

1. Javascript, React.js (Frontend), Node.js + MongoDB - 85 Hours

2. Data Engineering, Cloud, and AI Basics Training Program -270 Hours

Duration: 355 Hours (Approx. 24 Weeks)

Per Day Training- 3 hours and 5 days in a week

Target Audience Level- Freshers

Course Summary-

1. **Javascript, React.js (Frontend), Node.js + MongoDB -85 hours**
 - Javascript – 20 Hours
 - React JS - 30 Hours
 - Node JS – 20 Hours
 - Mogo DB – 15 Hours
2. **Data Engineering, Cloud, and AI Basics Training Program (240 Hours)**
 - Prerequisite Module – 32 Hours
 - Data Engineering – 70 Hours
 - Cloud Computing – 60 Hours
 - AI Basics – 78 Hours
3. **Capstone Project – 30 Hours**

JavaScript + React.js (Frontend) + Node.js + MongoDB (Full-Stack)

Module 1: JavaScript – 20 Hours

Introduction to JavaScript (1 Hour)

- What is JavaScript?
- How JavaScript works in the browser
- Linking JS to HTML (internal vs external)
- Console & basic debugging

Variables & Data Types (2 Hours)

- var, let, and const usage
- Primitive data types: String, Number, Boolean, Null, Undefined
- Type conversion and coercion

Operators & Conditionals (2 Hours)

- Arithmetic, assignment, comparison, logical operators
- if, else, else if, switch
- Ternary operator

Loops & Iteration (2 Hours)

- for, while, do...while loops
- Looping over arrays with for and for...of
- break and continue

Functions (3 Hours)

- Declaring and calling functions
- Function parameters and return values
- Arrow functions `(()=>{})`
- Scope and variable shadowing

Arrays (2 Hours)

- Creating and accessing arrays
- Common methods: `push()`, `pop()`, `shift()`, `unshift()`, `splice()`, `slice()`
- Iterating with for, `forEach()`, `map()`

Objects (2 Hours)

- Object creation and property access
- Dot vs bracket notation
- Nested objects
- Object methods (`Object.keys`, `Object.values`)

DOM Manipulation (2–3 Hours)

- `document.getElementById`, `querySelector`
- Changing content & styles dynamically
- Event handling (click, submit, keyup, etc.)

Asynchronous JavaScript (3 Hours)

- Introduction to callbacks
 - Promises and .then()/catch()
 - async/await
 - Basic example with fetch() API
- *****

Prerequisite Module: Core Concepts and Foundations (32 Hours)

1. Basic Programming Fundamentals (12 Hours)

- Basic for Programming Language, PLT, OOPS concepts

2. Data Structures and Algorithms (15 Hours)

- Arrays, Lists, and Dictionaries
- Searching and Sorting Algorithms
- Time and Space Complexity

4. Introduction to Linux and Git (5 Hours)

- Version Control Using Git
 - Git Init, Commit, Push, and Pull
 - Branching and Merging
- *****

Module 2: React.js - 30 Hours

Introduction to React

- What is React?
- Setting up Development Environment
- React vs Vanilla JS

JSX and Components

- JSX Syntax and Expressions
- Functional vs Class Components
- Component Props and Composition

State and Lifecycle

- useState and useEffect Hooks
- Component Lifecycle Methods
- Conditional Rendering

Events and Forms

- Event Handling in React
- Form Elements and Form Submission
- Controlled vs Uncontrolled Components

Routing and Navigation

- React Router Basics
- Nested Routes and Navigation

Working with APIs

- Fetch and Axios
- Displaying Fetched Data
- Error Handling

Styling in React

- CSS Modules and Inline Styling
- Styled Components (Optional)

Authentication Basics

- Authentication Basics Username and Password:
- **Traditional method using forms to collect credentials and server-side verification.**
- **Social Login (OAuth):**
 - Allowing users to authenticate through third-party providers like Google or Facebook.
- **Multi-Factor Authentication (MFA):**
 - Adding an extra layer of security by requiring multiple verification factors.
- **Token-Based Authentication (JWT):**
 - Using tokens to verify user identity after initial login, enabling stateless authentication.

Module 3: Node.js - 20 Hours

Introduction to Node.js

- Node.js Architecture
- npm and Package.json
- Core Modules: fs, path, http

RESTful APIs

- GET, POST, PUT, DELETE Methods
- Handling JSON Data
- Error Handling

Integration with MongoDB

- Connecting Node with MongoDB
- Performing CRUD Operations

Module 4: MongoDB (15 Hours)

Relational Databases (5 hours)

- SQL Basics (SELECT, INSERT, UPDATE, DELETE)
- Joins, Subqueries, and Indexing

Introduction to MongoDB (1 Hour)

- What is NoSQL and why MongoDB?
- Features of MongoDB
- Comparison: MongoDB vs. Relational Databases
- Use cases and real-world applications

MongoDB Installation & Setup (1 Hour)

- Installing MongoDB on Windows/Mac/Linux
- MongoDB Compass and MongoDB Atlas overview
- Basic CLI tools (mongod, mongo, etc.)
- Connecting to a MongoDB server

Understanding MongoDB Data Model (1 Hour)

- Documents, Collections, and Databases
- BSON vs JSON
- Schema-less design
- Data types supported by MongoDB

CRUD Operations (2 Hours)

► Create

- insertOne() and insertMany()

► Read

- find(), findOne(), projection
- Filtering with comparison operators

► Update

- updateOne(), updateMany(), \$set, \$inc

► Delete

- deleteOne(), deleteMany()

Hands-on: Inserting and querying sample student data

Indexing & Sorting (1 Hour)

- Why indexes matter
- Creating and dropping indexes
- Sorting query results
- Index performance basics

Aggregation Framework (2 Hours)

- What is aggregation?
- \$match, \$group, \$project, \$sort
- Real-world aggregation examples
- Hands-on: Building reports using aggregation

Data Engineering, Cloud, and AI Basics

Module 1: Fundamentals of Data Engineering *70 Hours*

1. Basic Programming Fundamentals (*20 Hours*)

- Introduction to Programming (Python)
- Data Types and Variables
- Control Structures (Loops, Conditionals)

2. Introduction to Data Engineering – *8 Hours*

- What is Data Engineering?
- Data Lifecycle and Data Types
- Data Sources and Data Formats

3. Data Processing and Storage -*16 Hours*

- ETL Concepts (Extract, Transform, Load)
- Basics of Data Warehousing
- Data Storage (SQL, NoSQL)

4. Tools and Platforms (-*16 Hours*)

- Introduction to Apache Hadoop and PySpark
- Data Cleaning and Transformation Using Pandas

5. Hands-On Practice – *10 Hours*

- Create a Basic ETL Pipeline
- Process and Store Data Using Spark

Module 2: Cloud Computing Basics -*60 Hours*

1. AWS Cloud Concepts -*10 Hours*

- Understanding Cloud Computing & AWS Ecosystem
- AWS Cloud Deployment Models: Public, Private (via VPC), and Hybrid
- AWS Global Infrastructure: Regions, Availability Zones, Edge Locations

2. AWS Core Services and Deployment -*25 Hours*

- Compute:
 - ECS : What is container, Dockerfile and a way to build a Docker image configuration and deployment of containers.
 - Amazon EC2: Launch, configure, and manage instances
 - AWS Lambda:
 - Basic Lambda Structure
 - A Lambda function consists of:
 - Handler: The main entry point.
 - Event: The input (e.g., API Gateway request, S3 event).
 - Context: Metadata -timeout, memory limit
 - Lambda Function Code (Python)-Example

- Storage:
 - Amazon S3: Bucket creation, object lifecycle, versioning
 - EBS and EFS Overview
- Deployment Tools:
 - AWS Elastic Beanstalk (basic application deployment)
 - Auto Scaling Groups (ASG) and Launch Templates
 - Elastic Load Balancer (ELB): Concepts and setup

3. AWS Security Overview -10 Hours

- Security Overview:
 - AWS Identity and Access Management (IAM): Users, Roles, Policies
 - AWS Shared Responsibility Model
 - Security Groups & Network ACLs
- Monitoring & Logging:
 - Amazon CloudWatch: Metrics, Alarms, Dashboards
 - AWS CloudTrail: Tracking API calls

4. Hands-On Projects and Practice (15 Hours)-

Real-world guided lab exercises on AWS Console and CLI

- Launch and manage an EC2 instance
- Set up an S3 bucket with access policies
- Deploy a basic Node.js web app using Elastic Beanstalk
- Trigger a Lambda function with S3 event
- Set up CloudWatch alarms to monitor EC2 performance
- Use Cost Explorer to analyze usage and forecast expenses
- ECS Fargate - running containers without managing servers

Module 3: AI and Machine Learning Basics (78 Hours)

1. Introduction to AI and ML (15 Hours)

- Data Statistics
- ML vs AI vs Deep Learning
- Types of Learning (Supervised, Unsupervised, Reinforcement)

2. Machine Learning Models (20 Hours)

- Regression and Classification Basics
- Decision Trees and k-NN
- Model Evaluation and Tuning

3. AI Tools and Libraries (20 Hours)

- TensorFlow and PyTorch Basics
- Pretrained Models
- AI Model Deployment on Cloud

Module 1: Introduction to Generative AI (2 Hours)

- **What is Generative AI?**
 - Traditional AI vs. Generative AI
 - How it works: Large Language Models (LLMs)
- **Popular GenAI Tools**
 - ChatGPT, DALL·E, Midjourney, Bard, Claude
- **Use Cases with Live Demos**
 - Text generation (emails, reports)
 - Image creation (text-to-image)
 - Code generation (basic Python snippet)
- **Ethical Use & Limitations**
 - Bias, hallucinations, responsible AI usage

Module 2: Prompt Engineering Basics (2 Hours)

- **What is Prompt Engineering?**
 - Why prompt clarity matters
 - Types of prompts: Instructional, contextual, role-based
- **Effective Prompt Techniques**
 - Zero-shot vs. few-shot prompting
 - Examples: Writing emails, summarizing text, writing SQL
- **Hands-on Activity**
 - Group-based prompt experiments with ChatGPT or Bard

Module 3: Introduction to GitHub Copilot (2 Hours)

- **What is GitHub Copilot?**
 - Overview, purpose, and how it helps developers
- **Using Copilot in VS Code**
 - Installation & setup
 - Writing code with natural language prompts
- **Live Demo**
 - Auto-generating a basic function
 - Refactoring existing code
- **Use Cases for Freshers**
 - Learning to code faster
 - Getting help with debugging or documentation

Module 4: Overview of Agentic AI (2 Hours)

- **What is Agentic AI?**
 - Difference between a tool and an AI “agent”
 - Concepts: Autonomy, decision-making, task chaining
- **Examples of Agentic AI**
 - AutoGPT, LangChain, ReAct, BabyAGI
- **Real-World Applications**
 - Business automation, customer service bots, task runners
- **Future of Work with Agents**
 - How roles may evolve
 - Responsible use and monitoring

5. Hands-On Practice (*15 Hours*)

- Build and Train a Simple ML Model
- Deploy an AI Model on Cloud

Module 4: Capstone Project (30 Hours)-6 Capstone Projects (for Each Topics)

1. Student Management System

Objective: Manage students, teachers, classes, attendance, and results.

Frontend (React):

- Student & admin dashboards
- Forms for student entry, marks, and attendance
- Charts for result visualization (use Chart.js)

Backend (Node.js + Express + MongoDB):

- User authentication (JWT) for admins and students
- CRUD APIs for students, attendance, results
- Role-based access

2. Real-Time Chat Application

Objective: Enable users to chat 1-on-1 or in groups.

Frontend (React):

- Chat UI with contact list
- Typing indicator, message timestamps
- Responsive design

Backend (Node.js + Express + Socket.io):

- Real-time chat with WebSockets
- MongoDB to store message history
- User management & authentication

3. Online Voting System

Objective: Secure platform for online elections or polls.

Frontend (React):

- Voter dashboard, list of polls
- Voting interface with live results
- Admin panel to create/manage polls

Backend (Node.js + Express + MongoDB):

- Voter authentication
- One-time voting per user
- Secure data storage and poll analytics

4. Healthcare Appointment System

Objective: Patients can book appointments with doctors online.

Frontend (React):

- Calendar-based booking interface

- Doctor & patient dashboards
- Notifications/reminders UI

Backend (Node.js + Express + MongoDB):

- Booking, scheduling, and cancellation APIs
- Role-based access (doctor, patient, admin)
- Email notifications using NodeMailer

5. Inventory Management System

Objective: For small businesses to track inventory and stock levels.

Frontend (React):

- Dashboard showing low stock alerts
- Product listing and transaction history
- Add/edit/delete inventory UI

Backend (Node.js + Express + MongoDB):

- Inventory CRUD APIs
- Authentication for managers/employees
- Data exports (CSV/PDF)

6. Social Media Mini Platform

Objective: Create a mini social media site with posts, likes, and comments.

Frontend (React):

- News feed, post creation, and profile views
- Like/comment interactions
- Notifications and follow system

Backend (Node.js + Express + MongoDB):

- User authentication and session management
- Media upload (Multer or Cloudinary)
- API for posts, comments, followers

Delivery Methods:

- **Instructor-led Sessions** (Online)
- **Interactive Group Discussions & Peer Reviews**
- **Practical based Learning** with Real-World Scenarios
- **Self-Paced Learning Modules** for Reinforcement
- **Quizzes & Assessments** to Track Progress
- The training will be delivered using **LMS**.
- **LMS access** will be given to all Learners

Assessments-

- Pre Assessment- Just to check the knowledge of the participants.
- Mid Assessment/Module Assessment/Monthly Assessment
- Post Assessment
- Capstone Project Evaluation

Certificate-

- Completion Certificate from TeamLease Digital