

# Fullstack

# Development

6 Months Programme





## Introduction To Web Technologies

Need of creating a website  
Technologies to learn and use of it  
Need of text editors (VS Code)  
Creating a Project in VS Code  
Structure of HTML  
HTML Tags/Elements & Attributes

## Working with Text Elements

Heading Tags  
Paragraph tags  
Formatting tags  
Quotations & Comments  
Mark Tag, sup, sub, Progress,  
Iframe tag  
Entities & References

## Introduction to CSS

Types of CSS  
CSS Syntax  
Applying CSS to text elements  
Adding class and Id  
Adding a Custom font

## Adding Images to Webpage

Adding image via HTML  
Adding background image  
Resizing images using CSS  
Adjusting image position

## Working with links

Creating links and attributes  
Creating mail link  
Creating a website by  
linking pages

UnOrdered list  
Definition list

## Working with Pseudo Classes

Link, Visited, Hover, Active  
nth-child  
first-child, last-child

## Working with Pseudo Elements

first-line  
first-letter  
before and after



## HTML5 Semantic Elements

- Article
- Aside
- Section
- Header
- Footer
- Nav
- Figure
- Figcaption
- Summary

## Working with Multimedia Tags

- Audio
- Video

## Play with Tables

- Understanding Rows & Columns
- Creating a mock table
- Styling my Table
- Working with Rowspan
- Working with Colspan

## Understanding HTML Element Types

- Block Level
- Inline element

## Styling Div Elements

- Understanding Box Model in CSS
- Adding Padding and Margin
- Adding Borders and Border Radius
- Applying Shadow props
- Working with outlines
- Working with Position

## Applying Background Color

- Linear Gradient and Radial Gradient
- Opacity
- Overflow-x and Overflow-y

- Flex-direction
- Flex-flow
- Flex-wrap
- Align content & Justify Content
- Align Items

## Creating Responsive Web Design

- Media queries
- Working with Animation
- Working with Transition

## **Javascript – The BACKBONE**

Introduction to Javascript

Code editors

Developer console

Syntax & Statements

Variables & DataTypes

Code structure

Type Conversions

Interaction: alert, prompt, confirm

## **Understanding DOM**

Changing HTML Elements using JS

## **JavaScript Fundamentals**

Conditional branching: if, '?'

Switch Statements

Loops

Functions Basics

Arrow functions, the basics

Comments

## **Advanced working with functions & Objects**

Variable scope, closure

Objects

Object methods, "this"

Call/apply

Function Prototypes

## **Datatype and its functions**

String Functions

Number Functions

Math Object

Array Functions

## **DOM Manipulation**

Add / Remove classes using Javascript

Working with Events

DOM Events

## **Network requests**

Understanding AJAX

Components of AJAX

XMLHttpRequest

Working with Status Codes

JSON Vs XML

Loading Mock data into UI

Designing e-commerce

layout using AJAX



## Need of jQuery

Why jQuery is introduced?  
Difference between-  
Javascript & jQuery  
How easy is jQuery?  
Prerequisites for jQuery  
Download jQuery library  
CDN links  
Need of noConflict

## Introduction

jQuery syntax  
Working with events  
Understanding selectors & Recap  
Program using jQuery and JS  
Writing CSS using jQuery

## Animations using jQuery

Show/Hide  
Fade In/ Fade Out  
Slide Up/Slide Down  
Animate

## Getters and Setters

Get the DOM element value  
Set the DOM element value  
Adding an element  
Removing an element  
Working with addClass &  
removeClass  
Need of Callback  
Use of Chaining

Siblings Selectors  
jQuery Filter Methods

## AJAX using jQuery

How is it different from JS?  
Building e-Commerce layout  
jQuery UI discussion

## Creating a Project

Small Application using jQuery  
Practise programs



## Introduction to Bootstrap

Need of learning Bootstrap  
Download ready-to-use code  
Media Queries Vs Bootstrap  
What's there and what makes it easy?  
Creating responsive templates  
Adding Bootstrap to Project

## Grid System Layout

Need of a container & container-fluid  
Bootstrap 3 Vs Bootstrap  
Understanding rows and  
Bootstrap Classes and Breakpoints  
Auto layout Columns

## Bootstrap Content

Typography  
Images  
Tables  
Figures  
Forms

## Bootstrap Components

Creating Buttons with contextual Classes  
Badges & Labels

## Cards

Spinners and Tooltips  
Progress  
Toasts & Alerts  
Breadcrumb  
Pagination

## Advanced BS

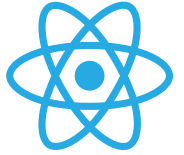
### Components:

Creating Carousel  
Popover

Collapse  
Scrollspy

## Creating a Responsive Website

Template Creation 1  
Template Creation 2



## Getting Started with React

- Create a New React App
- Installing dependencies
- Understanding React Architecture
- Running the project in localhost

## Integrating with Other Libraries

- Understanding Package.json
- Installing Material UI
- Installing React Bootstrap

## Basic Concepts

- Printing Statements from JSX
- What are Components and its types?
- Props
- State and setState

## React Forms

- Working with forms and form validation

## Event Handling

- Different ways of binding events

## Life Cycle Methods of React

- Initializing Phase
- Mounting Phase
- Updating Phase
- Unmounting Phase

## React Routing

- Creating Route and Navlinks

## Handling Errors

- Working with Error Boundaries

## Higher Order Components

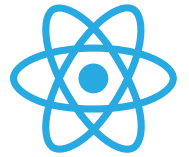
- Creating a common function to share among multiple components

## Creating Context

- Creating Context with Provider and Consumer

## Working with Ref's

- Ref and forward ref's



## Working with Axios

Handling HTTPS Requests  
Deriving data from API  
and populate on UI

## Creating a Project POC

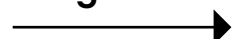
React Project 1 from reactjs.org  
Hosting a React Proj using Netlify

## Introduction to React Hooks

Need of learning Hooks  
Using the State hook  
Using the Effect hook

Talking about React, it is actually the best library developed and maintained by Facebook that mainly focuses on building and rendering components for a web page. It is useful for creating web applications like Facebook, Instagram etc. React will efficiently update your web page by rendering the right components. React is most widely used by Facebook, Airbnb, Uber, Netflix, Instagram, Whatsapp, Dropbox.

Angular







## Introduction to Angular

What is Angular JS?  
Need of migrating to  
Angular 2+  
Current Version of Angular  
and updates

## Typescript Features

Getting Started with TypeScript  
Working with tsc  
TypeScript Features  
TypeScript Classes  
Function: Inferring the  
Interfaces  
Generic Types  
Generic Classes  
Generic Constraints  
Decorators  
Property, Class, Parameter  
Decorators  
Exports  
External Modules  
Namespace & Mixins

## Software Requirements

Setting up Environment  
Creating first angular app

## Understanding Component

Basics of component  
Component Decorator  
Component Decorator details  
Creating Components  
Passing data to a Component  
Responding to Component Events  
Component Lifecycle  
Advantage and limitations  
of component

## Forms in Angular

Two way data binding with ngModel  
User Input  
Form Controls in Angular  
Controls and Control Groups  
Form & NgForm  
Validation & Error Handling  
CSS-Forms

## **Dependency Injection in Angular**

Dependency Injection  
Why dependency injection?  
Dependency Injection Usage  
Configuring the injector  
Injectable Decorators  
Dependency Injection Tokens

## **Introduction to Directives**

Attribute Directives  
Difference between Directives and Components  
Create new Attribute Directives  
Structural Directives  
Creating Structural Directives  
Directive Metadata  
Directive Types

## **Pipes**

Chaining pipes  
Parameters in Pipe  
Custom Pipe  
Async Pipes  
Built In Pipes

## **Routing & Navigation**

Basic introduction to Router  
Route configuration  
RouteConfig,RouterLink, RouterOutlet  
Child Routes

## **Services, Providers**

Services in Angular  
Injecting the HeroService  
An Overview on Providers  
Example on Services

## **Events in angular**

Event Emitter Class  
Emit Method & Next Method,  
Subscribe Method

Observables

The RxJS Library

Observables in Angular

Practical Usage

Angular, It is a JavaScript framework maintained by Google, which is client-side and browser-based. It is actually like all in one framework unlike react. Angular is basically built on TypeScript which is a superset of JavaScript. Even with Angular we will get a very good support during our dev phase. Angular is used widely by most popular brands like Google, Nike, Forbes, Upwork, General Motors, HBO, Sony.



## Core Python

### Introduction

- History
- Features
- Setting up path
- Working with Python
- Basic Syntax
- Variable and Data Types
- Operator

### Conditional Statements

- If
- If- else
- Nested if-else

### Looping

- For
- While
- Nested loops

### Control Statements

- Break
- Continue
- Pass

### String Manipulation

- Accessing Strings
- Basic Operations
- String slices
- Function and Methods

### Tuple

- Introduction
- Accessing tuples
- Operations
- Working
- Functions and Methods

### Dictionaries

- Introduction Accessing values in dictionaries
- Working with dictionaries

### Functions

- Defining a function
- Calling a function
- Types of functions
- Function Arguments
- Anonymous functions

### Global and local variables

- Modules
- Importing module
- Math module
- Random module
- Packages
- Composition

## **Input-Output**

Printing on screen  
Reading data from keyboard  
Opening and closing file  
Reading and writing files  
Functions

## **Exception Handling**

Exception  
Exception Handling  
Except clause  
Try ? finally clause  
User Defined Exceptions

## **Advance Python**

### **OOPs concept**

Class and object  
Attributes  
Inheritance  
Overloading  
Overriding  
Data hiding

### **Regular expressions**

Match function  
Search function  
Matching VS Searching  
Modifiers  
Patterns

### **CGI**

Introduction  
Architecture  
CGI environment variable  
GET and POST methods  
Cookies  
File upload

### **Database**

Introduction  
Connections  
Executing queries  
Transactions

### **Networking**

Socket  
Socket Module  
Methods  
Client and server  
Internet modules

### **Multithreading**

Thread  
Starting a thread  
Threading module  
Synchronizing threads  
Multithreaded Priority Queue

# Benefits of learning at



Experienced Mentors



Regular Workshops



At your own pace



Anytime Doubt  
Clarification



Alumni Connect



Placement Training



Portfolio Assistance



Freelance Training

### What are the merits of being a Full stack developer?

Based on the requirements of the project, a competent Full Stack developer is often able to switch between front end and back end creation. Studies also found that a Full Stack developer can get fresh technologies and tools up - to - date faster than someone who only specializes in front end or back end technology.

### What are their career possibilities?

Each project would require a user interface, and therefore, relative to any other technology, the need for web developers and the necessary job opportunities are plentiful.

### How will this course will add value to my portfolio?

Since web development tools are commonly used in many projects regardless of backend technology, qualification in these courses would have a huge benefit during the recruiting process, and opportunities are large in numbers.

### What are my options if I happened to miss a session?

No need to worry, another session or a recorded video of the missing session will reward you.

### Do I get schedule according to my requirement?

As our intention is to take this in a much more friendly way, absolutely yes.