Math function

date object

DOM Manipulation

Window Object

Location Object

Selecting Element

creating and inserting Elements

Deleting Elements from DOM

Events

Event Objects

Event Handlers

Event Listeners

HTTP Request (AJAX)

Setup and sending a get request

Post Request

Promises
Regular Expressions
Basics of Typescript
Difference between ES5 and ES6

TYPESCRIPT:

WEEK 1:

Introduction
What is Type script
Why Typescript and how to use it
Installing typescript
Using Types for the better code
Type Basics
Number and Booleans
Assigning types Explicitly
Array and types
Tuples
Enums
The 'Any' type
understanding the created js code
using types in functions(Arguments and return values)
Function as types
Object as types

Putting all together in a complex object
Creating custom types and type aliasing
Allowing multiple types and unions types
The 'NEVER' type
Understanding the TypeScript Compiler
How code gets compiled
changing compiler behaviour on error
Debugging your ts code using source map

TypeScript and ES6
let and const
Block scope
Arrow function
Using Classes to create Objects
creating classes and class properties
class methods and access modifiers

Inheritance

Inheritance and constructors
static properties and methods
Abstract classes

Namespaces and modules
Introduction
Namespaces and multiple files
Namespace imports

WEEK 2:

Doing contract work with interfaces
Basics about interfaces
Interfaces and properties
Interfaces and methods
Interfaces and classes
Interface inheritance

Generics
Why and what?
creating a generic function
A built-in generic types:array
using generic types
creating a generic class

Typescript and workflows
using tsc and tsconfig file
wrap up

angular 5:

week 1:

what is Angular ?
Project setup and first app
Basic project setup and using bootstrap for styling
How an Angular App get loaded and started
Components are Important
Creating a new component
understanding the role of appmodule and component declaration
using custom components
working with component templates
working with component styles
fully understanding the component selector

What is Data binding?
String interpolation
Property binding
Property binding vs string interpolation
Two way data binding
Important : formsModule required for the two-way databinding
understanding directives
using the *ngIf
using ngClass
using *ngFor
getting the index value using the *ngFor

week 2:

Components and data Binding Deep Dive
Splitting into Components
Property and Event binding Overview
Binding to custom Properties
Assigning a Alias to custom properties
Binding to custom Events
assigning a alias to custom events
More on view Encapsulation
Using Local reference in templates
Getting access to template and DOM using @ViewChild
Understanding component LifeCycle
Routing
Why do we need a Router
setting up and loading routes
Navigating with router links
Understanding Navigation paths
Styling Active router Links
Passing params to routes

week 3:

Template Driven Forms
Creating a form and registering the controles
Submitting and using the form
Understanding Form state
Accessing the form using viewChild
Outputting validation error messages
Set Default with NgModel property binding
Using NgModel with two-way data binding
Grouping form controls

- Using form data
- Assignment on Template Driven Forms
- Reactive Forms
- Creating a form in component
- Syncing HTML and Form
- Submitting the form
- Adding the validation
- Getting access to controls
- Grouping controls
- arrays of form controls
- Assignment on Reactive Driven Forms

week 4:

- Using Services and Dependency Injection
- why do we need a service
- Creating the Logging Service
- Injecting Logging Service into components
- Creating Data Services
- Understanding Hierarchical Injector
- How many Instances of Service Should It Be?
- Injecting the services into services
- using services for the cross component communication
- Observables
- Building and using a simple observable
- Building and using a custom observable from scratch
- .map
- .subscribe
- Making the http requests
- Introduction and how the http requests work in the spa's
- Http calls using GET,POST,PUT methods
- setting up the firebase for dummy backend
- sending put request for SAVE the data
- GETting back the data

Node Curriculum

Week 1

- Introduction to Node Js
- Installing Node
- What is Node

- Why should we use Node
- Hello World
- Requiring Modules & Own Files
- Event Loop & Event Emitter
- Buffers & Streams
- Debugging Node.js Applications
- Debugging via chrome dev tools
- Async
- Promises
- Introduction to Express
- Hello Express
- Rendering Templates with data
- Express middleware

Week 2

- Passing Data & Handling
- Test cases for Application
- Testing the Application
- MongoDB
 - Introduction to MongoDB
 - Installing MongoDB
 - Creating a database
 - Connecting with Mongo & Writing Data
 - Fetching Data
 - Updating Data
 - Deleting Data
- Mongoose ORM
 - Setting up Mongoose
 - Validators, Types and Defaults
 - Mongoose Queries & Validation
 - Creating Model Schema
 - Inserting Data
 - Fetching Data
 - Updating Data
 - Deleting Data

Week 3

- Postman
- Registration & Login
- Private Routes & Public Routes
- Hashing Values
- Generating auth tokens and setting headers
- Handling Sessions
- Data Validation using Third Party Modules

Week 4

Async/Await

Sending Mail from Server using Nodemailer

Uploading Files with Multer

Passport Authentication

Social Login with Passport

Socket.io

Introduction

Creating a project

Adding socket.io to an app

Emitting and listening to events

Emitting and listening to events(solution)

Broadcasting events

Message generator and tests

Event acknowledgement

Projects:

1)Jobs portal

In this project, we have two end users one is Job Seeker and Recruiter. The Jobseeker applies for a job posted by all recruiters. The Recruiter will post job requirements and get all details of jobseekers who applied for that job posted by him/her.