Mern Stack Development

CODE INTEL
ACADEMY

Premier Institute For Computer Science
Where Dreams Becomes True.

CODE INTEL
ACADEMY

Mern Stack Development
ROADMAP
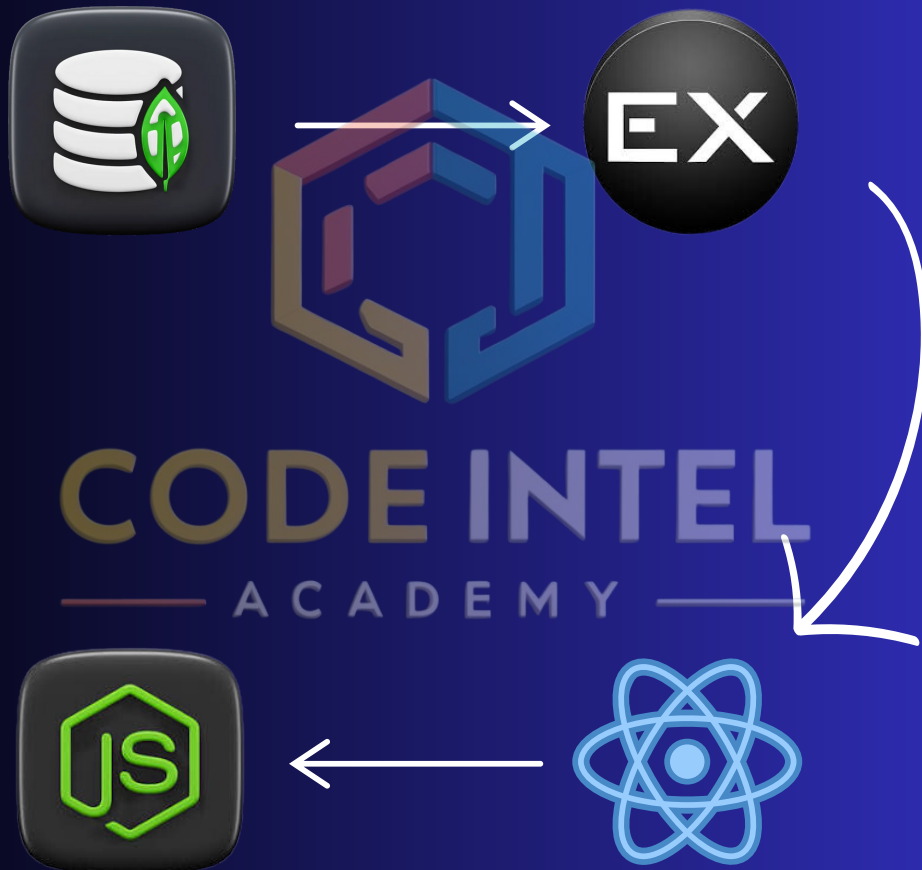
MongoDB Syllabus

Module Objectives:

• Building strong expertise in Document Oriented Non-Relational Database Management System

• Implement frontend and backend scenarios to read, write and update data stored in MongoDB

Overview:

This course provides you hands-on experience and exposure to implement frontend and backend scenarios to read, write and update data using MongoDB. This course builds strong foundation for web application development based on client-server architecture.

Platform:

• Linux (Fedora / Mandriva / Ubuntu) or Windows Host system

• Firefox, Chrome

Prerequisites:

Academic level web application knowledge

Delivery method:

Instructor lead, hands-on exercises backed with assignments and mini project

Detailed course contents:

Module 1:CRUD Operations Using MongoDB

- Installing MongoDB
- Connecting to MongoDB
- Schemas
- Models
- Saving a Document
- Querying Documents
- Comparison Query Operators
- Logical Query Operators
- Regular Expressions
- Counting
- Pagination
- Updating Documents- Query First
- Updating a Document- Update First
- Removing Documents

Module 2:Mongoose - Data Validation

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

Module 3:Mongoose- Modelling Relationships Between Connected Data

- Referencing Documents Population
- Embedding Documents
- Using an Array of Sub-documents
- Transactions
- Object ID
- Validating Object ID's

Module 4:Authentication and Authorization

- Introduction
- Creating the User Model
- Registering Users
- Using Lodash
- Hashing Passwords
- Authenticating Users
- Testing the Authentication
- JSON Web Tokens
- Generating Authentication Tokens
- Storing Secrets in Environment Variables
- Setting Response Headers
- Encapsulating Logic in Mongoose Models
- Authorization Middleware
- Protecting Routes
- Getting the Current User
- Logging Out Users
- Role Based Authorization
- Testing the Authorization

Handling and Logging Errors Deployment

- Introduction
- Preparing the App for Production
- Getting Started With Heroku
- Preparing the App for Deployment
- Adding the Code to a Git Repository
- Deploying to Heroku
- Viewing Logs
- Setting Environment Variables
- MongoDB in the Cloud

CODE INTEL
ACADEMY

Express Framework Syllabus

Module Objectives:
• Building Strong expertise on Express framework to develop responsive web application
• Implement MVC and responsive design to provide unified and intuitive user experience across PC, tablet and Mobile Phone

Overview:
This course provides you strong foundation on Express framework driven by hands-on experience and exposure to develop
responsive web application using MVC design pattern. During the course, you should be able to builds strong background
about Express and Node.js architecture, usage of services and applying Node.js concept to develop frontend application
supported by backend web servers.

Platform:
• Linux (Fedora / Mandriva / Ubuntu) or Windows Host system
• Firefox, Chrome

Prerequisites:
Academic level web application knowledge

Delivery method:
Instructor lead, hands-on exercises backed with assignments and mini project

Address: 13, First Floor, Jda market, Riddhi Siddhi Crossing, Gopalpura By Pass Road , Jaipur 302018

CODE INTEL
ACADEMY

Detailed course contents:

Module 1:Express Framework - (Building RESTful 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

Module 2:Express - Advanced Topics

- Middleware
- Creating Custom Middleware
- Built-In Middleware
- Third-party Middleware
- Environments
- Configuration
- Debugging

Module 3:Asynchronous JavaScript

- Synchronous Vs. Asynchronous Code
- Patterns For Dealing With Asynchronous Code
- Callbacks
- Callback Hell
- Named Functions to Rescue
- Promises
- Replacing Callbacks with Promises
- Consuming Promises
- Creating Settled Promises
- Running Parallel Promises
- Async and Await

CODE INTEL
ACADEMY

ReactJS Syllabus

Module Objectives:
• Create Single Page Applications (SPA) using React
• Learn core React features and use-cases
• Integrate React Applications into Firebase backend
• Deploy React applications into popular servers

Overview:
This course provides you building SPA using React and its core features by taking a hands-on approach. Starting
with Introduction this course deep dives into core React features like components, state and event binding and
rendering. Followed by core features, participants will learn to integrate React with backend services (ex: Firebase)
and consuming HTTP services in forms of REST APIs. The course exposes participants with advanced features like
Routing, Authentication and authorization followed by deploying into popular servers (ex: GITHub)

Platform:
• Linux (Fedora / Mandriva / Ubuntu) or Windows Host system
• Firefox, Chrome Browsers
• Firebase (Backend integration)

Prerequisites:
• OOPS concepts
• Knowledge of HTML, CSS and JavaScript

Delivery method:
Instructor lead, hands-on exercises backed with assignments and mini project

Address: 13, First Floor, Jda market, Riddhi Siddhi Crossing, Gopalpura By Pass Road , Jaipur 302018

Detailed course contents:

Modulel 1: Introduction to React

- History of React
- Key Benefits of React
- React development environment
- Creating your first React Application
- React Source code structure

Modulel 2: JSX

- Introduction to JSX
- Coding in JSX
- Expressions in JSX
- Working with HTML
- Conditional Constructs

Modulel 3: Components

- Introduction to components
- Why Components?
- Writing JSX code in components
- Adding CSS
- Populating Data Dynamically
- Passing data through "props"
- Multiple Components

Modulel 4: Components

- Introduction to Events
- Event Handlers
- Working with state
- Data Binding
- Controlled and Uncontrolled Components

Modulel 5: State and Event Binding

- Introduction to Events
- Event Handlers
- Working with state
- Data Binding
- Controlled and Uncontrolled Components

Modulel 6:Rendering Lists and Conditional Contents

- Rendering lists of data
- Using stateful list
- Keys in data
- Conditional Contents
- Adding dynamic styles

Modulel 7:Debugging Techniques

- Understanding error message
- Code flow and warnings
- Breakpoints
- Using React Dev Tools

Modulel 8:Class based Components

- What and Why?
- Adding first class based component
- Working with states and events
- Component Life cycle and in action
- Class based vs functional component

ModuleI 9:HTTP

- **Introduction to HTTP**
- **Methods in HTTP**
- **Response code in HTTP**
- **Introduction to REST interfaces**
- **Characteristics of REST**
- **Introduction to JSON**
- **JSON data representation**
- **GET request**
- **Using async and await**
- **Handling Http errors**
- **useEffects()**
- **POST request**

ModuleI 10:Custom React Hooks

- **What are custom hooks?**
- **Creating and using custom hooks**
- **Custom HTTP hooks**

ModuleI 11:Forms

- **Introduction to Forms**
- **Working with user input and submission**
- **Adding validation**
- **Working with custom hooks**

ModuleI 12:Routing and Navigation

- **Introduction to Single Page Applications**
- **What is Routing?**
- **Why Routing?**
- **Installing react router**
- **Defining and using routes**
- **Working with links**
- **Dynamic routes with Params**
- **Nested routes**
- **Redirecting the user**
- **Working with query Param**
- **Sending and getting data via HTTP**

ModuleI 13:Authentication

- **What is authentication?**
- **Why and How?**
- **Authentication tokens**
- **Setting up**
- **Adding signup**
- **Showing feedback to the user**
- **Adding User login**
- **Managing authstate with context**
- **Using tokens**
- **Redirecting the user**
- **Adding logout**
- **Protecting frontend pages**
- **Persisting User authentication status**
- **Adding auto logout**

ModuleI 13:Deployment

- **Introduction**
- **Deployment steps**
- **Adding lazy loading**
- **Building the code for production**
- **Getting started with deployment**
- **Handling routes and finishing deployment**

Node.js Syllabus

Module Objectives:
• **Building Strong expertise on Node.js core to develop web application**
• **Implement web application and deployment**
Overview:
This course provides you strong foundation on Node.js driven by hands-on
experience and exposure to develop NodeJS
based application and deployment. During the course you should be able to
builds strong background about Node.js
architecture, usage of services and applying Node.js concept to develop
frontend application supported by backend web
servers.
Platform:
• **Linux (Fedora / Mandriva / Ubuntu) or Windows Host system**
• **Firefox, Chrome**
Prerequisites
Academic level web application knowledge
Delivery method:
Instructed lead, hands-on exercises backed with assignments and mini project

CODE INTEL
ACADEMY

Detailed course contents:

Module 1:Overview and Architecture

- **What is Node**
- **Node Architecture**
- **How Node Works**
- **Installing Node**
- **Your First Node Program**
- **Event Loop**

Module 2:Node Module System

- **Introduction**
- **Global Object**
- **Modules**
- **Creating a Module**
- **Loading a Module**
- **Module Wrapper Function**
- **Path Module**
- **OS Module**
- **File System Module**
- **Events Module**
- **Event Arguments**
- **Extending Event Emitter**
- **HTTP Module**

Module 3:Node Package Manager (NPM)

Node Package Manager (NPM)
- **Introduction**
- **Package.json**
- **Installing a Node Package**
- **Using a Package**
- **Package Dependencies**
- **NPM Packages and Source Control**
- **Semantic Versioning**
- **Listing the Installed Packages**
- **Viewing Registry Info for a Package**
- **Installing a Specific Version of a Package**
- **Updating Local Packages**
- **DevDependencies**
- **Uninstalling a Package**
- **Working with Global Packages**
- **Publishing a Package**
- **Updating a Published Package**