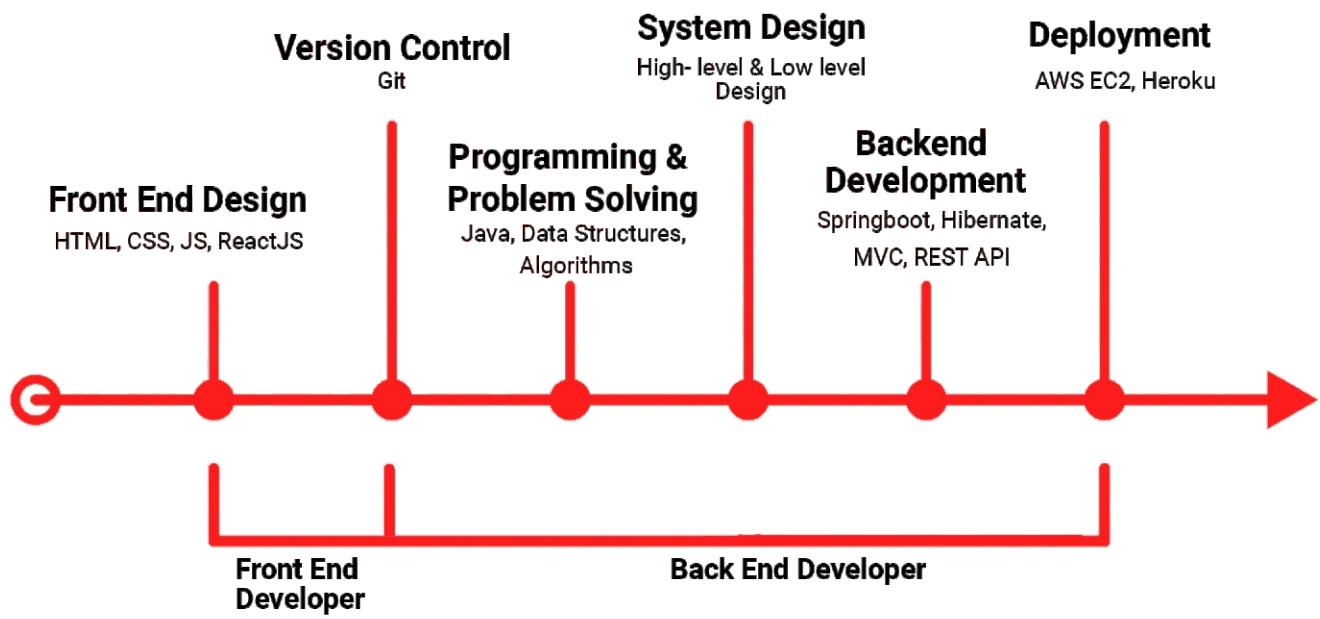


# FULL STACK DEVELOPMENT CURRICULUM



## About this course

This course will serve as a comprehensive introduction to various topics in Software Development. This course is a proper blend of theory and the practical hands on session for each and every concept. Throughout the course participants will work on a complete end to end tech stack in order to implement the concepts learned during the course.

## Program Highlights

- Duration: 5-6 months (Course) and 45 days Internship
- 250+ Hours of Live Lectures
- 100+ Hours of Live Handson Sessions
- 200+ Hours of Assignment and Projects
- 5+ Mini Projects
- 5+ Case Studies
- 3+ Major Projects
- 1 Capstone Project

## Programming Languages and Tools



React



## Learning Outcomes

At the end of the course participants should be able to -

- Build an extensive backend API with Java SpringBoot.
- Protecting routes/endpoints with JWT (JSON Web Tokens)
- Extensive API testing with Postman.
- Integrating React with our backend in an elegant way, creating a great workflow.
- Building our frontend using React to work with backend API's
- Creating the build scripts and securing the keys before deployment
- Get handson experience on Amazon EC2 Cloud Instance

## Prerequisites

There are no prerequisites required for the course. But a little knowledge of programming and HTML, CSS and JavaScript is a benefit for candidate.

## Schedule

NO.	Topic	Total hours
1	HTML, CSS and Introduction to Object Oriented JavaScript – ES6, Classes, JS tips and tricks	20
2	Introduction to React Library – CDN, script and npm packages, Components, functional and class based	30
3	Java – Environment Setup, IDE, operator, expressions, statements, methods, OOPs concepts, arrays and collection, File and Exception Handling	30
4	Databases – Creating a DB using PostgreSQL, CRUD operations	10
5	Problem Solving	60
6	System Design – Low Level and High Level Design	40
7	SpringBoot – Introduction to MVC, MVC and RestAPIs, AOP, ORM and migrations, Testing, Views and Security	40
8	SpringBoot App Deployment and Capstone Project	40

# Module – 1 (Front End Development and Version Control)

## 1. Introduction

- a. Download and install Visual Studio Code (VS Code)
- b. Overview of Full Stack Development Program

## 2. HTML

- a. Inline elements
- b. Block elements
- c. Element Attributes
- d. HTML Forms and Form Elements
- e. Type of Inputs
- f. Buttons
- g. Anchor tags
- h. Images
- i. Video
- j. Meta tags
- k. Miscellaneous Tags (code, blink, table, span, etc.)

## 3. CSS

- a. Cascading
- b. CSS Selectors
- c. Elements, Pseudo and Attributes
- d. CSS Colors
- e. Color Picker and gradients
- f. Cascading Algorithms
- g. Background & Background properties
- h. Box Model
- i. Padding, Margin & Border
- j. CSS Alignment
- k. Flexboxes, Grids and Position

- I. CSS Transformations
- m.CSS iFrames
- n. CSS Animation
- o. Webkit and frames

## 4. Bootstrap

- a. Bootstrap Paragraphs
- b. Bootstrap Alignments
- c. Bootstrap Quotes
- d. Bootstrap Lists
- e. Bootstrap Code
- f. Bootstrap Keyboard
- g. Bootstrap Contextual Classes
- h. Bootstrap Buttons, Links & Submit
- i. Bootstrap Forms
- j. Bootstrap Tables
- k. Bootstrap List Groups & List-Group-Items
- l. Bootstrap Pagination
- m.Bootstrap Labels
- n. Bootstrap Alerts
- o. Bootstrap Panels
- p. Bootstrap Wells
- q. Bootstrap Progress Bar
- r. Bootstrap Grids

## 5. JavaScript

- a. JS Programming Fundamentals (ES6)
- b. DOM Access and DOM Manipulations
- c. DOM Events
- d. JavaScript Classes & OOPs
- e. JavaScript Functions & JavaScript Arrow Functions
- f. Higher Order Array Methods - forEach, map, filter and reduce
- g. JavaScript tips and tricks

## 6. Ajax

- a. Introduction to Client-Server Architecture
- b. Response Notations - XML & JSON
- c. Introduction to APIs
- d. Testing APIs with Postman
- e. Introduction to asynchronous communication
- f. Inserting APIs with Front End

## 7. React

- a. Creating React App
- b. Folder Structure
- c. Understanding React Components
- d. Adding components
- e. Listening to events
- f. Understanding JSX
- g. Creating functional components
- h. Working with props
- i. Getting user inputs
- j. Handling Events
- k. Creating class based components
- l. Component Lifecycle
- m. React Hooks (use State, use Effect, )
- n. Making HTTP requests using Axios.

- o. Creating class based components
- p. Component Lifecycle
- q. React Hooks (use State, use Effect, )
- r. Making HTTP requests using Axios.
- s. State Management
- t. React content API
- u. Third party state management library - REDUX
- v. Redux middlewares like logger and thunk.
- w. Adding forms
- x. Routing in React
- y. Third party APIs

## 8. Project on Front End Development

## Module – 2 (Backend Development - I)

### 9. NodeJs + Express + MongoDB

- a. Building a server using Nodejs
- b. Using the Express js as a middleware
- c. Handling various requests like GET, PUT, POST, PATCH & DELETE
- d. Recording responses using MongoDB
- e. Additional middleware like Mongoose, Morgan, Nodemon & Body Parser
- f. Using MongoDB Atlas and cluster

### 10. Version Control

- a. Introduction to Version Control
- b. Introduction to Git and GitHub
- c. Why Git
- d. Centralized and Distributed Version Control System
- e. Installing Git
- f. Git Basics
- g. Forking
- h. Cloning
- i. Making changes to Local Repositories
- j. Committing
- k. Branching
- l. Collaborating with Multiple Developers

## Module -3 (Java Programming and Problem Solving)

### 1. Java - Basics

- a. Environment Setup - Download and install IntelliJ Idea and Java JDK 8
- b. Control Flow
- c. Conditionals Statements
- d. Arrays
- e. Loops
- f. Methods
- g. Introduction to OOPs - Classes and Objects

### 2. Java - Intermediate

- a. Pillars of OOPs (Inheritance, Polymorphism, Encapsulation and Abstraction)
- b. Collection Framework (ArrayList, LinkedList, Set, List and Maps)
- c. Java Streams
- d. Stream Operations
- e. Lambda Expressions

### 3. Java Exception and Exception Handling

- a. Introduction to Exception Handling
- b. Exception Types and Exception Trees
- c. Exceptions vs. Errors
- d. Handling Exceptions using Try, Catch and Finally
- e. Creating and firing user-defined Exceptions

## 4. Databases (MySQL)

- a. Introduction to Databases
- b. Relational Databases
- c. Creating a Databases
- d. Creating a Table
- e. Insert, Select, Delete and Update
- f. CRUD Operations on Databases
- g. Orders and Groups
- h. Joins
- i. Advance database queries

## 5. File Handling

- a. Introduction to Stream IO
- b. Understanding Streams and it's hierarchy.
- c. Reading and writing Files
- d. Input and Output

## 6. JDBC

- a. Introduction to API's
- b. Introduction to JDBC and its components
- c. JDBC Drivers
- d. Creating a connections
- e. Statements and ResultSet
- f. Best Practices on writing JDBC code

## 7. Problem Solving

- a. Arrays
- b. Bit Manipulation
- c. Searching and Sorting
- d. Recursion and Backtracking
- e. Hashing
- f. Stacks
- g. Queues
- h. Linked List
- i. Trees
- j. Heaps
- k. Greedy Programming
- l. Dynamic Programming
- m. Graphs

## 8. Project using Java Programming

## Module – 4 (Systems Design)

1. Introduction to Distributed Systems
2. Load Balancers
3. Consistent Hashing
4. CAP Theorem
5. Databases – SQL vs NoSQL
6. Caching
7. Messaging Queues - Kafka
8. Case Study 1 – Designing High Level of URL Shortening Service
9. Case Study 2 – Designing High Level of YouTube or Netflix
10. Case Study 3 – Designing High Level of API Rate Limiter
11. OOPs Concepts
12. Design Patterns
13. UML Designs
14. Activity Diagrams
15. Class Diagrams
16. Case Study 4 – Designing URL Shortening Service like TinyURL
17. Case Study 5 – Designing Ticket Booking System like BookMyShow
18. Case Study 6 – Designing Hotel Booking System like OYORooms

## Module – 5 (Backend Development - II)

### 1. Introduction to the Web

- a. Hypertext Transfer Protocol
- b. What are Web Applications
- c. Client – Server Architecture
- d. Introduction to MVC Architecture
- e. Model, View and Controller

### 2. Backend Design Patterns

- a. Introduction and utility of Design Patterns
- b. Design Patterns - MVC, Active-Record, IOC, Singleton, Facade, Adapter, Mediator
- c. Introduction to Spring MVC
- d. Understanding the Folder architecture
- e. Introduction to Servlets

### 3. Servlets, JSPs and ServletContainers

- a. Use of Servlets
- b. Servlet life cycle - Servlet creation and Servlet destruction
- c. Mapping Servlets to JSPs
- d. Introduction to ServletContainers
- e. Utility and use of ServletContainers

### 4. SpringBoot Framework

- a. Introduction to Application Development using Springboot
- b. SpringBoot “Hello World” Application
- c. Understanding Dependency Injection
- d. Inversion of Control

## 5. Object Relational Mapper

- a. JDBC vs ORMs
- b. Mapping between Object and Database Tables
- c. Introduction to Java Persistence API (JPA)
- d. Introduction to Hibernate
- e. Transient and Persistent
- f. Entity and Entity Manager
- g. Repository Class
- h. Migrations

## 6. REST APIs

- a. Understanding REST Architecture
- b. MVC vs REST API
- c. Building RESTfull APIs
- d. Testing Backend APIs with Postman

## 7. User Authentication

- a. Creating User Models
- b. Creating API routes for user sign up and login
- c. Sending token to the frontend
- d. Implementing JWT
- e. Token Expiration
- f. Introduction to Authorization with 'Auth0'

## 8. Creating a Blogging Website in Springboot

- a. Designing the UML for Application
- b. Login and Register Functionality Implementation
- c. Create, Update and Delete a Blog Post
- d. User and Post Mapping
- e. Designing the Frontend
- f. Integrating frontend with the backend APIs

## Module - 5 (AWS Deployment and Capstone Project)

### 1. Springboot App Deployment on AWS

### 2. Capstone Projects

- a. Food Ordering Application
- b. Blogging Web Application
- c. Social Media Web Application
- d. Youtube Clone
- e. Netflix Clone
- f. Spotify Clone
- g. Splitwise Clone
- h. Etc...