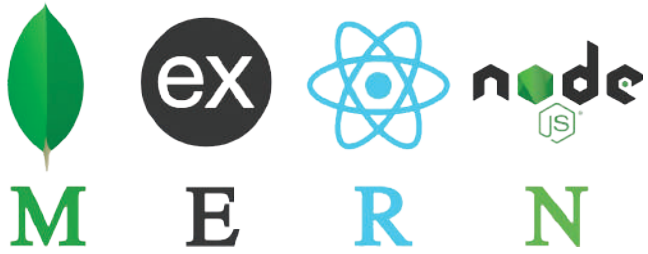
incapp®
The CODING Institute



MERN

Get the skills to get ahead, stay relevant and earn more

Book your seat now

Google
Reviews 2,500+ • Excellent

MSME

13+ YEARS EXPERIENCE

CERTIFIED ISO 9001:2015 COMPANY

12800+ STUDENTS TRUST US

About INCAPP

INCAPP Coding Institute, established in 2011, was founded with the goal of addressing the global tech skills shortage. Our commitment lies in offering high-quality training programs to students, professionals, and organizations. We strive to empower individuals with coding skills, facilitating personal and professional growth, and assisting organizations in enhancing their workforce's productivity and effectiveness.
Our company boasts a team of seasoned instructors, experts in their fields. We employ the latest teaching methodologies and technologies to provide engaging and interactive training programs.

> We foster innovation and empower aspiring coders. As founders, we are excited to welcome you aboard. Whether you're new to coding or already experienced, our hands-on curriculum and expert instructors will guide you. Coding is more than just writing lines; it involves creativity and problem-solving. Embrace challenges and celebrate your successes, knowing that coding is a journey of continuous growth. Let's get started!

RAHUL CHAUHAN
Co-Founder & Instructor

PRAVEEN CHAUHAN
Co-Founder & Instructor

Oracle & Microsoft Certified

How We Help You To Learn

1. **Expert Instructors**
   Top-class instructors, experts in their fields, teach through practical training.

2. **Assignments**
   Understand all concepts through well-structured assignments.

3. **Doubt Resolutions**
   Dedicated assistance provided to clarify doubts, featuring two types of instructors: Class Instructor and Lab Instructor.

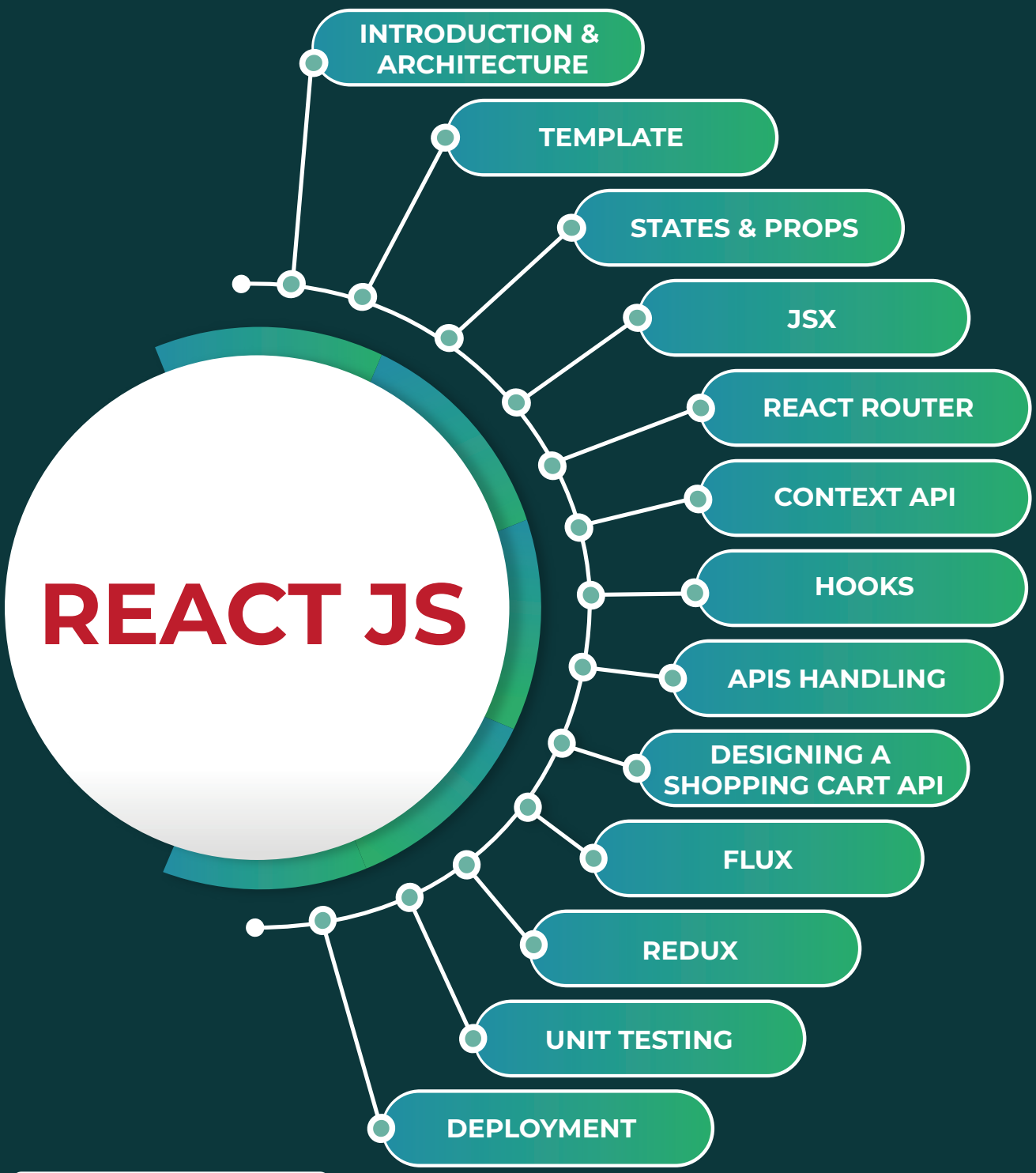
4. **Projects**
   Gain a comprehensive understanding of the technology through project work, guided by your instructor.

Why INCAPP Coding Institute

Outstanding students deserve the finest learning environment. At INCAPP, we guarantee a superior learning experience and personalized support to ensure your success.

- Top-Notch Classroom with Expert Instructor
- Comprehensive Study Materials
- Continuous Feedback and Monitoring
- Guaranteed Course Completion
- Project-Based Learning
- Course Completion Certification
- Dedicated Support for Doubt Resolution
- Placement Assistance
- Individual Attention to Each Student
- In-Class Assignment Sessions

What You Will Learn



PROJECT WORK

5 Reasons To Learn React JS

- **Component-Based Architecture:** Facilitates reusable UI components, making code more manageable and scalable.
- **Strong Community Support:** React has a large, active community and an extensive ecosystem of libraries and tools.
- **High Performance:** Offers efficient rendering and a virtual DOM for improved web application performance.
- **Widely Used in Industry:** Popular among tech giants, ensuring strong job prospects and industry relevance.
- **Ease of Integration:** Can be seamlessly integrated with other frameworks and technologies for flexible development.

CURRICULUM

Course Overview:

React JS is a popular JavaScript library used for building user interfaces, particularly for single-page applications. It allows developers to create reusable UI components, enabling efficient and dynamic rendering of web pages. React's declarative approach simplifies the process of building interactive user interfaces and managing the state in web applications. It's widely used for its high performance, flexibility, and compatibility with other libraries or frameworks, making it a favored choice among developers for creating responsive and visually engaging web applications.

REACT JS

Introduction to ReactJS

- What is ReactJS?
- Why ReactJS?
- Features of ReactJS
- Scope of ReactJS
- ReactJS Versions
- What is a Single Page Application?
- Why Single Page Application?

Getting Started with ReactJS

- ReactJS Installation
- Understanding ReactJS Architecture
- Creating First ReactJS Hello World Application
- Debugging the First ReactJS Application
- Executing the First ReactJS Application

ReactJS Basics

- Finishing the Hello World Application
- Delete and recreate everything
- Adding CSS to Hello World Application
- Reusable Components

ReactJS Templates

- Understanding the existing template
- Converting the HTML template into React App
- Reusable Card and Assignment

ReactJS States and Props

- What are states?
- What are props?
- Preparing the state-based applications
- Making the Counter Application
- Assignment for Counter Application

ReactJS JSX

- Introduction to Virtual DOM
- DOM Vs Virtual DOM
- What is ReactJS JSX?
- Difference between JS and ReactJS JSX

CURRICULUM

- Rendering Elements
- Components Overview
- Types of Components
- Function Components
- Class Components
- Nested Components
- Using Newly Created Components
- Components Collections
- Styling
- Properties
- Creating Components using Properties
- Properties Validation
- Constructor
- Component Life Cycle
- Conditional Rendering
- Dynamic Data Rendering
- Property Binding
- Lists and Keys
- Forms Handling
- Forms Validations
- Understanding Events and Event Handling
- Passing arguments to Event Handlers
- Material UI for Designing
- Formik Validation
- Local Storage
- Cookies
- Session

Routing using React Router

- Setting up the React Router
- Installing the React Router
- Understanding the routing in single page application
- Working with Browser Router Components
- Configuring route with Route Component
- Making routes dynamic with Route Params
- Working with nested routes
- Navigating through pages using Link and NavLink Components
- Redirect Routes using Redirect Component

React Context API With Projects

- The problem that contextAPI solves
- Details on Context and Provider
- Details on Consumer in Context API
- Understanding the working of dark and light mode
- Creating a theme Toggler with Context API
- Finishing the theme switcher app

Context API using Reducers and Actions

- What are we building here?
- Create brain of the application
- Use Reducer for our app
- Sending a dispatch
- Display the context data and dispatch

Hooks

- Introduction to Hooks
- Understanding the need of Hooks
- Types of Hooks
- useState Hooks
- useEffect Hooks
- useContext Hooks
- useRef Hooks
- useReducer Hooks

CURRICULUM

- useCallback Hooks
- useMemo Hooks
- Custom Hooks
- Hooks Rules

Flux

- Flux Introduction
- Flux Architecture
- Flux Working
- Flux Components
- Stores
- Dispatchers
- View Controllers
- Actions
- Views
- Understanding the working of React and Flux together

Redux

- What is Redux?
- Why Redux
- Installation and Setup
- Store
- Reducer
- Actions
- Provider Component
- High order Components
- Dispatchers
- View Controllers
- Selector
- Asynchronous Actions
- Middleware
- Redux Thunk
- Understanding Redux Saga
- Installing and setting up redux saga
- CRUD Operations using redux saga
- Saga Vs Promises

Third Party Modules

- Pagination
- Searching
- Filtration
- JWT Token
- Social Login
- File Uploading

API Handling

- Learn to read docs for API
- Let's read Axios docs
- Drill down the API
- Extracting information from API

Designing a Shopping Cart API

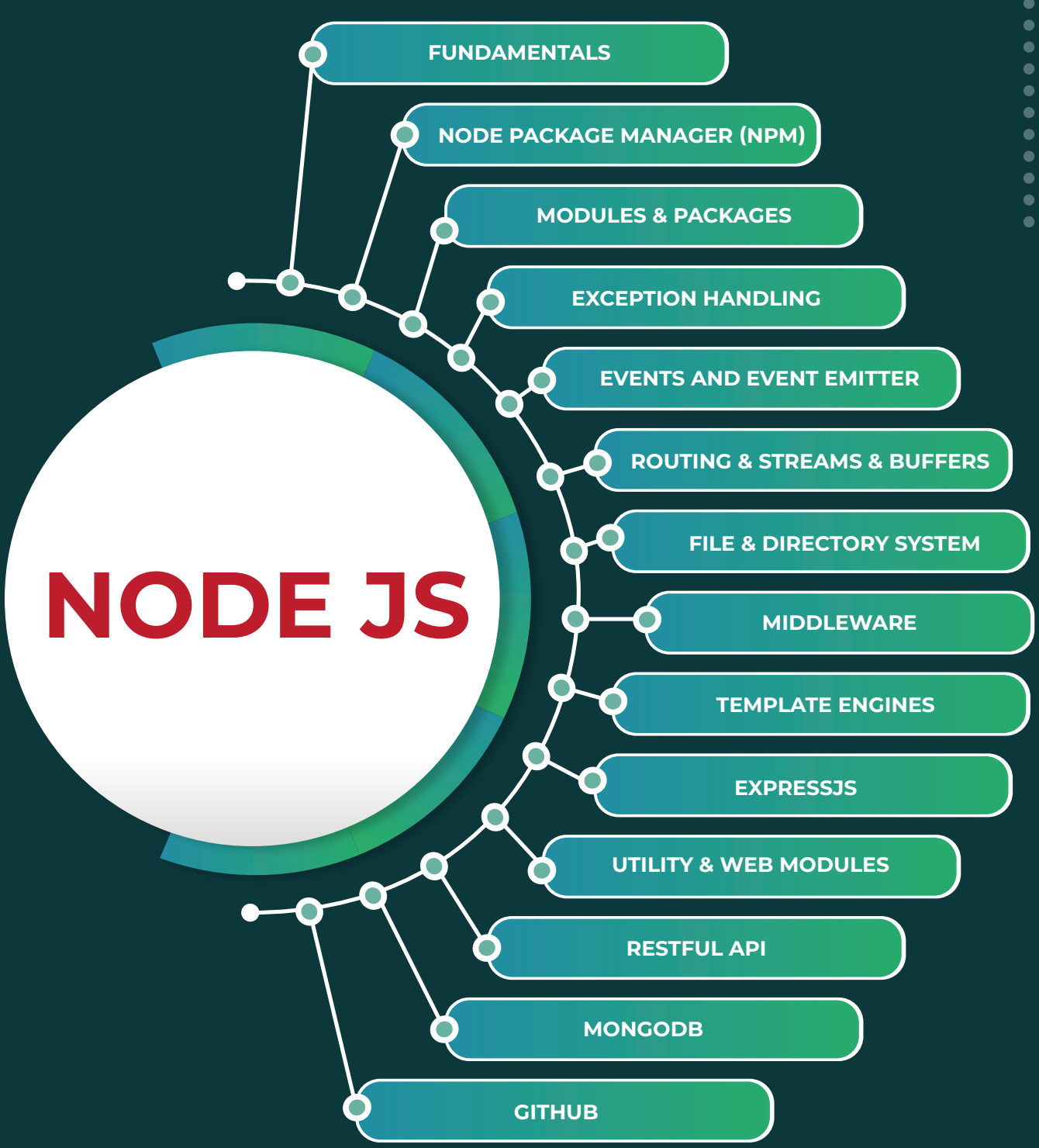
- A walk of Pexels and JSON
- Buy the item and remove the item
- Fetching photos from the API
- Storing everything as state
- Card for every product
- Create cart section
- Bring the shop together
- Removing the duplicate

ReactJS Unit Testing and Deployment

- Tools required for Unit Testing
- React Unit Testing overview
- Introduction to JEST
- Introduction to Enzyme
- Testing ReactJS Application using JEST and Enzyme
- Deploying the ReactJS Application

Project Work

What You Will Learn



PROJECT WORK

5 Reasons To Learn Node JS

- **Efficient Performance:** Utilizes non-blocking I/O model, ensuring fast and efficient performance, especially for data-intensive applications.
- **JavaScript on Server-Side:** Enables using JavaScript for both client and server-side, simplifying development and reducing context-switching.
- **Strong Community Support:** Benefiting from a vast and active community, offering a wealth of libraries and tools.
- **Suitable for Building Scalable Applications:** Ideal for developing scalable network applications due to its event-driven architecture.
- **Versatility in Web Development:** Facilitates the development of various types of web applications, from APIs to full-fledged web servers.

CURRICULUM

Course Overview:

Node.js is an open-source, cross-platform JavaScript runtime environment that enables the execution of JavaScript code server-side. It's designed for building scalable network applications, particularly web servers. Node.js uses an event-driven, non-blocking I/O model, making it lightweight and efficient for data-intensive real-time applications that run across distributed devices. Its ability to handle numerous simultaneous connections with high throughput makes it ideal for developing applications like chat applications, online gaming, and real-time data processing systems.

NODE JS

Introduction to NodeJS

- NodeJS Introduction
- History of NodeJS
- Features of NodeJS
- Scope of NodeJS
- NodeJS Applications
- Advantages of NodeJS
- NodeJS Versions
- What is V8 JavaScript Engine?
- Why Server-side JavaScript?
- NodeJS Vs Other server-side technologies

Getting Started with NodeJS

- NodeJS Architecture
- NodeJS Installation
- Creating Web Server in NodeJS
- Creating First NodeJS Application
- Debugging First NodeJS Application
- Event Loop
- Event Driven Architecture
- Node REPL
- Writing Asynchronous Code
- Blocking vs Non-Blocking Code

Modules and Packages in NodeJS

- Understanding Modules
- Understanding require and exports
- Creating Modules
- Importing Modules
- Exporting Modules
- Built-in Modules
- Process Module
- HTTP Module
- URL Module
- Creating a Node Package
- Publishing the Package
- Using published package

Node Package Manager (NPM)

- Introduction to NPM Package
- Understanding CLI
- Understanding NPM
- Installing Modules using NPM
- Local and Global Packages
- Installing a Module
- Updating a Module
- Uninstalling a Module
- Working with Node's Package Manager (NPM)
- Understanding package.json
- Using package.json
- Attributes of package.json

NodeJS Built-in Packages

- NPM Package
- Express NPM Package
- Multer NPM Package
- Node Mail NPM Package
- Boot Strip NPM Package Integration
- JSONWEBTOKEN NPM Integration
- Introduction of Crypto-JS

Callback

- What is Callback?
- Blocking Code Example
- Non-Blocking Code Example

Code Debugging

- Built-in Debugger
- IDE Debugger
- Node Inspector

Exception Handling

- Try-Catch
- Call Back

Events and Event Emitter

- Understanding Events
- Event-Driven Programming
- How do Node Applications work?
- Event Emitter Class

Methods

- Class Methods
- Emitting Events
- Listening to Events

HTTP

- Building a Web Server
- HTTP Request Methods
- HTTP Request Headers
- HTTP Response Codes
- HTTP Response Headers

ExpressJS

- Introduction to ExpressJS Framework
- Installing ExpressJS Framework
- Building a Web Server
- Creating Hello World Application using ExpressJS
- Debugging and Executing Hello World Application
- Request and Response
- Request Object
- Response Object
- Basic Routing
- Serving Static Files
- GET Method
- POST Method
- FILE Upload
- Cookies Management
- Sending Emails

Routing

- Understanding Routing
- Router Object
- Route Methods
- Route Paths
- Parameterized Routes
- Route Handlers
- Express Router

Streams

- Understanding Streams
- Types of Stream
- Creating Streams
- Readable Streams
- Writable Streams
- Piping the Streams
- Chaining the Streams

Buffers

- Creating Buffers
- Writing to Buffers
- Reading from Buffers
- Convert Buffer to JSON
- Compare Buffers
- Copy Buffer
- Slice Buffer
- Buffer Length
- Method Reference
- Class Methods

File and Directory System

- Synchronous Vs Asynchronous
- Introduction to fs Module
- File operations
- Creating a File
- Opening a File
- Writing to a File
- Reading from a File
- Closing a File
- Deleting a File
- Creating a Directory
- Reading a Directory
- Removing a Directory

Middleware

- Middleware Introduction
- Middleware Types
- Express Middleware
- Error Middleware
- Body Parser
- Cors
- Cookie Parser
- Session Management

Template Engines

- Introduction to Template Engines
- EJS
- Jade
- Vash
- GruntJS
- Handlebars

ExpressJS Security

- Authentication
- JWT
- Securing Routes
- Debugging in ExpressJS

Global Objects

- __filename
- __dirname
- setTimeout(cb, ms)
- clearTimeout(t)

CURRICULUM

- setInterval(cb, ms)
- Global Objects
- Console Object
- Process Object

Utility Modules

- OS Module
- Path Module
- Net Module
- DNS Module
- Domain Module

Web Modules

- What is a Web Server?
- Web Application Architecture
- Creating a Web Server using Node
- Making a Request to Node Server
- Creating a Web Client using Node

RESTful API

- What is REST Architecture?
- Http Methods
- RESTful Web Services
- Creating RESTful for a Library
- List Users
- Add Users
- Show Detail
- Delete a User

Introduction to MongoDB

- Understanding NoSQL DB
- NoSQL Vs SQL DB
- Understanding MongoDB
- Document-oriented Vs Other kind of storages
- Installing MongoDB
- MongoDB Data Types
- MongoDB Shell Commands
- Understanding DB, collection and document
- Understanding Embedded documents
- Querying Database Tools and API
- MongoDB Tools

CRUD Operations on MongoDB

- Creating Database
- Creating Collections
- Creating Documents
- Inserting Data
- Querying Data
- Updating Data
- Deleting Data
- Limiting Data
- Sorting Data
- Dropping Collection
- Dropping Database

MongoDB Indexing and Relationships

- Types of Indexes
- Creating an Index
- Defining Relationships between Documents
- Dropping an Index

MongoDB Mongoose

- Introduction to Mongoose
- Mongoose Schemas
- Mongoose Data Types
- Mongoose Models
- Mongoose Relationships
- Mongoose CRUD Operations

MongoDB ODM Mongoose

- Introduction to ORM
- Introduction to ODM
- MongoDB ODM Mongoose

GitHub

- Real-time Environment setup with GitHub

REST API with Mongoose

- REST API with Mongoose, MongoDB and Postman
- Creating REST API using Express and Mongoose

Project Work

- Developing the Project
- Deploying the Project

Our Impact

12800+

Students Successfully Trained

13+

Years of Experience

99%

Students Recommend Us to Their People

100%

Course Completion Rate

100+

Projects Developed

4.9/5

Google Rating

What our students say about us

I had learned core java, web developement, advance java from the Rahul sir and i am very grateful to him for giving us the best knowledge.

Isha

I did web designing course by Incapp and this is the best institute in G.Noida. It has really helped me understand and feel more confident in what I am doing. Thanks Rahul sir.

Vishal Kumar

The class is well maintained and i like what mam teaches us. We have learned programming language. Thankyou INCAPP for best coding experience.

Pratyush Mishra

The environment of coaching is very good. Notes, PDFs, & books everything is provided. Notes are really very nice. Overall, coaching is very good.

Khushi Mundra

Incapp is a best institute for coding, here staff is very supportive and our Praveen Chauhan Sir explain every topic very well.

Anand Kumar

I had a fantastic learning experience at Incapp! The instructors were highly knowledgeable and skilled in their respective fields. I would highly recommend Incapp to others.

Kartik Chauhan

Our Students Are Place In

Everyone should learn how to program a computer, because it teaches you how to think.

- **Steve Jobs**

All of my friends who have younger siblings who are going to college or high school - my number one piece of advice is: You should learn how to program.

- **Mark Zuckerberg**

FAQ's

What are the criteria for admission?

No criteria, anybody who has an interest in coding can join.

Do you Provide Study material?

Yes, Immerse yourself in a superior learning experience with study materials meticulously crafted by our expert instructors.

Do I need to be good at maths to complete this program?

No, Only your dedication and ambition about learning is needed.

Do I get a certificate after course completion?

Yes. Upon successful completion of the course, you will be awarded a prestigious certificate that validates your achievement.

Is coding difficult?

No, it is not difficult. Coding is fun and challenging as you learn to create apps, games, websites, and lots more out of your creativity.

Are there tests/exams in the program?

Yes, In between the course, your instructor conducts the test to monitor your performance.

Courses we offer

- Java Language
- Oracle Certified Foundation Associate
- Data Structure & Algorithms
- Spring Boot Framework
- Website Development
- Python Language
- Data Science
- Django Framework
- C Language
- C++ Language
- CFP Coding Foundation Program
- FSSD Full-Stack Software Development
- React JS
- Node JS
- MERN

incapp®
The CODING Institute

Are you ready to transform your career?

Our course may be demanding, but the incredible transformation you can experience will make it all worthwhile!

Get In Touch

0120-4108484, 9811272031

info@incapp.in

www.incapp.in

5th Floor, OM TOWER, Commercial Belt,
Alpha I, Greater Noida, UP

Follow us on: /incapp /incapp.in /incapp