

## MEAN Stack Development

<https://ipacademy.in/>

+91 8870079862

### MongoDB

Introduction to MongoDB

What is NoSQL

Types of NoSQL DBs

Necessity for NoSQL DBs

Relational Databases Vs NoSQL Databases

Advantages of NoSQL Databases

Introducing MongoDB

### Installation and Configuration of MongoDB

Downloading and Installing MongoDB

Command Line Mode - Shell

GUI mode with MongoDB Compass

Starting and Stopping MongoDB

Getting Help

### Fundamentals of MongoDB

Understanding MongoDB model

SQL Terms Vs MongoDB Terms

Understanding Database

Understanding Collections

Collections Vs Tables

Understanding Schemas and Documents

Creating and Dropping a Database

### Working with Data - CRUD Operations

Creating Collections

Inserting Single Document and Multiple Documents

Inserting with `_id` Field

Inserting Document with Arrays

Understanding Embedded and Nested Documents

Updating Single and Multiple Documents

Using Update Operators – Field Update Operators and Array Update Operators

Deleting Single and Multiple Documents

Understanding and Implementing Bulk Write Operations

Working with Retryable Writes

## **Querying Collections**

Using Query Operator - find() to query Documents

Querying Fields and Nested Fields

Using Logical Operations in Query

Querying Nested and Embedded Documents

How to Query an Array?

Projecting Fields in a Query

Using \_id Field in Query

Querying Null and Missing Fields

Querying Text in MongoDB

Indexing Text for Search

Understanding Text Search Operators

What is Geospatial Search

Understanding and using Geospatial data

Geospatial Indexes and Queries

Aggregation Operations

Understanding Aggregation Operations

Using Aggregation Pipelines

Pipeline Expressions and Behaviors

Understanding Map-Reduce Functions in MongoDB

Understanding Map and Reduce

Using Map-Reduce

Using Map-Reduce Javascript Functions

Map-Reduce results

Understanding Incremental Map-Reduce

Understanding and using Map-Reduce Concurrency

Exploring Aggregation References

Understanding Data Models and Schema

Introduction to Data Modelling

Understanding Schema in MongoDB

Flexible Schema and References

Understanding Schema Validation

Create Validation Rules

Using Query Expression to Validate Data

Bypassing the Document Validation

Deep dive into Data Modeling Concepts

Handling Transactions in MongoDB  
Understanding Transactions and Transactions API  
Understanding Atomicity  
Executing Multi-document Transactions or Distributed Transactions  
Transactions Operations and Restricted Operations  
Production Considerations  
Handling Indexes in MongoDB  
What is Indexing and Advantages of Indexes  
Single Field and Compound Indexes  
Multi key Indexes, Text Indexes, Wild-card Indexes  
Hashed Indexes  
Managing Indexes

Managing Security  
Implementing Security in MongoDB  
Understanding and Enabling Access Control  
Handling Authentications  
Understanding and Configuring Encryptions  
Configuring and Managing Auditing

Replication  
Importance of Replication for High Data Availability  
Implementing Replication in MongoDB with Replica Sets  
Understanding Asynchronous Replication  
Change Stream

Sharding in MongoDB  
Understanding the purpose of uses of Sharding  
Understanding Sharded Cluster Components  
Harshed Sharding and Ranged Sharding  
Handling Collation in Sharding  
Understanding Zones  
Introduction to Sharded Cluster Administration

## **Node.js**

Understanding the Web

Architecture of a Web Application  
Server Side Scripting  
Client Side Scripting  
Importance of Performance

### **Introduction to Node.js**

What is Node.js  
Advantages of Node.js  
Installing Node.js and Configuring Node.js  
Architecture of Node.js  
Appreciating the Performance of Node.js  
Understanding Event Driven code execution and Asynchronous Events in Node.js  
Understanding Event Emitters

### **Implementing Node.js**

The Node lifecycle  
Creating and running a Node Server  
Handling Requests and Responses  
Routing and Redirecting Requests  
Handling Streams, buffers and Events  
Handling Multi-processing in Node.js  
Handling Run time Errors

### **Modules in Node.js**

Understanding Modules in Node.js  
Exploring various built-in Modules  
Third party Modules for Node.js  
Introduction to Node Package Manager - NPM  
Understanding Global and Local NPM Packages  
Installing and Updating a Module  
Deleting a Module  
Introduction to various NPM Modules

### **ExpressJS Framework**

Introduction to ExpressJS Framework  
Why the need for Express Framework  
Overview of ExpressJS framework  
Installing ExpressJS

Understanding Request and Response Objects  
Exploring ExpressJS  
Configuring ExpressJS  
Understanding Middleware

Understand How Middleware Works in ExpressJS  
Add a Middleware  
Creating Custom Middleware  
Handling Requests and Responses  
Handling Routing  
Using ExpressJS Router  
Path Filtering  
Creating and Serving HTML Pages  
Using Jade Templates  
Navigating Datasets  
Using REST in ExpressJS  
Handling POST  
Handling JSON data  
Handling Errors

## **TypeScript Framework**

TypeScript Framework  
Introduction to TypeScript  
Advantages of TypeScript  
Overview of Node and Node Environment Setup  
Installing and Configuring TypeScript Engine  
Understanding Transpilation  
TypeScript Variables and Data types  
Declaring and Defining Variables  
About “any”  
“var” Vs “let” keywords  
Static and Dynamic Type  
Data types : String, Number, Array, Object, Tuple, Enum, Void and NULL

OOPS in TypeScript  
Defining Class and Creating Objects  
Property, Methods and Constructors

Handling Inheritance  
Types of Inheritance in TypeScript  
Understanding Access Modifiers  
Static Methods in TypeScript  
Understanding Interfaces in TypeScript

## **Programming with Angular11 / 12**

Introducing Angular  
What is Angular  
Advantages of using Angular  
Understanding Differences between a Framework and a Library  
Understanding Single Page Applications (SPA)  
SPA Vs. Traditional Applications  
Understanding the MVC Architecture  
MVC in Client and Server Side  
Elements of Angular  
Modules  
Data Binding Directives  
Templates  
Metadata  
Services  
Dependency Injection  
Angular Components  
Angular 7 / 8 – Environment Setup  
Installing Node.js and understanding Node Package Manager ( NPM )  
What is Angular CLI  
Advantages of using Angular CLI to manage Projects  
Installing Angular CLI  
Create a Angular App with CLI  
Angular Application File Structure  
Executing Angular Application  
Code Linting  
Installing and Understanding Webpack

Angular Modules  
Understanding Angular Modules  
When to use Angular Modules

Understanding NgModule  
Understanding Root Modules, Feature Modules, Shared Modules  
Using Providers, Managing Imports  
Bootstrapping Angular Application  
Angular Components  
Understanding Components  
Understanding Components Structure  
Components Interaction and Component Lifecycle

Using ngOnInit  
Exploring All Lifecycle Hooks  
Understanding Change Detection  
Working with Zone.js  
Working with Component Decorator and its Meta data  
Passing Data between Child and Parent  
Angular – Data and Event Bindings  
Understanding Data Binding  
Explaining Two-way Data Binding  
Working with Style Binding and Class Binding  
Understanding and Exploring Event Binding  
Element Reference and Event Filtering  
Angular Directives and Custom Directives  
What is Directives in Angular?  
Understanding Different types of Angular Directives  
Structural Directives Vs. Attribute Directives  
Exploring all the Angular Structural and Attribute Directives  
Creating Custom Attribute Directives  
Creating Custom Structural Directives  
Writing Attribute Directive Code  
Respond to User Initiated Events  
Pass and Receive property values

### **Angular Pipes**

Understanding Pipes in Angular  
Uses of Pipes  
Parameterizing Pipes  
Pipes Vs. Directives

Exploring Pipes : Uppercase, Lowercase, Number, Percent, Currency, Date, Slice, Json  
Difference between Pure Pipes and Impure Pipes  
The PipeTransform Interface  
Chaining Pipes  
Creating Custom Pipes

Angular Services and Dependency Injection  
Understanding Services  
Purpose of using Services  
Creating Services in Angular  
Overview of Singleton Object  
Understanding Dependency Injection  
Injectors and Providers  
@Injectable() and Hierarchical DI

### **Angular Routing and Advanced Routing**

Understanding the Essentiality of Routing  
Configuring and Defining Routing in Angular Application  
Understanding the Routing Parameters  
Understanding Static and Dynamic Routes  
Passing and Fetching Route Parameters  
Creating Child Route

Angular Template Driven Forms  
Introduction to Forms  
Overview of Template Driven Forms  
Setting up Forms in Angular and Forms Control  
Understanding ngForm  
Accessing Form Values and setting default form values  
Understanding Various Form States  
Form Validations  
Form Groups and Form Control Class  
Understanding One Way Data Binding in Angular Forms

Angular Reactive Forms  
Understanding Reactive Forms  
Template Driven Forms Vs. Reactive Forms  
Reactive Form Control and Sync View  
Dynamic Validations in Reactive Forms



## Getting Reactive Form Values

### Angular Asynchronous Operations and Advanced HTTP

- Understanding Event Emitter
- Transferring Data with Event Emitter
- Creating Custom Events and Trigger
- ng-content and ng-container
- Introduction to Async
- Understanding HTTP mechanism
- Handling HTTP Request and Response
- Introduction to Observable and Observer
- Call back Methods in Observable
- Creating Custom Observable
- Connecting backend and APIs

### Angular Authorizations

- Understanding Client side Authorization and Server Side Authorization
- Creating Guard in Angular
- Understanding Route Guard
- Setting up Route Guard
- Understanding HTTP Interceptors
- Angular Animations
- Introduction to Animations
- Animations Setup
- Understanding States & Transitions
- Entering and Leaving from States
- Animatable Properties
- Animatable Units
- Automatic Property Calculations
- Understanding Animation Timing
- Multistep Animation using Keyframes

### Testing Angular Applications

- Introduction to Testing
- Unit Testing
- Setup Jasmine Framework
- Basics to Test Components
- Components Test Scenarios
- Introduction to Karma
- E2E Testing and Protractor

Angular Security and Internalization

Importance of Security

Security in Angular

Sanitizing the Dangerous Value

Trusting Values

bypassSecurityTrustHtml

bypassSecurityTrustScript

bypassSecurityTrustStyle

bypassSecurityTrustUrl

bypassSecurityTrustResourceUrl

Cross-site Request Forgery

Pre-Compiled and runtime

Using ng2-Translate

Angular Performance, Optimization and Deployment

Change Detection Strategy

Understanding Web Workers

Precompiling (AoT)

Lazy Loading

Deployment Best Practices

Production Mode

**Real Time Project using MEAN Stack**

ipacademy.in