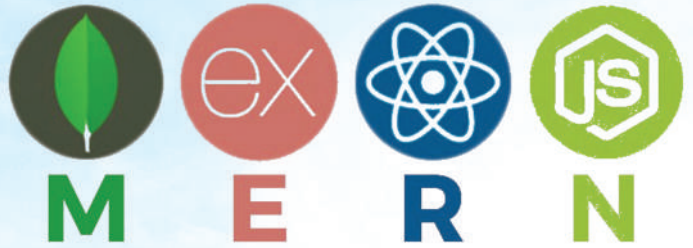


CELEBRATING
13 YEARS

QualityThought®



Full Stack Developer



The technology stack is a set of frameworks and tools used to develop a software product. This set of frameworks and tools are very specifically chosen to work together in creating a well-functioning software.

The MERN Stack is a collection of technologies used to develop web applications. It was developed by Facebook back in 2013. MERN is an acronym for the four technologies used: MongoDB, Express.JS, React, and Node. This collection of techs aims to make the development process easier and smoother.

Express and Node make up the middle (application) tier. Express.js is a server-side web framework, and Node.js is the popular and powerful JavaScript server platform. Regardless of which variant you choose, ME(RVA)N is the ideal approach to working with JavaScript and JSON, all the way through.

The MERN architecture allows you to easily construct a three-tier architecture (front end, back end, database) entirely using JavaScript and JSON.

Introduction to Node.JS

- ⇒ What is Node.js?
- ⇒ Why Node.js?
- ⇒ Node Architecture
- ⇒ How Node Works?
- ⇒ Your First Node Program

Node Module System

- ⇒ Introduction
- ⇒ Global object
- ⇒ Modules
- ⇒ Creating & Loading a Module
- ⇒ Module Wrapper Function
- ⇒ Path Module
- ⇒ OS Module
- ⇒ File System Module
- ⇒ Events Module
- ⇒ Event Arguments
- ⇒ Extending Event Emitter
- ⇒ Http Module

Node Package Manager

- ⇒ Introduction
- ⇒ Package.json
- ⇒ Installing a Node Package
- ⇒ Using a Package
- ⇒ Package dependencies
- ⇒ NPM packages and source control

- ⇒ Semantic versioning
- ⇒ Listing the installed packages
- ⇒ Viewing the registry info for a package
- ⇒ Installing a specific version of a package
- ⇒ Local and Global Packages
- ⇒ Updating local packages
- ⇒ Dev Dependencies
- ⇒ Uninstalling a package
- ⇒ Working with Global packages
- ⇒ Publishing a package
- ⇒ Updating a published package

Practices

- ⇒ Install Node.js
- ⇒ Run a simple Node.js Script
- ⇒ Create a Node Module and use it in your code
- ⇒ Search the NPM Repository to find a package and install it in your project

Asynchronous Programming

- ⇒ What is Asynchronous Programming in JavaScript?
- ⇒ JavaScript EventLoop
- ⇒ Callbacks
- ⇒ Callback Hell
- ⇒ Promises
- ⇒ Promise Chaining
- ⇒ Async Await

Practices

- ⇒ Demonstrate the use of call backs, promises, and async-await

FileSystem

- ⇒ Synchronous vs Asynchronous IO
- ⇒ File Constants
(`__dirname` and `__filename`)
- ⇒ Synchronous File Operations
- ⇒ Asynchronous File Operations
- ⇒ FileSystem Watcher

Practices

- ⇒ Write Node.js code to asynchronously perform file operations
- ⇒ Write Node.js code to log to the console if a file changes in a directory

Streams and Events

- ⇒ What are Streams in Node.js
- ⇒ Reading and Writing Streams
- ⇒ Understanding and using EventEmitter class
- ⇒ Emitting Events
- ⇒ Responding to events

Practices

- ⇒ Read data from a file using Input Stream and display on screen

HTTP

- ⇒ What is HTTP
- ⇒ How Does HTTP Work?
- ⇒ Standard HTTP response codes
- ⇒ Node HTTP Module
- ⇒ Processing Form Data
- ⇒ Processing Query Strings from URL
- ⇒ Sending Back response

Practices

- ⇒ Create a simple HTTP Server using Node

Express JS

- ⇒ What is ExpressJS?
- ⇒ Installing ExpressJS
- ⇒ Creating a simple HTTPServer using ExpressJS
- ⇒ Routing in Express
- ⇒ Templating in Express
- ⇒ Templating using JADE
- ⇒ Middlewares

Practices

- ⇒ Create a simple Express JS Application with routes and templates

Express JS and Mongo DB

- ⇒ What is MongoDB
- ⇒ MongoDB Advantages
- ⇒ Documents
- ⇒ Collections
- ⇒ Mongoose ODM
- ⇒ Creating Schemas and Models using mongoose
- ⇒ CRUD Operations in Mongo DB using Mongoose and Express

Practices

- ⇒ Creating a simple Application using ExpressJS and Mongoose

Testing

- ⇒ What is Unit Testing?
- ⇒ Why Unit Testing?
- ⇒ Unit Testing Frameworks
- ⇒ What are Mocks and Stubs?
- ⇒ Writing and Running tests in Express

Practices

- ⇒ Write and run some unit tests to test a module

REST API using Express and Mongoose

- ⇒ What is a REST API
- ⇒ Why REST API?
- ⇒ Data Exchange Formats
- ⇒ JSON vs. XML
- ⇒ Monolith Architecture
- ⇒ Microservice Architecture

Practices

- ⇒ Build a small application using the REST API

Web Development Fundamentals

- ⇒ What is Web?
- ⇒ Web Server, Web Client
- ⇒ Website, Webpage
- ⇒ Request and Response
- ⇒ Http/Https
- ⇒ Languages and Tools of Web Development
- ⇒ How the Web Works
- ⇒ Inspecting HTTP Requests and Responses
- ⇒ HTML Basics
- ⇒ CSS Basics
- ⇒ Formatting Code
- ⇒ Inspecting Pages Using DevTools
- ⇒ Validating Web Pages

HTML & Html 5

- ⇒ Introduction
- ⇒ The Head Section
- ⇒ Text
- ⇒ Entities
- ⇒ Hyperlinks
- ⇒ Images
- ⇒ Video and Audio
- ⇒ Lists
- ⇒ Tables
- ⇒ Containers
- ⇒ Semantic Elements
- ⇒ Structuring a Web Page

Layout

- ⇒ Introduction
- ⇒ The Box Model
- ⇒ Sizing Elements
- ⇒ Overflowing
- ⇒ Measurement Units
- ⇒ Positioning
- ⇒ Floating Elements
- ⇒ Flexbox
- ⇒ Grid
- ⇒ Hiding Elements
- ⇒ Media Queries
- ⇒ Summary
- ⇒ Practices

Typography

- ⇒ Introduction
- ⇒ Styling Fonts
- ⇒ Embedding Web Fonts
- ⇒ Flash of Un-styled Text
- ⇒ Font Services
- ⇒ System Font Stack
- ⇒ Sizing Fonts
- ⇒ Vertical Spacing
- ⇒ Horizontal Spacing
- ⇒ Formatting Text
- ⇒ Practices

Bootstrap

- ⇒ Introduction
- ⇒ Bootstrap classes
- ⇒ Responsive grid system
- ⇒ Columns and rows
- ⇒ Components
- ⇒ Responsive navbars
- ⇒ Margins
- ⇒ Modals, dialogs
- ⇒ Buttons
- ⇒ Forms
- ⇒ List groups
- ⇒ Badges, pills
- ⇒ Cards
- ⇒ Tables
- ⇒ Alerts
- ⇒ Navigation options
- ⇒ Links

CELEBRATING
13 YEARS

QualityThought®



QualityThought

☎ 73373 44490, 74162 36806, 83676 48293

Quality Thought Infosystems India (P) Ltd.

#306, 308, Nilgiri Block, Ameerpet, Hyderabad-500016 | www.qualitythought.in | info@qualitythought.in