CROMA CAMPUS 15 YEARS | in Collaboration with Microsoft

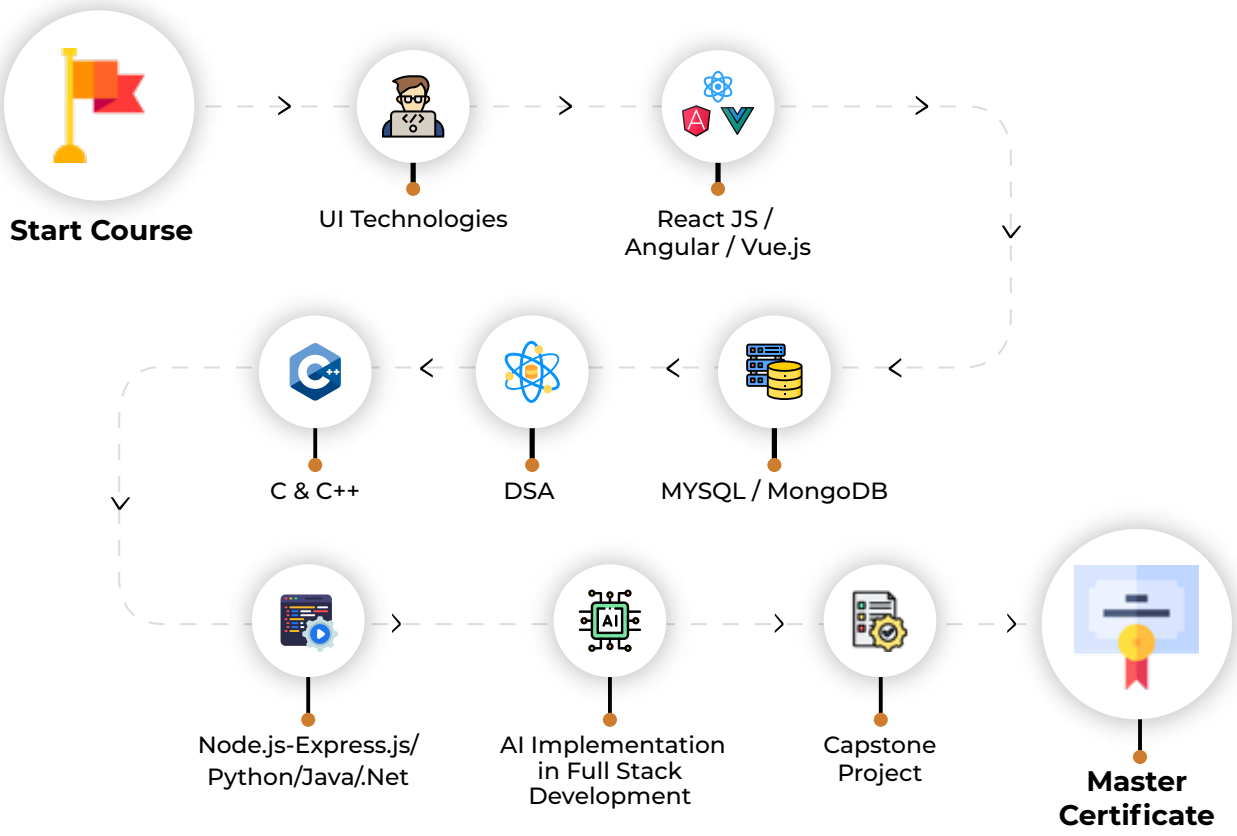
Course Curriculum

MASTERS IN FULL-STACK DEVELOPMENT TRAINING CURRICULUM

500K+	100K+	300+	300K+	200+	150+
Satisfied Students	Online Students	Trainers	Placements	Global Certifications	Companies

Program INDEX

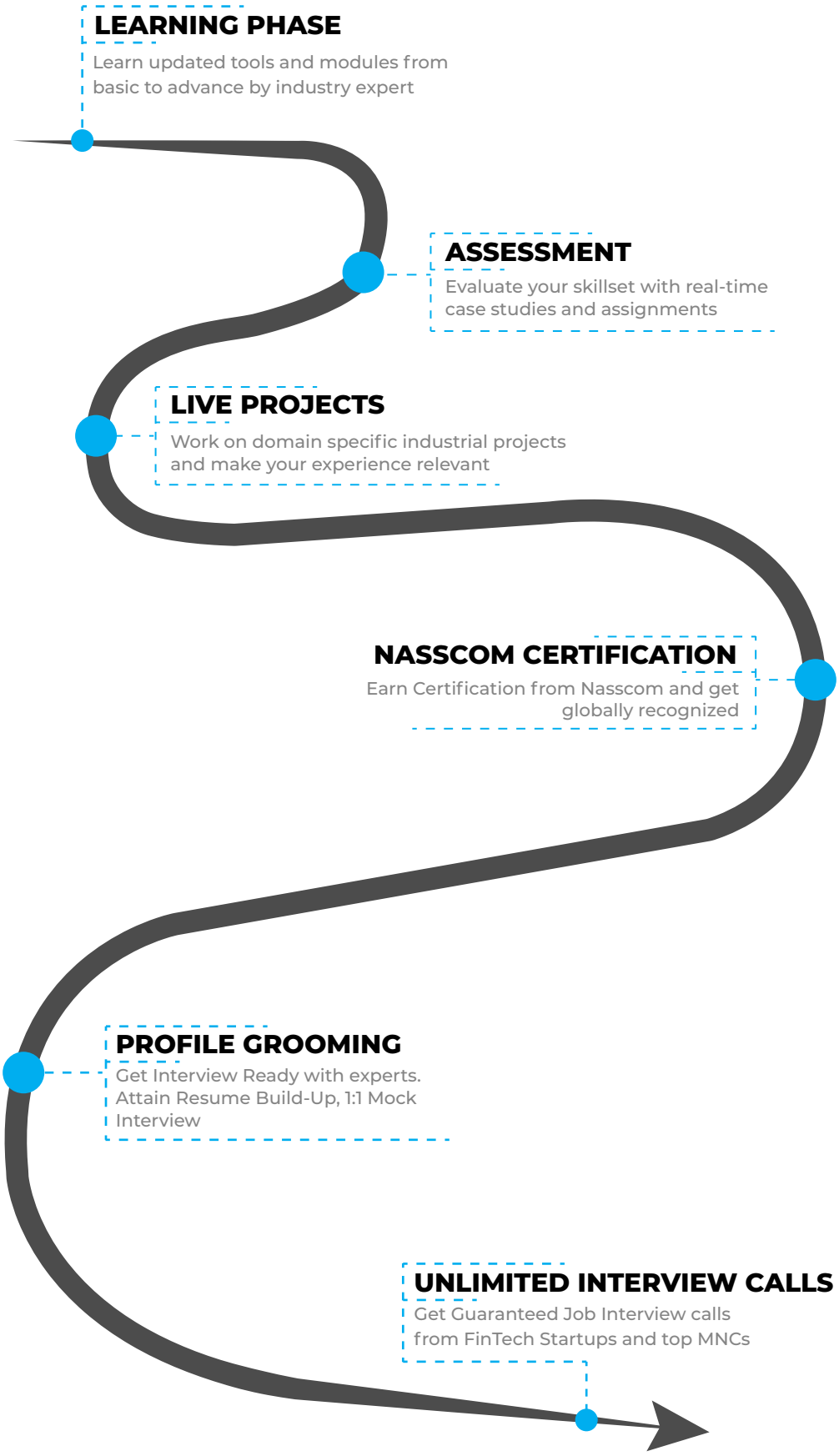
Learning Curve



Modules of **Master in Full Stack Development**

- **UI Technologies**
- **React JS / Angular / Vue.js**
- **C & C++**
- **DSA**
- **MYSQL / MongoDB**
- **Node.js-Express.js / Python / Java / .Net**
- **AI Implementation in Full Stack Development**

Transition Process



Module 1: UI Technologies

Introduction

- Software architecture and its flow to design a layout
- Technologies needed For Full Stack
- Current market requirements on UI/UX
- Difference between UI/UX design
- UI/UX Developer roles and Responsibilities
- Figma design and prototypes

HTML5

- Basics
  - HTML5 & difference between previous versions
  - Structure of HTML Page & DOM
  - Tags, Attributes, HTML Elements, Style attribute
  - Heading tags (H1...H6), Tags, Paragraph tag
  - Typography, Inserting images & formats, Margin, Padding, Borders
  - Hyperlinks, buttons, Colors
  - Create layout using div and section tags
  - Difference between Inline and block-level elements
  - Table, Lists, Forms & Controls, Inserting Maps
- HTML5
  - Header, Footer
  - iFrames, Audio, Video tags

CSS 3

- Introduction
  - CSS3 and difference with previous versions
  - CSS Types, Syntex, Property Values
  - Selectors, Box Model, DOM
  - Background, Display, Position, height & width and other properties
- More Properties
  - CSS Overflow, blur, opacity
  - CSS Float and Clear
  - Pseudo Class and Element
  - Animations with keyframes
  - box shadow
  - Transform, translate, transition Property
  - Flexbox property in depth and web fonts
  - Web page layout and Media Queries

JavaScript

- Introduction, History and Syntax
- How to use JS in different ways
- Popup Boxes, alerts, prompt, confirm
- Variable type

- Data Types
- Operators
- Conditional Statements & Loops
- Functions
- Arrays & Strings
- Events
- Object and its property
- This keyword
- RegExp
- ErrorHandling & async-await conepts

Bootstrap 5

- Introduction
- Bootstrap5 and comparison with previous versions
- How to download or start with CDN
- Grid System and Flex Concept in bootstrap 5
- Typography, Colors, Buttons
- Images, tables, Alerts, Badges, spinner
- Cards, Pagination, list groups
- Dropdown and collapse concept
- Responsive Forms (Contact Form)
- Navbar and footer
- Carousel and Modal
- Tooltip, popover and Toast
- Scroll spy and offcanvas
- Utilities

JQuery

- Introduction and uses
- Selectors and Events in jQuery
- jQuery Effects
  - Hide/Show, Fade, Slide, Animate, stop(), call back
- jQuery HTML
  - get, set, add/remove, CSS classes, css()
- jQuery Traversing
  - Finding elements using jQuery techniques & Filtering elements
- jQuery AJAX
  - Advantages with Ajax and its limitations
  - Samples working with Ajax
  - Different data formats used in Ajax (string, xml, Json, etc)
  - XML and JSON difference

Module 2: React JS / Angular / Vue.js (Any of 1)

React JS

- Concept of Modern JavaScript (ES6)

    - Introduction to ECMAScript 2015
    - Modern Java Script and React js
    - Var, Let and Const in JavaScript
    - Template literals (Template strings) in ES6
    - New Strings Methods in ES6
    - Array Destructuring & Object Destructuring in ES6
    - Fat Arrow Function
    - Default & REST Parameters in ES6
    - Spread Operator in ES6
    - New Array Methods in JavaScript
    - New Numbers & Global Methods
    - OOPS (Classes, Object & Inheritance) in ES6
    - Async Await in JavaScript
    - Error Handling in JavaScript
    - Fetch API in Promises in JavaScript
- React Modules
    - Introduction, prerequisites and uses
    - Package Manager (yarn, npm, pnpm, bun with vite)
    - Environment Setup, download and install with VS Code
    - Basic Syntex and Folder Structure
    - JSX and Java script Expression in JSX
    - React Components (Functional Components)
    - Render Multiple Elements (React fragment)
    - Module import and Export
    - Inserting images in react
    - CSS styling in React
    - JS Map Method & key concept in react
    - Conditional rendering in react
    - Arrays in React
    - Props Concepts in react
    - Event Propagation in react
    - Bootstrap in React
    - Hooks Concept
    - State Management (useState)
    - useEffect
    - useContext
    - useCallback
    - useParam
    - useNevigate
    - useRef
    - useReducer
    - Router and create a navbar in react
    - Use Bootstrap5 in React
    - Fetch API Data in React
    - Axios With Get & Post
    - Form in React with Login Form Submition
    - Component Lifecycle

- Mounting and Unmouting in react
- Animations in React

Angular

- Angular Overview
  - History of Angular
  - The leap from AngularJS to Angular
  - What's new in Angular 16
  - Angular 11 vs Angular 16
  - Desktop Application class User Experience
  - Productivity and Tooling
  - Performance
  - Community
  - Full-featured Framework
  - Supported Browsers (Angular 16)
  - Platform for Targeting Native Mobile not just Web Browsers
- All about TypeScript 4.2
  - Introduction
  - What is Typescript?
  - Why Typescript?
  - Setup and installation
  - IDE support
  - Different typescript versions
  - Typescripts 4.2 for Angular 16
  - Scoping using let and const Keywords (ES6)
  - Template Literals (ES6)
  - Rest and Spread Parameters (ES6)
  - De-structuring (ES6)
  - Introduction to Types
  - Type inference
  - Type Annotations
  - Number
  - Boolean
  - String
  - Array
  - Tuple
  - ENUM
  - Any
  - Void
  - Null and Undefined
  - Never
  - Introduction to Functions
  - Using types in functions
  - Function as types
  - Optional and default parameters
  - Arrow functions
  - Introduction to Classes

    - Inheritance
    - Access modifiers
    - Getters and setters
    - Read-only & static
    - Introduction to Interfaces
    - Optional properties and methods
    - Strict structural contract
    - Extending interface
    - Implementing interface
    - Introduction to Modules
    - Import / Export
    - Default & Decorators
- Angular CLI
    - Angular CLI
    - Anatomy of the project
    - Setting up a workspace
    - Updating Angular apps using ng update
    - Adding support for external libraries using ng add
    - Debugging Angular apps
    - Working with Augury
    - Using the Angular Language Service with Microsoft VS Code
- Angular building blocks
    - Modules
    - Components
    - Templates
    - Metadata
    - Data binding
    - Directives
    - Services
    - Dependency injection
    - Angular Ivy
- Angular Modules
    - Why modules?
    - How to create modules?
    - Built-in modules
    - Root Module
    - feature module
- Components in Angular
    - Introduction
    - @Component decorator
    - Component configuration object
    - Custom components
    - Component with templates
    - Inline
    - External
    - Component with Styles
    - Inline

    - External
    - Angular Elements
- Templates in Angular
    - HTML basic syntax
    - Template expressions
    - Template syntax
    - Attribute, class, and style bindings
    - @Input ()
    - @Output
    - Template reference variables
    - Safe navigation operator
- Data binding
    - Interpolation
    - Event binding
    - Property binding
    - two-way binding
    - uses and examples
- Angular Directive and Pipes
    - Introduction to Directives
    - Built in Structural Directives
    - Built in Attribute Directives
    - Building Custom Attribute Directives
    - Building Custom Structural Directives
    - Introduction to Pipes in Angular
    - Using Built-in Pipes
    - Creating Custom Pipes in Angular
    - Pure and Impure Pipes in Angular
- Angular Forms
    - Types of Form in Angular
    - Reactive form in Angular
    - Dynamically Adding or Removing Form Control (s) or Form Group (s) using Form
    - Array(s)
    - Adding Synchronous Custom Validations to your Reactive Form
    - Adding Asynchronous Custom Validations to your Reactive Form
    - Resetting the value of a form
- Dependency Injections
    - Introduction
    - Why DI?
    - @Injectable decorator
    - Custom service development
    - Registering the service with Ng Module using providers key
    - Provider Types
    - Class
    - Factory
    - Value
- Angular Routing with Guards
    - Implementing Routing in an Angular App

        - Abstracting the user link to a separate component
        - Implementing Child Routes
        - Path Match and Route Types
        - Introduction to Route Guards in Angular
        - CanActivate and CanActivateChild Guards in Angular
        - CanDeactivate Guard in Angular
        - Prefetching Data for a Component using Resolve
    - RxJs Primer
        - Introduction
        - Why RxJs?
        - Observable interface
        - Streams
        - Operators
        - Subscription
        - Subject
        - Schedulers
    - Back-end services, HTTP Client
        - Setup installing the module
        - Making a request for JSON data
        - Type checking the response
        - Error handling
        - Sending data to the server
        - Making a POST request
        - Configuring other parts of the request
    - Angular Advanced Features
        - Creating Libraries
        - Angular Material Essentials
        - Lazy Loading
        - Customizing the Angular CLI using the Builder API
        - Server Side Rendering with Angular Universal
        - Working with Service Workers
        - Building a Progressive Web App
    - Testing
        - Creating Libraries
        - Angular Material Essentials
        - Lazy Loading
        - Customizing the Angular CLI using the Builder API
        - Server Side Rendering with Angular Universal
        - Working with Service Workers
        - Building a Progressive Web App
    - Learn Deployment
        - Manually
        - Using the Angular CLI with Ahead-Of-Time (AOT) Compilation and Tree-Shaking
        - (removing unused library code)
        - Deployment Platforms for Angular Apps

Vue.Js

- Introduction to Vue.js
  - What is Vue.js
  - Why Vue.js
  - Quick tour of Vue-cli
  - Hello World with Vue.js
  - Anatomy of Hello world vue app
- Working with Template & viewmodel
  - What is vue.js template
  - What is viewmodel / vue instance?
  - Template syntax
  - Interpolation
  - Directive
  - Filters
  - Shorthands
  - Javascript expressions
  - Computed properties
  - Watchers
  - Class and style bindings
  - HTML classes
  - Inline styles
  - Conditional Rendering
  - Iterative Rendering
- Vue Instance Revisited
  - Basic vue instance usage
  - Multiple vue instances
  - Vue instance Lifecycle
  - Accessing data from outside vue instance
  - Data and methods
  - More about $el, $data and $refs
- Event handling
  - Introduction
  - Listening to events
  - Method Event handlers
  - Event modifiers
  - Key modifiers
- Working with forms & two way binding
  - Introduction to two way binding via v-model
  - Handling user input with HTML form controls
  - Data binding
  - .lazy, .number, .trim modifiers
  - Submitting a form
- Introduction to components
  - What are components?
  - Why components
  - Hello World component
  - Anatomy of a component
  - Registering Components Locally and Globally

    - Root Component
    - Thinking in components
    - Composing application as components
- Passing data to components
    - Props, parent to child communication
    - Custom events, child to parent
    - Non parent child communication event emitter style using .emit, .on
- Content distribution with slots
    - Compilation scope
    - Single slot
    - Named slot
    - Scoped slot
- Dynamic components
    - Switching Multiple Components with Dynamic Components
    - Dynamic Component Behavior
    - Keeping Dynamic Components Alive
    - Lifecycle hooks
- Custom directives
    - Introduction
    - Directive hook functions
    - Developing custom directive
    - Passing values and arguments to custom directives
    - Directive hook arguments
- Mixins
    - Introduction
    - Creating and using mixins
    - Option merging
    - Global mixin
- Filters
    - Introduction
    - Local filters
    - Global filters
    - Chaining filters
- Remote communication with vuejs
    - Using vue-resource to working with HTTP
    - Fetch data
    - Post data to server
    - Request Response Interception
    - Template URLs
- Routing
    - Introduction
    - Setting up vue-router
    - Loading routes
    - Routing modes
    - Navigation with router links
    - Styling active links
    - Imperative navigation

        - Route Parameters
        - Setting up child or nested routes
        - Named routes
        - Query parameters
        - Named router views
        - Wildcard routes
        - Route guards
    - State Management
        - Introduction
        - What is need for state management
        - What is vuex
        - Centralized store
        - Understanding Getters
        - Using Getters
        - Mapping Getters to Properties
    - Understanding Mutations
        - Using Mutations
        - Using Actions
        - Mapping Actions to Methods

Module 3 : C & C++

Introduction To Programming With C & C++

- History and evolution of C and C++
- Differences and similarities between C and C++
- Writing and compiling your first program
- Structure of a C/C++ program
- Basic syntax and best practices

Variables, Data Types, And Operators

- Keywords and identifiers
- Basic data types: int, float, char, double
- Modifiers: signed, unsigned, long, short
- Constants and const keyword
- Operators: arithmetic, relational, logical, assignment, bitwise, etc.
- Type casting and operator precedence

Control Structures

- Decision-making: if, if-else, nested if, switch-case
- Loops: for, while, do-while
- Jump statements: break, continue
- Differences in syntax and usage in C and C++

Arrays And Strings

- 1D and 2D arrays
- Array operations (insertion, deletion, search, sort)
- Character arrays and string manipulation in C (gets, puts, strcpy, etc.)

- String handling in C++ (string class and functions)
- Multidimensional arrays

Functions And Recursion

- Function declaration, definition, and calling
- Parameter passing: call by value and call by reference
- Recursion: factorial, Fibonacci, etc.
- Scope and lifetime of variables
- Inline functions in C++

Pointers And Memory Management

- Introduction to pointers
- Pointer arithmetic
- Pointers and arrays
- Pointers to functions
- Dynamic memory allocation

Structures, Unions, And File Handling

- Structures: definition, declaration, and usage
- Nested structures and array of structures
- Introduction to unions
- File handling in C (fopen, fread, fwrite, fprintf)

Object-Oriented Programming (OOP) Basics In C++

- Introduction to classes and objects
- Access specifiers: public, private, protected
- Constructor and destructor
- Function overloading and operator overloading
- Basic concepts of inheritance and polymorphism
- Difference between procedural and object-oriented approach

Module 4: DSA

Arrays

- Traversal, insertion, deletion
- Searching & sorting (binary search, merge sort, quick sort)
- Two-pointer technique
- Sliding window

Strings

- String manipulation (palindrome, anagram, substring)
- Pattern matching
- StringBuilder vs StringBuffer
- Regex basics

Linked Lists

- Singly & doubly linked list

- Reversal, cycle detection (Floyd's algorithm)
- Merge two sorted lists

Stacks & Queues

- Stack using array/linked list
- Infix, postfix, prefix expressions
- Queue, Circular Queue, Priority Queue
- Deque

Recursion

- Basic recursion patterns
- Backtracking

Hashing

- HashMap, HashSet
- Frequency count, duplicates

Trees

- Binary Tree, Binary Search Tree (BST)
- Traversals (in-order, pre-order, post-order)

Module 5: MYSQL / MongoDB (Any of 1)

MYSQL

- SQL Server Fundamentals
  - SQL Server 2019 Installation
  - Service Accounts & Use, Authentication Modes & Usage, Instance Configurations
  - SQL Server Features & Purpose
  - Using Management Studio (SSMS)
  - Configuration Tools & SQLCMD
  - Conventions & Collation
- SQL Server 2019 Database Design
  - SQL Database Architecture
  - Database Creation Using GUI
  - Database Creation Using T-SQL Scripts
  - DB Design Using Files And File Groups
  - File Locations And Size Parameters
  - Log Files And Placement
  - Database Structure Modifications
- SQL Tables in MS SQL Server
  - SQL Server Database Tables
  - Table Creation Using T-SQL Scripts
  - Naming Conventions For Columns
  - Single Row And Multi-Row Inserts
  - Table Aliases
  - Column Aliases & Usage
  - Table Creation Using Schemas

    - Basic INSERT
    - UPDATE
    - DELETE
    - SELECT Queries And Schemas
    - Use Of WHERE, IN And BETWEEN
    - Variants Of SELECT Statement
    - ORDER BY
    - GROUPING
    - HAVING
    - ROWCOUNT And CUBE Functions
- Data Validation and Constraints
    - Table Creation Using Constraints
    - NULL And IDENTITY Properties
    - UNIQUE KEY Constraint And NOT NULL
    - PRIMARY KEY Constraint & Usage
    - CHECK And DEFAULT Constraints
    - Naming Composite Primary Keys
    - Disabling Constraints & Other Options
- Views and Row Data Security
    - Benefits Of Views In SQL Database
    - Views On Tables And Views
    - SCHEMA BINDING And ENCRYPTION
    - Issues With Views And ALTER TABLE
    - Common System Views And Metadata
    - Common Dynamic Management Views
    - Working With JOINS Inside Views
- Indexes and Query tuning
    - Need For Indexes & Usage?
    - Indexing Table & View Columns
    - Index SCAN And SEEK
    - INCLUDED Indexes & Usage
    - Materializing Views (storage Level)
    - Composite Indexed Columns & Keys
    - Indexes And Table Constraints
    - Primary Keys & Non-Clustered Indexes
- Stored Procedures and Benefits
    - Why To Use Stored Procedures?
    - Types Of Stored Procedures
    - Use Of Variables And Parameters
    - SCHEMABINDING And ENCRYPTION
    - INPUT And OUTPUT Parameters
    - System Level Stored Procedures
    - Dynamic SQL And Parameterization
- User-defined functions and Usage
    - Scalar Valued Functions
    - Types Of Table Valued Functions
    - SCHEMABINDING And ENCRYPTION

    - System Functions And Usage
    - Date Functions
    - Time Functions
    - String And Operational Functions
    - ROW_COUNT
    - GROUPING Functions
- Triggers, cursors, memory limitations
    - Why To Use Triggers?
    - DML Triggers And Performance Impact
    - INSERTED And DELETED Memory Tables
    - Data Audit Operations & Sampling
    - Database Triggers And Server Triggers
    - Bulk Operations With Triggers
- Cursors and Memory Limitations
    - Cursor Declaration And Life Cycle
    - STATIC
    - DYNAMIC
    - SCROLL Cursors
    - FORWARD_ONLY And LOCAL Cursors
    - KEYSET Cursors With Complex SPs
- Transactions Management
    - ACID Properties And Scope
    - EXPLICIT Transaction Types
    - IMPLICIT Transactions And Options
    - AUTOCOMMIT Transaction And Usage
    - SAVEPOINT And Query Blocking
    - Complex SPs With Transactions
    - Linked Server And DTC Operations

MongoDB

- Introduction to MongoDB - Architecture and Installation
    - Understanding the basic concepts of a Database
    - Database categories: What is NoSQL? Why NoSQL? Benefit over RDBMS
    - Types of NoSQL Database, and NoSQL vs. SQL Comparison, ACID & Base Property
    - CAP Theorem, implementing NoSQL and what is MongoDB?
    - Overview of MongoDB, Design Goals for MongoDB Server and Database, MongoDB tools
    - Understanding the following: Collection, Documents and Key/ Values, etc.,
    - Introduction to JSON and BSON documents
    - Environment setup (live Hands-on) and using various MongoDB tools available in the MongoDB Package
    - Case study discussion
- Schema Design and Data Modelling
    - Data Modelling Concepts
    - Why Data Modelling? Data Modelling Approach
    - Analogy between RDBMS & MongoDB Data Model, MongoDB Data Model (Embedding & Linking)

- Challenges for Data Modelling in MongoDB
- Data Model Examples and Patterns
- Model Relationships between Documents
- Model Tree Structures
- Model Specific Application Contexts
- Use Case discussion of Data modeling

- CRUD Operations
  - MongoDB Development Architecture
  - MongoDB Production Architecture
  - MongoDB CRUD Introduction, MongoDB CRUD Concepts
  - MongoDB CRUD Concerns (Read & Write Operations) –
  - Concern Levels, Journaling, etc.,
  - Cursor Query Optimizations, Query Behavior in MongoDB
  - Distributed Read & Write Queries
  - MongoDB Datatypes
  - MongoDB CRUD Syntax & Queries (Live Hands on)
- MongoDB on the Cloud
  - Overview of MongoDB Cloud products
  - Using Cloud Manager to monitor MongoDB deployments
  - Introduction to MongoDB Stitch
  - MongoDB Cloud Atlas
  - MongoDB Cloud Manager
  - Working with MongoDB Ops Manager
- MongoDB Administration
  - Administration concepts in MongoDB
  - Monitoring issues related to Database
  - Monitoring at Server, Database, Collection level, and various Monitoring tools related to MongoDB
  - Database Profiling, Locks, Memory Usage, No of connections, page fault etc.,
  - Backup and Recovery Methods for MongoDB
  - Export and Import of Data to and from MongoDB
  - Run time configuration of MongoDB
  - Production notes/ best practices
  - Data Managements in MongoDB (Capped Collections/ Expired data from TTL), Hands on Administrative Tasks.
- Scalability and Availability
  - Introduction to Replication (High Availability)
  - Concepts around Replication
  - What is Replica Set and Master Slave Replication?
  - Type of Replication in MongoDB
  - How to setup a replicated cluster & managing replica sets etc.,
  - Introduction to Sharding (Horizontal Scaling),
  - Concepts around Sharding, what is shards, Key,
  - Config Server, Query Router etc.
  - How to setup a Sharding,
  - Type of Sharding (Hash Based, Range Based etc.), and Managing Shards.
- MongoDB Security

        - Security Introduction
        - Security Concepts
        - Integration of MongoDB with Jaspersoft
        - Integration of MongoDB with Pentaho
        - Integration of MongoDB with Hadoop/Hive
        - Integration of MongoDB with Java
        - Integration of MongoDB with GUI Tool Robomongo
        - Case Study MongoDB and Java
    - Application Engineering and MongoDB Tools
        - MongoDB Package Components
        - Configuration File Options
        - MongoDB Limits and Thresholds
        - Connection String URI Format/ Integration of any compatible tool with MongoDB API and Drivers for
        - MongoDB
        - MMS (MongoDB Monitoring Service)
        - HTTP and Rest Interface
        - Integration of MongoDB with Hadoop and Data Migration MongoDB with Hadoop (MongoDB to Hive)
        - Integration with R
    - Diagnostics and Fixes
        - Overview of tools
        - MongoDB Diagnostic Tools
        - Diagnostics Commands
        - MongoDB Deployment
        - Setup & Configuration, Scalability, Management & Security
        - Slow Queries
        - Connectivity
    - Indexing and Aggregation Framework
        - Index Introduction, Index Concepts, Index Types, Index Properties
        - Index Creation and Indexing Reference
        - Introduction to Aggregation
        - Approach to Aggregation
        - Types of Aggregation (Pipeline, MapReduce & Single Purpose)
        - Performance Tuning.

Module 6 : Node.js-Express.js/Python/Java/.Net (Any of 2)

Node.Js

- Introduction to Node.js
    - What is Node.js?
    - Features & Advantages
    - Installing Node.js & npm
    - Running basic Node.js code
- Node.js Architecture
    - Create a Server using node.js

- https server request
- Event loop explanation
- Call async flow

- Node.js Core Modules
  - fs, path, os, http, events
  - Global objects: __dirname, process, require
  - Understanding REPL
- Working with npm
  - What is npm?
  - package.json and package-lock.json
  - Installing local & global packages
- Modules in Node.js
  - Creating custom modules
  - Event-driven model
  - Modular code structure
- File System Operations
  - Reading & writing files
  - Text file as a data store
  - Synchronous vs asynchronous file handling
- Asynchronous Programming
  - Callback functions
  - Promises
  - Async/Await
  - Error handling in async code

Express.Js

- Introduction to Express.js
  - What is Express?
  - Installing and setting up Express
  - First basic Express server
- Basic Routing
  - Route methods: GET, POST
  - Route parameters
  - Modularizing routes
- Request & Response in Express
  - req object – params, query, body parser
  - res object – send, json, status
  - Handling form and JSON data
- Middleware in Express
  - What is middleware?
  - express.json(), express.urlencoded()
  - Custom middleware
- REST API with Express
  - Understanding RESTful architecture
  - Route structure
  - Build local connection with MongoDB
- Using Postman for Testing

        - Installing Postman
        - Testing GET, POST, PUT, DELETE
        - Saving Postman collections
    - Error Handling in Express
        - Try-catch & next()
        - 404 and global error handler
        - Custom error responses
    - Environment Configuration
        - Using .env file
        - Storing secrets securely
        - dotenv package
    - Mini REST API Project (No DB)
        - Create a User Management API
        - Store data in JSON file
        - Add proper routes, validation, and error handling
    - Introduction to MVC Pattern
        - What is MVC?
        - Creating Models, Controllers, Routes
        - Refactor existing project using MVC
    - REST Full API Project (Connection with mongoDB)
        - Create a User Management REST Full API using MVC Pattern
        - Store data in MongoDB database
        - Add proper routes, validation, and error handling

Java

Core Java

- Introduction:
    - Java History and Versions
    - Differences between java and others
    - Java Features
    - Java Naming Conventions
- Java Architecture
    - Introduction to Java architecture: JDK, JRE, and JVM.
    - Java Installation and setup with VS code
    - Text-Editors
    - Run and Compile a Java Files
    - Explain the Java file Structure
- Language Fundamentals
    - Operators
    - Identifiers
    - Literals
    - Data Types and Type casting
    - Java Statements
    - Arrays
    - Variable and Identifiers
    - Data Types and Type casting

    - Tokens and Literals in Java
    - Operators
    - Arrays and Strings in Java
    - User input with Scanner Class
    - Control Flow Statement in java
- Methods in Java
    - Create and Call Methods
    - Parameters and Arguments
    - Recursion in Methods
    - Method Overloading & Scopes
- OOPS
    - Introduction
    - Class and Objects
    - Attributes and Methods with example
    - Constructors and it's parameters
    - Modifiers
    - Encapsulations
    - Inheritance concept
    - Overloading Vs Overriding
    - Polymorphism in Java
    - Static, This and final Keyword in java
    - Abstract class and method in java
    - Interfaces in java
    - Enums in Java
    - Types of classes
    - Exceptional handling in Java
    - Regular expression (RegEx) in Java
    - I/O Streams in Java
    - Generics in java
- Packages:
    - What is a package?
    - Types of packages
- Reflection API:
    - Class
    - Field
    - Method
    - Constructor
- Garbage Collection:
    - Introduction
    - Approaches to make an object for GC
    - Methods for requesting JVM to run GC
    - Finalization

Advanced Java

- Collection Framework
    - Collection Architecure
    - List and its implementations

    - Hash Set
    - Hash Map
    - Queue and its implementations
    - Iterators
    - Collectors
- Java 8 features
    - Lambda Expressions
    - Stream APIs
    - Date and Time APIs
    - Java 8 features into collection framework
    - MultiThreading
- JSP and Servlet
    - Introduction to Web Development
    - Three layer architecture
    - Servlet Basics
    - Advanced Servlet Concepts
    - Introduction to JSP
    - Advanced JSP Concepts
    - Integrating JSP and Servlets
- JDBC
    - Introduction to JDBC
    - JDBC Basics
    - Advanced JDBC Concepts
    - Transaction Management
    - Connection Pooling
    - Error Handling in JDBC
    - Security in JDBC Applications
    - Deployment of JDBC
- Hibernate Framework
    - Introduction to Hibernate
    - Basic Hibernate Concepts
    - Mapping Relationships
    - Querying with Hibernate
    - Performance Optimization
- Spring and Spring Boot
    - Introduction to Spring Framework
    - Spring Core Concepts
    - Annotations and @Autowired in Spring
    - Spring AOP (Aspect-Oriented Programming)
    - Three-Layer Architecture in Spring
    - Spring Data Access
    - Spring MVC
    - Introduction to Spring Boot
    - Spring Boot Core Concepts
    - Building RESTful Web Services with Spring Boot
    - Spring Boot Data Access
    - Spring Boot Testing

- Spring Security
- Microservices with Spring Boot
- Advanced Spring Boot Features
- Capstone Projects

.Net

- C# Basic Programming
  - Introduction About C#
  - Data Types And Variables
  - Operators
  - Type Casting
  - If Else
  - Switch
  - Loop
  - Break And Continue
  - Array
  - String
  - Assignment And MCQ
- C#
  - Defining Methods In C#
  - Various Elements Of A Method
  - Methods Calling
  - Passing Parameters To A Method And Access Modifiers
  - Assignment And MCQ
- OOP Concepts in C#
  - Introduction About Oop
  - Classes And Object
  - Properties And Constructors
  - Data Abstraction
  - Encapsulation
  - Polymorphism
  - Inheritance And Interface
  - Assignment And MCQ
- Introduction to ASP.NET Core
  - Introduction
  - What Is ASP.NET Core?
  - ASP.NET Core Features
  - Advantages Of ASP.NET Core
  - MVC Pattern
  - Understanding ASP.NET Core MVC
  - ASP.NET Core Vs. ASP.NET MVC Vs. ASP.NET Web Forms
- ASP.NET Core First Application
  - ASP.NET Core Environment Setup
  - ASP .NET Core First Application
  - Project Layout
  - Understanding Life Cycle Of ASP.Net Core Request
- Controllers & Action Methods

    - Controllers Overview
    - Action Methods And IActionResult Object
    - Passing Data From Controller To View
    - Understanding Action Selectors
    - Action Filters
    - Building Custom Action Filters
    - Middleware
    - Asynchronous Action Methods
- Views
    - Introducing Razor View
    - Advantages Of Razor View
    - Razor Syntax
    - Types Of Views
    - Partial Views
    - Layout Pages
    - Special Views
    - View Categorization Based On Model
- Helpers
    - Html Helpers
    - Built-In Html Helpers
    - URL Helpers
    - Tag Helpers
    - Custom Tag Helpers
- Model Binding
    - Html Form Behavior
    - Model Binder Overview
    - DefaultModelBinder
    - Binding To Complex Classes
    - IFormCollection Model Binding
    - IFormFile Model Binder
    - Bind Attribute
    - TryUpdateModelAsync
- Validations & Data Annotations
    - Data Annotations And Validations Overview
    - Validations With Data Annotation
    - Server Side And Client Side Validation
    - Custom Server Side Validation
    - Model Level Validation Using IValidatableObject
    - Custom Unobstrive Client Side Validation
    - Remote Validation
- State management Techniques
    - Cookies
    - Sessions
- Security
    - Authentication And Authorization
    - Implementing Security Using ASP.NET Core Identity
- MVC and Entity Framework Core

        - Basic CRUD Operations Using Entity Framework
        - Separation Of Work Using BO Classes
        - Writing Generic Class / Repository
        - Caching In Repository
    - ASP.NET Core - Web Caching
        - Cache Tag Helpers
        - Memory Caching Introduction
        - In-Memory Caching
        - Response Cache
        - Distributed Cache
    - Routing
        - Url Routing Overview
        - Custom Routes
        - Attribute Routing
        - Routing Constraints
    - Module Development
        - Understanding Areas
        - Adding Areas
        - Defining Area Routes
        - Linking Between Areas
    - Web API and JQuery Ajax
        - Introduction To Web API
        - AJAX Implementation Using JQuery
        - Calling The Web API With JQuery Ajax
        - Creating A Web API That Supports CRUD Operations Using EF
    - Bundling & Minification
        - What Is Bundling And Minification In ASP.Net Core?
        - Bundler And Minifier Extension
        - How To Bundle Your Files
        - How To Minify Your Bundles
        - Convert To Gulp

Module 7: AI Implementation in Full Stack Development

Introduction To AI & ML

- What is AI? What is ML? What is Deep Learning?
- AI vs ML vs DL
- Where AI fits in web development
- Real-world examples
- Objective: Build interest and clarity in the AI landscape.

Weka – ML For Beginners

- Introduction to Weka
- Explore Weka GUI: load CSV, run classifiers

Encog – Lightweight Neural Networks In Java

- Introduction to Encog
- Creating a small feedforward neural network

DL4J – Deep Learning In Java

- What is DL4J (DeepLearning4J)
- Understanding layers, activations, loss functions

Spring AI API Overview – Modern AI Services

- What is Spring AI
- Overview with OpenAI, Hugging Face, and others
- Prompt templates and responses

Course Projects

Domain: Healthcare Technology

Project Name:
MediConnect

The Distributed Task Scheduler efficiently allocates, schedules, and monitors tasks across multiple systems using Java multithreading and distributed computing. It ensures high availability, fault tolerance, and optimized resource use, while offering real-time insights into task execution and node performance.

Tool & Technology Used:

JAVA SE & MULTITHREADING | APACHE ZOOKEEPER | SPRING BOOT | HIBERNATE WITH MYSQL

Domain: Financial

Project Name:
AI-Powered Expense Tracker

The AI-Powered Expense Tracker is a Java application using machine learning to categorize expenses, offer spending insights, and suggest budgets. It securely integrates with bank APIs and delivers real-time financial analytics for better money management and decision-making.

Tool & Technology Used:

JAVA SE & SPRING BOOT | TENSORFLOW JAVA API | MYSQL WITH JPA | RESTFUL APIS

Domain: Security & Surveillance

Project Name:
Safety & Security Guard Management System

The Safety & Security Guard Management System streamlines operations by tracking guard assignments, managing shifts, handling emergencies, and monitoring performance. It enhances workflows, maintains security protocols, and improves communication for both security personnel and administrators.

Tool & Technology Used:

HTML | CSS | JAVASCRIPT | JQUERY | BOOTSTRAP | REACTJS | NODEJS | EXPRESSJS | MONGODB

Domain: Human Resource

Project Name:
HRMatrix

HRMatrix is an all-in-one HR management application offering employee data handling, payroll, attendance, recruitment, performance tracking, training, benefits, and compliance. It boosts efficiency, minimizes administrative tasks, and supports strategic HR goals through automation, analytics, and complete employee lifecycle management.

Tool & Technology Used:

HTML | CSS | JAVASCRIPT | JQUERY | BOOTSTRAP | REACTJS | NODEJS | EXPRESSJS | MONGODB

Course Projects

Domain: E-Commerce

Project Name:
TradeCatalyst (E-commerce Website)

This MERN stack e-commerce platform offers a seamless shopping experience, enabling users to browse products, add to cart, make secure payments, and manage orders. It features a user-friendly interface, real-time updates, and robust backend support for efficient order processing.

Tool & Technology Used:

HTML | CSS | JAVASCRIPT | JQUERY | BOOTSTRAP | REACTJS | NODEJS | EXPRESSJS | MONGODB

Domain: Library Management System

Project Name:
LearningSphere System

LearningSphere is a desktop-based Library Management System that streamlines book inventory, issuance, returns, and overdue tracking. It offers a user-friendly interface for CRUD operations on books and students, ensuring accurate data handling through robust database integration for efficient library operations.

Tool & Technology Used:

JAVA SE | MYSQL | JDBC & MAVEN

e-Learning through LMS

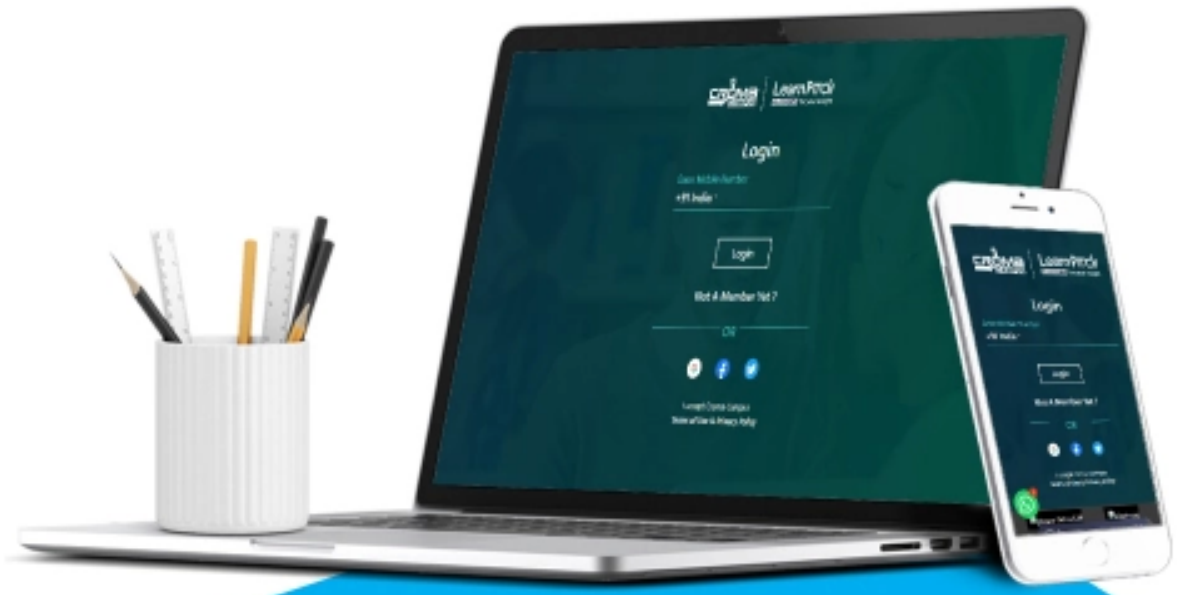
Learning **Management System**

Our LMS (LearnPitch) is for the administration, documentation, tracking, reporting, automation, and delivery of educational courses, training programs, or learning and development programs.

Our LMS has been designed to identify training and learning gaps, using analytical data and reporting to keep you up with the class activities.

Key Features **Learning Management System**

- Live Sessions with Class recordings
- Get study material with Assignments.
- Track your curriculum covered.
- Track your class wise attendance
- Share your feedback for Trainer & Training
- Get your Training Certificate from LMS



Training Certification

Earn Your **Certificate**

Your certificate and skills are vital to the extent of jump-starting your career and giving you a chance to compete in a global space.

Croma Campus is Nasscom Certified

Croma Campus is now

NASSCOM®

Certified Member


nasscom

CERTIFICATE

OF MEMBERSHIP

CromaCampus Training & Development Private Limited

Membership ID : NSCM/2023/33/7617

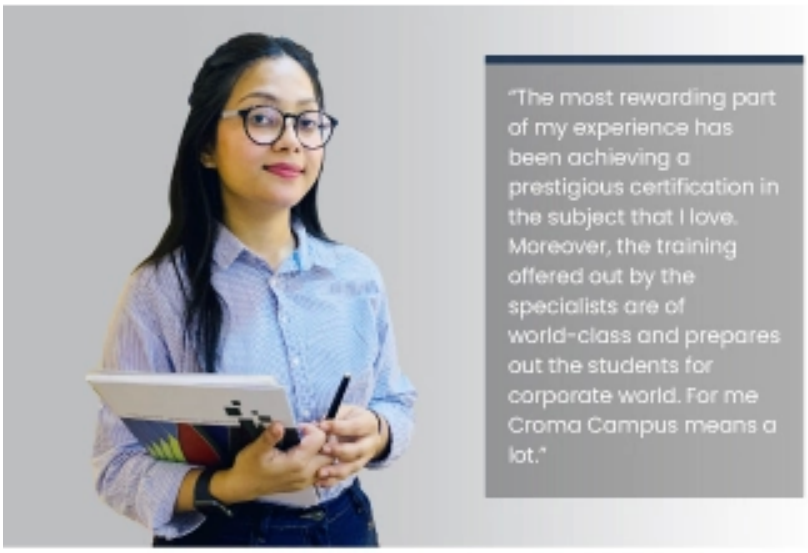
Valid Till : 31-March-2020

Rajesh Nambiar

President, nasscom

www.cromacampus.com

|Croma Campus! Reviews



"The most rewarding part of my experience has been achieving a prestigious certification in the subject that I love. Moreover, the training offered out by the specialists are of world-class and prepares out the students for corporate world. For me Croma Campus means a lot."

"By The Students For The Students"

Your Success is **Our Story**

Bharat

"I am fully satisfied with the excellent training services received by the expert staff at Croma Campus. I want to thank Croma Campus for providing me with the most innovative and affordable training services for learning all the software testing procedures and guidelines."

Ankit

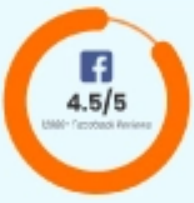
"It was a lifetime experience for me to get trained by IT Experts of Croma Campus. What I liked most about the training was the consistent high-quality education, which was friendly and co-active. The placement department was also proactive, they keep me updated regarding new job opportunities and provide the grooming session to crack the interview. At last, I would like to thank all faculty members of Croma Campus for their immense help and support."

Umesh

"Without any second thought, I will give Croma Campus 10/10. Their placement department is highly proactive. I remember they started scheduling interviews for me from the very next day when I told them my course has been completed. These people are doing a phenomenal job and I highly recommend Croma Campus to everyone."

Shams Khan

"Croma Campus is doing a phenomenal job in the IT training industry. The reason why I decided to join their training program was that they provide quality training at very a nominal price. Plus, the online training mode was also a factor due to which I decided to join the training program of Croma Campus as I didn't want to attend physical classes."



Meet Our Team

Sales Team

Our Sales team is highly passionate, emphatic, positive attitude, great listening skills, ability to deliver quick solutions, and they are multitasker too. Our team always remains up-to-date about all the latest technologies and market trends. With effective communication skills, they always work to deliver the right information to customers when it is needed.

Product Team

Our product team is highly functional and collaborative working together to achieve the common outcome of designing exceptional digital experiences. Each of our members is a contributor to help us achieve success in long-run. Sitting at the high-end of technology and innovation, team helps to deliver high-end customer experiences and always comes out with a big idea as a game-changing plan.

Marketing Team

Our Marketing team works as gladiators and helps us to achieve business success in all possible ways. They are included in almost everything either it is building a brand, creating brand awareness, promoting products or services, delivering trailblazing customer experiences or increasing engagement at public forums. They are the true backbone of the Company.

Content Team

Our content team is responsible for ideation, creation, optimization, and distribution of content throughout the company. The team always starts its work with a strategy, how to create high-quality contents, and how to promote or share the content. Our in-house content team help us to produce all types of contents either they are educational content pieces, marketing content, SEO content, or any other forms too.

Customer Access Team

This is the team that has actually been taken up us from reactive state to a pro-active state. The team utilizes high-valued solutions to satisfy customers in all possible ways. It is truly said that no company can succeed if your customers are not satisfied. And our customer success team is dedicatedly working to keep all the customers satisfied and we always consider our customer feedback on priority.

HR Team

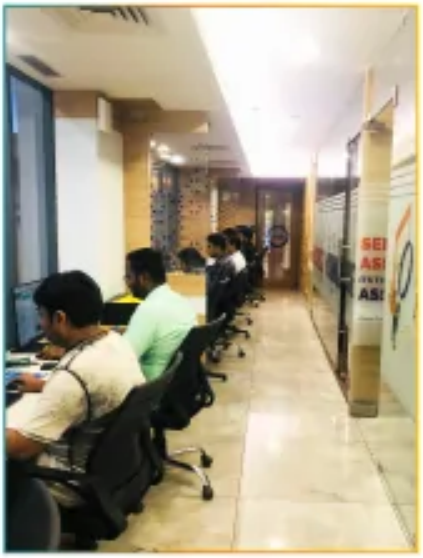
Our HR team is committed to provide high-end solutions to employees as they require. Our HR team has the right skills and knowledge to make sure that the HR department can always be legally and strategically successful. They know how to keep employees motivated all the time with the best HR policies and fun activities too from time to time.

|Glimpse Of Our Office

Look **Who We are**

Our office's infrastructure comprises all the necessary software and network resources that are required to deliver IT & Design, Human Resources, Digital Marketing, and training services.

We are well-equipped with bright designed work bays for employees and managers having separate cabins with spacious cafeteria and training classrooms.



About Croma Campus

“Our Mission is to Build Nation through Education & Beyond Limitation.”

CROMA CAMPUS #1 Asia's Leading Edtech Company

Croma Campus Training & Development Private Limited is an education platform providing rigorous industry-relevant programs designed and delivered in collaboration with world-class faculty, industry & Infrastructure. In the past 15 years we have trained 18000+ candidates and out of which we are able to place 12000+ professionals in various industries successfully.

We're Here to Help –

Reach Out to Our Global Offices

India

Croma Campus Training & Development Private Limited

G-21, Block-G, Sector 3, Noida, Uttar Pradesh-201301

+91-971 152 6942 Chat on WhatsApp

View on Google Maps

Canada

Croma Campus Training & Development Private Limited

279-B Baltzan BLVD, Saskatoon, Saskatchewan, Canada-S7W0S1

+1 (647) 370-6470 Chat on WhatsApp

View on Google Maps

USA

Croma Campus Training & Development Private Limited

454 church street Cairo, GA-39828, United States of America

+1 (647) 370-6470 Chat on WhatsApp

View on Google Maps

follow us on:

CromaCampusNoidaOfficial/facebook

cromacampus/pinterest

cromacampus/instagram

cromacampusofficial/linkedin

cromacampus/twitter

CromaCampusOfficial/youtube