



Online/Offline | 6 Months

MERN Stack Development



Learning Collaboration



Our Hiring Partners

cost (2) action

Triazine[®]
software
Committed to Deliver

birlasoft

enveu

ADDVERB

AR Software
Pvt. Ltd.

JELLYFISH
TECHNOLOGIES

TRACKO BIT

SERVOSYS

MOPTRA[™]
INFOTECH

Traviyo
Empowering Travel Professionals

Telecon Systems



Introduction of MERN Stack Development

MERN stack development is a way to build web applications using four main technologies: MongoDB for storing data, Express.js for handling requests, React for creating user interfaces, and Node.js for running the server. It's popular because everything is based on JavaScript, which makes it simpler to construct dynamic and quick websites.

What's Included in this Course

This comprehensive course is crafted by industry experts, enabling students to learn and master the following topics:

- Introduction to MongoDB
- Introduction to Express
- React
- Node.js
- HTML
- CSS(Cascading Style Sheet)
- JavaScript
- Bootstrap
- Express.js
- Basic Web Application Development

Objectives of the MERN Stack Development Course

- Learn the MERN stack components MongoDB, Express.js, React, and Node.js and how they work together to build web applications.
- Develop full-stack web applications that integrate front-end and back-end functionalities.
- Perform CRUD operations in MongoDB to create, read, update, and delete data.
- Create RESTful APIs using Express.js to facilitate communication between the front-end and back-end.
- Develop interactive user interfaces with React, focusing on creating dynamic and responsive designs.
- Manage application state effectively in React, ensuring smooth data flow and rendering.
- Deploy applications on hosting platforms like Heroku or Vercel to make them accessible online.
- Implement security features for user authentication and authorization, protecting user data.
- Follow coding best practices to write clean code and organize projects efficiently.
- Use Git and GitHub for version control, enabling collaboration and tracking changes in projects.



MERN Stack Developer Salary



Source: Scholarhat

Our MERN Stack Development Course Benefits

- Gain in-demand skills for today's job market.
- Learn from industry experts with practical experience.
- Hands-on projects to build a strong portfolio.
- Flexible learning schedule to fit your lifestyle.
- Comprehensive coverage from basics to advanced topics.
- Access to cutting-edge tools and technologies.
- Networking opportunities with peers and professionals.
- Career guidance and job placement assistance.
- Personalized mentoring and support.
- Industry-recognized certification upon completion.
- Hands-on experience with real projects using the MERN stack.
- Stay updated with the latest trends in MERN Stack.



HTML Overview

- Getting Started
- Structure
- Color
- Working with Text
- Working with Links
- Working with Images
- Working with Multimedia
- Creating Lists
- Using Tables
- Developing Frames
- Employing Forms
- Positioning Page Elements

CSS (Cascading Style Sheet) Overview

- Benefits of CSS
- CSS Versions History
- CSS Syntax
- External Style Sheet using < link >
- Multiple Style Sheets
- Value Lengths and Percentages

CSS2-Selectors

- ID Selectors
- Class Selectors
- Grouping Selectors
- Universal Selectors
- Descendant / Child Selectors

- Attribute Selectors
- CSS - Pseudo Classes

CSS-Color Background Cursor

- Background-image
- Background-repeat
- Background-position
- CSS Cursor

CSS-Text Fonts

- Color
- Background-color
- Text-decoration
- Text-align
- Vertical-align
- Text-indent
- Text-transform
- White Space
- Letter-spacing
- Word-spacing
- Line-height
- Font-family
- Font-size
- Font-style
- Font-variant
- Font-weight



CSS-Lists Tables

- List-style-type
- List-style-position
- List-style-image
- List-style
- CSS Tables
- Border
- Width & Height
- Text-align
- Vertical-align
- Padding
- Color

CSS-Box Model

- Borders & Outlines
- Margin & Padding
- Height and Width
- CSS Dimensions

CSS-Display Positioning

- CSS Visibility
- CSS Display
- CSS Scrollbars
- CSS Positioning
- Static Positioning
- Fixed Positioning
- Relative Positioning
- Absolute Positioning
- CSS Layers with Z-Index

CSS Floats

- The Float Property
- The Clear Property
- The Clear Fix Hack

JavaScript Overview

- What is JavaScript?
- Javascript History
- How to Run JavaScript?
- Tools You Need
- A Simple JavaScript Program
- Hello World! Example
- JavaScript Variable
- JavaScript Array Methods
- JavaScript Loops
- Conditional Statement
- JavaScript Function
- JavaScript Cookies
- Object Oriented JavaScript (OOJS)
- JavaScript DOM(Document Object Model)

Bootstrap Overview

- What is Bootstrap
- CDN (Content Delivery Network)
- How to Use?

React Overview

- What is React?
- Motivation for Creating React
- A React JavaScript Example
- One-Way Data Flow
- JSX
- A JSX Example
- The Virtual (Mock) DOM



- Only Sub-components that Actually Change are Re-Rendered
- React Libraries

ES6 in a Nutshell

- What is ES6?
- ES6 Features
- Using ES6
- Transpiling
- Major Syntax Changes
- let and const
- Variable Scope
- Shadowing Variables
- Arrow Functions
- Arrow Functions As Parameters
- Using 'this' Within Arrow Functions
- Template Literals
- Spread Operator
- ES6 Classes
- Declaring Classes
- Declaring Instance Methods
- Accessor Methods
- Static Methods
- Inheritance With Classes

Basic Components and JSX

- What is JSX?
- JSX Transpilation to React Code Example
- Running the Transpiled Code
- Babel
- The Babel Runtime JavaScript Library
- Script Import Skeleton Code
- Playing Around in CodePen
- React Components
- Ways to Create UI Components

- Creating a Functional Component Example
- Component Names Must Be Capitalized
- Creating a React Class-Based Component in ES5
- The render Method
- Creating a UI Component Using ES6 Class Notation
- Using ES6 Classes with React
- Which UI Component Creation Syntax Should I Use?
- Components vs Elements
- Elements Are Immutable
- Properties
- Property Naming Convention
- Properties Default to 'True'
- Spread Attributes (an ES6 Feature)
- Expressions
- Fragments

React Functional Component Concepts

- Functional Components
- Nesting JSX Elements
- Example of JSX Nesting
- Comments in JSX Code
- Setting CSS Styles Using Classes
- Setting CSS Styles Directly
- JSX Escapes Values
- Working with Lists of Items
- Keys in Lists
- Example List With Key



- Container vs. Presentational Components
- State
- Types of State Data
- State Hierarchy
- Lifting State Up
- Props vs. State
- Pass Down a Function
- Immutability
- Immutability – Why?
- Virtual DOM and State
- Setting State
- Updating Input Fields
- Passing Props to Components
- Passing Functions to Components
- Event Handling
- Event Handler Example
- Event Binding - Do's
- Event Binding – Don'ts
- Passing Parameters to Event Handlers
- Component Life-cycle
- Life-cycle in Functional Components
- App Development Workflow – 1/3
- App Development Workflow – 2/3
- App Development Workflow – 3/3

React Components with ES6 Classes

- Classes in ES6
- Functional Components
- Extending React.Component
- The render() Method
- State
- Props
- DefaultProps

- PropTypes
- Component Lifecycle
- Component Life-cycle: Overview
- Component Life-cycle – Render Phase
- Component Life-cycle – Commit Phase
- Component Life-cycle – Unmounting
- Constructor() Example
- ComponentDidMount() Example
- SetState (newStateValue)

React Router

- Routing and Navigation
- React-router
- Creating a React-router Based Project
- A Basic Routed Component
- Router vs. BrowserRouter
- The Route Component
- <Switch>
- Redirect Route
- Navigating with <Link>
- Navigating with <NavLink>
- Route Parameters
- Retrieving Route Parameters
- QueryString Parameters
- Using Router with Redux

State Management for React

- React State Basics – Props and State
- Props
- State in Class Based Components
- Managing State with Hooks in Functional Components
- The Problem with Props and State
- Redux State Library
- Redux Advantages





- Redux Disadvantages
- Basic Rules for State Management
- Types of State
- Data State
- Communication State
- Control State
- Session State
- Location State
- Location State Side Effects

Using React Hooks

- Functional Component Short Comings
- Hooks Overview
- Hook Rules
- React Linter Example
- Functional Component Props
- The use State Hook
- Functional Component Using the useState Hook
- useState with Multiple Variables
- useState can also be used with Objects
- The useEffect Hook
- Use Effect Hook Example
- Using useEffect Hook to Load Data
- Restricting when use Effect is Called
- The use Context Hook
- Additional Hooks
- The use Reducer Hook
- An Example Reducer Function
- Calling and Using useReducer
- The use Memo Hook
- Use Memo Example
- The useCallback Hook
- useCallback Example
- The use Ref Hook

- Using useRef to Hold Values
- The useImperativeHandle Hook
- UseImperativeHandle Hook Example
- The useLayoutEffect Hook
- Summary

Introduction to MongoDB

- MongoDB
- MongoDB Features
- MongoDB on the Web
- Positioning of MongoDB
- MongoDB Applications
- MongoDB Data Model
- MongoDB Limitations
- MongoDB Use Cases
- MongoDB Query Language (QL)
- The CRUD Operations
- The Find Method
- The FindOne Method
- A MongoDB Query Language (QL) Example
- Inserts
- MongoDB vs Apache CouchDB
- Summary

Working with Data in MongoDB

- Reading Data in MongoDB
- The Query Interface
- Query Syntax is Driver-specific
- Projections
- Query and Projection Operators
- MongoDB Query to SQL Select Comparison
- Cursors
- Cursor Expiration





- Writing Data in MongoDB
- An Insert Operation Example
- The Update Operation
- Update Operation Options
- An Update Operation Example
- A Remove Operation Example
- Limiting Return Data
- Data Sorting
- Aggregating Data
- Aggregation Stages
- Accumulators
- An Example of an Aggregation Pipe-line
- Map-reduce

Introduction to Node.js

- What Is Node.js?
- Applications of Node.js
- Installing Node.js and NPM
- "Hello, Node World!"
- How It Works
- Node.js is Built on JavaScript: Benefits
- Traditional Server-side I/O Model
- Disadvantages of the Traditional Approach
- Event-Driven, Non-Blocking I/O
- Concurrency
- Using Node Package Manager (NPM)
- The Express Server Framework

Basic Web Application Development

- Introduction to the HTTPS Module
- The Request Handler Callback Function
- The Server Object
- Example Use of Server Object

- The Request Object
- The Response Object
- Parsing Request Body
- Serving Static Files
- The HTTPS Client API
- Making POST/PUT/etc. Requests

Events in Node JS

- Event Driven Programming
- Event Driven Programming (Contd.)
- EventEmitter
- EventEmitter Class
- EventEmitter Class – Inheritance
- The Event Loop and Event Handler
- Phases Overview
- Event Handlers
- Example (Using EventEmitter as an Object)
- Example (Inheriting from EventEmitter)
- EventEmitter Functions
- Issue with 'this' Keyword in Callback Functions
- Handling this Problem
- Controlling Event Callbacks in the Event Loop

Asynchronous Programming with Callbacks

- Synchronous and Asynchronous
- Callbacks
- Creating a Callback Function
- Calling The Callback Function
- Callback - Another Example
- Issue with 'this' Keyword in Callback Functions



- Handling this Problem
- Handling this Problem – Method 1 (Storing in Another Variable)
- Handling this Problem – Method 2 (Using Bind Function)
- Handling this Problem – Method 3 (Using ES6 Arrow Functions)
- Error Handling without Callback
- Error Handling with Callback
- Asynchronous Callback
- SetImmediate() and NextTick()
- API Example

Introduction to Express

- Introduction to Express
- Basic Routing Example
- Defining Routing Rules
- Route Path
- The Response Object
- Supplying URL Parameters
- Ordering of Routes
- Defining Catch All Route
- Full Example Web Service

Express Middleware

- Introduction to Express Middleware
- Writing a Middleware Function
- Binding to a Path
- Order of Execution
- Raising Error
- Handling Error
- Serving Static Files
- Handling POST Request Body
- Enable Response Compression

Building React Apps with Redux

- Redux
- Redux Terminology
- Redux Principles
- Redux: Actions
- Redux Action Types
- Action Creators
- Dispatching Actions
- Data Flow Basics
- Redux Reducers
- Pure Functions
- Reducer Example
- Returning Default State
- Creating a Development Environment
 - with Create-react-app
- Using Redux with React
 - Initializing the Store
 - Immutability
 - Benefits of Immutable State
 - Mutability of Standard types
 - Copying Objects in JavaScript
 - Copying Arrays in JavaScript
 - One Store - Multiple Reducers
 - Combining Reducers
- Components and Redux
 - The React-Redux Package
 - Wrapping App with Provider
 - mapStateToProps
 - mapDispatchToProps
 - Using Mapped Properties and Methods
 - Wrapping Components with Connect



Accessing MongoDB from Node.js

- Getting Started
- The Connection URL
- Obtaining a Collection
- Inserting Documents
- Updating a Document
- Querying for Documents
- Deleting a Document
- Connection Pooling

Indexes in MongoDB

- Indexes in MongoDB
- Creating an Index
- Listing Existing Indexes
- The Sort Order
- Using the Explain() Method
- Main Index Types
- The _id Field
- Single and Compound Indexes
- Multikey Indexes
- Other Index Types
- Index Properties
- Examples of Creating Indexes with
- Index Properties



**Join One of the Best Institutes in India
Helping Students Embrace Bright Career**
(Online and Offline Classes by Industry Experts)



22+

Years Industry
Experts



60000+

Professionals
Trained



100%

Placement
Assistance



24x7

Learners
Support





Talib Khan
Software Engineer



"CodeSquadz is a great place to learn and master your IT skills with updated study materials and training under the guidance of industry experts."



Alok Kumar Chaudhary
Software Developer



"It was a great experience to turn from a student to an IT professional. This all started when I joined the 6 months internship program under the guidance of Manish Bhatia sir, an expert of Java who is well-versed."



Saurav Kumar
Software Developer



"I joined the 6-month JAVA internship program from Manish Bhatia sir at CodeSquadz. Where I got an opportunity to learn all concepts of JAVA using the practical learning approach."



Ravi Prasad Panjiyar
Software Developer



"I joined the 6 months JAVA internship program at CodeSquadz where my mentor was Manish Bhatia sir who taught me how to level up the IT skills with practical training."

Our Recently Placed Students



Bhoopendra

Techcompiler
Software Engineer



Vishal Kumar

Krinvi Technologies
Full Stack Developer



Jyoti Sharma

MERN Developer
Cost 2 Action



Saurabh Kumar

Adverb Technologies
Java Developer



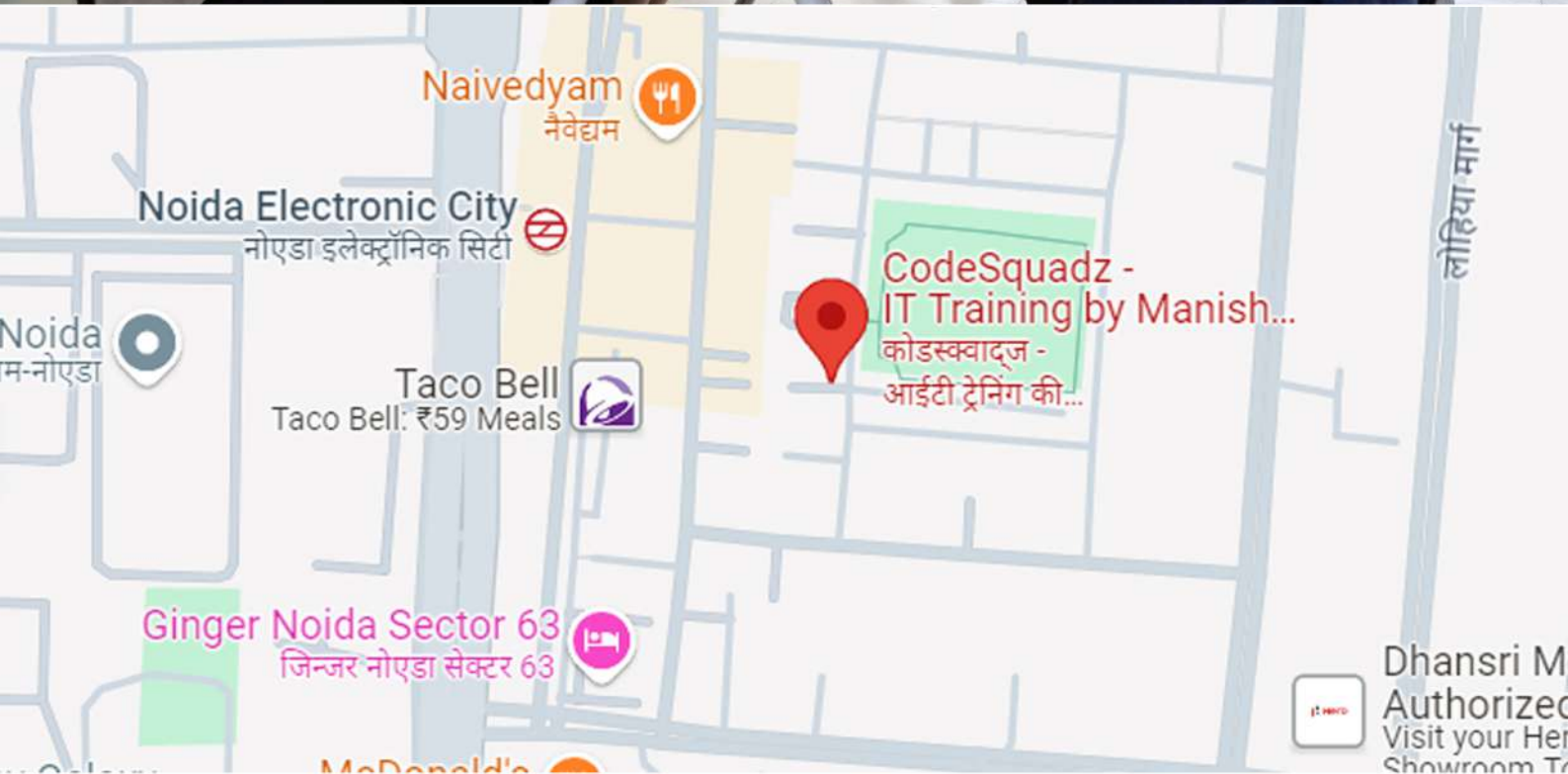
Vicky Mehta

Krinvi Technologies
Full Stack Developer



Suraj Brajesh

E2x Infotech
Software Developer



Address

H-65, H Block, Sector 63,
Noida, Uttar Pradesh, 201301

Connect Us

☎ 78-79-33-22-11

✉ enquiry@codesquadz.com

🌐 www.codesquadz.com



THANK YOU!