

FULL STACK DEVELOPMENT

FULL Stack Developer Training is a 360° training program offered by GRRAS, comprehensive and designed to train participants in creating enterprise-style applications with server-side code using Node.js, Express framework, front-end coding using React JS and databases with Mongo DB.

This training program has become one of the most popular and in-demand because of all the benefits it offers with a combination of technologies. Web developers who want to take their careers to a new level can do so with FULL Stack (MEAN) Developer Training (**Master Program**).

After completion of the training, participants will become experts in front-end and back-end JavaScript technologies of the popular **Mongo DB, Express, React and Node**. This training program is designed to give participants a thorough understanding of FULL Stack technologies, Mongo DB, Express, Angular JS, React JS and Node JS, and more.

Course Content: -

- HTML
- CSS with Bootstrap
- Java Script
- New SASS
- Java Script Advance
- Mongo DB
- React JS
- Node and Express js
- AWS Cloud Integration
- **Interview Preparation**
- **Resume Development**

HTML 5.1(Hypertext Mark-up Language)

- Introduction of tags
- Empty and Container Tags
- Html elements
- Html attributes
- Html tables
- Html Lists
- Html div with CSS float
- Html forms
- Html other important tags

CSS 4.0 (Casecadding Stylesheet) Bootstrap 5.1

- Introduction of CSS
- Inline CSS
- Internal CSS
- External CSS
- Box model and Margin, Padding
- Float property
- Positioning in CSS
- Animation

- Transition
- Background (Image and color)
- Advance Selector
- Units
- Flex
- Image Handling
- Display Css

BootStrap Framework

- Introduction of Bootstrap
- Bootstrap carousel
- BS Grid Basic
- BS Typography
- BS Tables
- BS Images
- BS Jumbotron
- BS Alerts BS Buttons
- BS Button Groups
- BS Badges/Labels
- BS Progress Bars
- BS Pagination
- BS Pager
- BS List Groups
- BS Dropdowns
- BS Collapse
- BS Tabs/Pills
- BS Navbar
- BS Forms
- BS Inputs
- BS Carousel
- BS Modal
- BS Tooltip
- BS Popover
- BS Affix
- BS Filters

JavaScript

- JS Introduction
- JS Variables
- JS Operators
- JS Arithmetic
- JS Assignment
- JS Data Types
- JS Functions
- JS Objects
- JS Events
- JS Strings
- JS Numbers
- JS Arrays
- JS Dates
- JS Date Formats
- JS Math
- JS Random
- JS Booleans



ORRAS

- JS Control statements
- JS Type Conversion
- JS Bitwise
- JS RegExp
- JS Errors
- JS Scope
- JS Strict Mode
- JS this Keyword
- JS Let
- JS Const
- JS Debugging
- JS JSON

Dom Feature

- How to target Element
- How to fetch current Content
- Modify, Update, and Append Content
- Attributes & Class through JS
- String with variables
- Arrow Function
- Inbuilt Functions
- Event
- Event Listener
- Form Validation
- Type and Data congestion
- Regular Expression
- Local, global and return concept

SASS

- Installation & Setup
- About SASS / CSS / BootStarp
- Implement of SASS in project
- Variable Declaration and reuse of Variables
- Benefit of SASS

JavaScript Advance

- Variable Hoisting
- Ways of Function Definition
- Ways of function calling
- Context binding
- JS Clousers
- JS Objects
- JS Constructor
- JS prototypal inheritance
- JS CallBacks
- JS CallBack Hell
- JS Promise intro
- JS Promise as a solution for CallBack Hell
- Observable
- Observable implementation with RxJs
- Async / Await in JS
- JS global object
- understanding 'this'
- Module Export and Import

React JS Content

Introduction to react JS

- Introduction to Basic building blocks of Web Application Development. HTML-CSS-JS
- SPAs and React Web Apps.
- ReactJS Overview.
- Single Page Apps VS Multi-Page Applications.
- Installing ReactJS and writing our first Hello World code.
- The Folder Structure of React Application.
- Basic Building Blocks of React JS – (components, state, props,jsx)
- What are the components? Understanding the component basics and different types of components.
- What are the functional components and container components?
- Creating your first class-based component and functional component.
- Understanding JSX and JSX restrictions.
- Component Nesting.
- What are props and a simple understanding of props?
- Passing Props into components.
- Accessing props inside the components.
- Creating reusable dynamic components

UI and Components Styling In React

- Styling React components. The different ways of styling the react components.
- Inline styles and external styles to the components.
- Using styled-components.
- Creating a navigation-bar in reactJS.
- Introduction to React Router.
- Creating Routes to our Navigation Bar.

React Architecture

- Components configuration with state, props and children.
- Understanding and using state and props, and Differences between props and state
- Handling Events
- Manipulating the state with setState() method
- Difference between stateless and stateful components.
- Adding two-way binding.
- Rendering content conditionally.
- Making API REQUESTS with React.
- Fetching data – Axios vs Fetch.
- Building Lists from data
- The purpose of keys in lists and Implementing keys in lists.
- Handling User Input with forms and events.
- Controlled Elements versus uncontrolled elements.
- Handling forms submitted.
- Understanding this in Javascript and solving context issues.
- Communicating child to parent.
- Handling user Input of dynamic forms
- Handling form submission(MAKING API CALL)
- Adding Custom Form Validation.
- Showing validation errors and other error messages.
- Handling overall form validity.

Redux and advance Concepts

- Using Refs for DOM Access.
- Redux, and The complexity of managing state.
- Understanding the Redux flow.
- Adding Redux to the React Project and Redux Devtools.
- What is the provider and adding the provider?
- Setting Up the Reducer and the Store
- Dispatching the Actions.
- Adding Subscriptions.
- Connecting React to Redux.
- Connecting the store to react.
- Dispatching the Actions from within the Component.
- Passing and Retrieving Data with Actions.
- Switch Case in the reducer
- Updating the State and the Array Immutable.
- Combining multiple Reducers.
- Handling Authentication in React.
- The Context System with React.
- Replacing Redux with Context or using both together.
- Higher-Order Components in React.
- The concept of Render Props.
- Error Boundaries.
- React.Fragments

Hooks

- Introducing Lifecycle methods.
- Why use Lifecycle methods
- React Hooks – An Introduction, Why React Hooks
- Moving from classes to functional components
- Introducing the first Hook: useState Hook and Second Hook: useEffect Hook
- Data Fetching with hooks/replacing the class lifecycle methods
- Fetching data on component mount with useEffect
- Using the useRef Hook
- Displaying loading state with useState
- Error Handling and displaying errors with useState
- The useReducer Hook
- Using the useContext Hook

Getting started with Node Welcome

- What is Node
- Node Architecture
- How Node Works
- Installing Node
- Your First Node Program

Node Module System

- Introduction
- Global Object
- Modules
- Creating a Module
- Loading a Module
- Path Module
- OS Module

- File System Module
- Events Module
- Event Arguments
- Extending EventEmitter
- HTTP Module

Node Package Manager (NPM)

- Introduction
- Package.json
- Installing a Node Package
- Using a Package
- Package Dependencies
- Installing a Specific Version of a Package
- Updating Local Packages
- DevDependencies
- Uninstalling a Package
- Working with Global Package

Express

- Installation
- Express over Node JS
- Template Engine
- Template Handling in Express Js
- Router Concept
- Express Connectivity with MongoDB
- Model Concept
- CRUD Concept
- Login
- Middleware concept
- Image/video upload
- Email
- Session and Session Variables
- Cookies and Feature
- Controller
- Count Data from Multiple table
- Sorting in data
- Concept of `foreach()` method in express

Express- Building REST API's

- Introduction
- RESTful Services
- Introducing Express
- Building Your First Web Server
- Nodemon
- Environment Variables
- Route Parameters
- Handling HTTP GET Requests
- Handling HTTP POST Requests
- Calling Endpoints Using Postman
- Input Validation
- Handling HTTP PUT Requests
- Handling HTTP Delete Requests

- Project- Build the Genres API
- Building RESTful APIs with Express Recap
- Testing API through **Postman** & VSCode Editor
- Try And Catch Concept
- Header status with API

Mongo DB

CRUD Operations Using MongoDB

- Introducing MongoDB
- Installing MongoDB on Windows
- Connecting to MongoDB
- Schemas
- Models
- Saving a Document
- Querying Documents
- Comparison Query Operators
- Logical Query Operators
- Regular Expressions
- Counting
- Updating a Document
- Removing Documents

Mongo – Data Validation

- Validation
- Built-in Validators
- Custom Validators
- Async Validators
- Validation Errors
- SchemaType Options

Mongoose- Modeling Relationships between Connected Data

- Modelling Relationships
- Referencing Documents
- Population
- Embedding Documents
- Using an Array of Sub-documents
- Transactions
- ObjectID
- Validating ObjectIDs
- A Better Implementation

Authentication and Authorization

- Introduction
- Creating the User Model
- Registering Users
- Hashing Passwords
- Authenticating Users
- Testing the Authentication
- JSON Web Tokens
- Generating Authentication Tokens
- Storing Secrets in Environment
- Variables Setting Response Headers
- Getting the Current User
- Logging out Users
- Role-based Authorization

- Testing the Authorization
- Authentication and Authorization Recap

Handling and Logging Errors

- Introduction
- Handling Rejected Promises
- Express Error Middleware
- Try Catch Blocks
- Express Async Errors
- Logging Errors
- Logging to MongoDB
- Uncaught Exceptions
- Unhandled Promise Rejections

Deployment

- Introduction
- Heroku intro
- Account creation on Heroku
- Preparing the App for Production
- Adding the Code to a Git Repository
- Deploying to Server
- Testing the app

Cloud Computing

- Introduction of AWS
- AWS Elastic Beanstalk (Run and manage web apps)
- Amazon RDS (Managed relational database service for MySQL, PostgreSQL, Oracle, SQL Server, and MariaDB)
- Amazon DynamoDB (Managed NoSQL database)
- Amazon Simple Storage Service (S3) (Scalable storage in the cloud)
- Heroku Cloud Platform

Projects on MEAN using all technologies:

- **Ecommerce Web app** - React and Express (Rest Based)
- **Online Examination Application**
- **Online Library Management**
- **Online Contact Management**
- **Multilevel Marketing Management**
- **CRM**