

MEAN STACK



MEAN Stack Developer Training

A complete 62 hours training program to master the field of Web Application development with the power of JavaScript Technologies.

What is MEAN-Stack?

MEAN stack stands for

M- MongoDB - An object oriented schema less database, largely different from a typical a relational database system like Oracle,mysql etc.. It is a object oriented database system a OODBMS if you will, meant for object oriented languages.

E- ExpressJS - A de-facto Model-View-Controller framework and a web-server provider for Node.JS applications.

A- Angular - Now a very much preferred User Interface framework.

N- Node.JS - A server side runtime running on V8 engine - An engine used in the Google Chrome browser, capable of understanding Javascript.

When these technologies come together, they form what is a called MEAN stack, a total knowledge of this stack allows you to create full fledged user-interactive applications, like in any other language eg: Java,.NET, php , etc. except for some subtle differences in thought and technology employed. Please note, the entire stack is open-source and free and they can be mixed with any other technology as we will see in the course.

HTML

- Explaining HTML5
- Explaining Bootstrap latest version 4
- Using various components to make UI
- Making complete pages
- Understanding the static nature

JavaScript

- What is Javascript and bit on history , current version of Javascript
- Design flaw in javascript
- Async pattern in Javascript, why is it async in the first place.
- Type less(no data type) javascript,var,
- Declaring functions anywhere
- Passing functions in functions
- Declaring functions in functions
- What is JSON—Very important to understand, as it is one of the pillars in the foundation for understanding Javascript
- Understanding functional Scope
- Knockout.js, data-binding and earlier angularjs, a bit history on jQuery
- Typescript, what is it and why does it exist and its predecessors

Angular JS

- Why did Angular happen in the first place, the problem statement solved by it
- Approach in Angular 1
- A rewrite of angular 1 to 2 , again react.js thought process being the predecessor here
- Introduction to component based thinking and srp

- Understanding features of Typescript, more on each feature as and when encountered in the project
- Setting up the Angular Application
- Understanding that Tooling comes with it to make our life easy to focus on bigger problems
- Using browser console
- Creating a project using the cli
- Creating other entities using the cli
- If created without the cli , where all the changes would have to be made and why
- Understanding components and creating them
- Creating custom components and how to use them
- Composing one custom component of other child components
- Making custom events and properties in custom components
- Data-binding in custom components
- Understanding View encapsulation
- Using local references in templates
- What are lifecycle hooks in Angular
- Seeing lifecycle hooks in action
- What is ngcontent and @Contentchild
- Understanding directives
- Creating a basic directive attribute
- ngClass and ngStyle
- Using Services and Dependency injection(recap)
- Reusability with services and SoCs and DI
- Building a custom data service
- Understanding page routing
- Why do we need a router

- Routing with Parameters
- Retrieving query parameters
- Setting up nested child routes
- Introduction to Guards
- Protecting routes with canActivate
- Understanding observables and why we need them
- How observables can create a memory leak if not handled correctly
- Understanding observable operators
- Handling Forms in Angular Apps
- Using Pipes in Angular 4
- Making object attribute based filters
- Making HTTP requests
- Deploying Angular Application
- Unit Tests in angular
- Analyzing Testing Setup
- Running Tests
- Using developer console to debug and other tools
- Using sessionStorage in browser

Node JS

- What is a server side runtime
- What is Node.js, and why node.js who made Node.js and when and who owns the code now, what is the current version of node.js
- What is the LTS schedule of Node.js
- Using the node.js console
- What is npm
- Introducing package.json file

- Meaning of attribs in package.json
- Initiating a simple project in node.js
- Every node.js project has an entry point
- Explaining how Node.js starts from within the application, rather than behave like a traditional server
- Starting a server using only node.js runtime
- Introducing express, a de-facto node.js web-container-for web based projects
- How to configure express in Node.js
- Using express.static for static assets
- Configuring routes and action handlers
- Explaining concept of request,response and session
- Making business logic classes
- Configuring database in node.js
- Making calls to dbfunctions
- Explaining concept of drivers and(Reading documentation and writing code)
- Getting connection to database(using blank functions..do not write queries)
- Sending response to browser
- Using Async and Underscore and Later.js
- Using Async and Underscore in code
- Coding style,patterns and best practices
- Debugging in nodeJS using node-inspector, remote and local
- Error handling using bunyan-error logging
- What are REST APIs v/s SOAP API , which is in trend now
- What are request methods and when to use what
- What is JWT and how to authenticate requests
- Using local storage on browsers
- What is TDD

- TDD in node.js
- Building a project using grunt/gulp
- One step build, fail on unit test fail
- Using a CI node.js
- Integrating everything into a CI
- StackOverflow Joel Spolsky 12 steps

Database/ MongoDB

- What is database and why do we need it
- What is RDBMS strategy and what is OODBMS strategy
- What is MongoDB or noSQL, install and pre-req
- Getting connection
- Making Create query
- Making Read query
- Making Update query
- Making Delete query
- Making aggregate,count,distinct,map-reduce queries
- Doing it via node.js application

Project Development

- Making an entire framework complete with frontend,backend and error logging

After Training Talks

- Angular 5 Material for UI
- Opbeat real time error logging
- Explaining Loopback.io

Contact

Asterix Solution

Vashi Branch Address: Shop No 7 & 8, Shivshankar Tower, Opposite Kadambari CHS, Sanpada East, Navi Mumbai-400 705 (Nearest Railway Station: Vashi)

Thane Branch Address: 201 2nd Floor, Rajhans Annexe, Opposite Gaondevi Bus Depot, Above Rajmal Lakhichand Jewellers, Thane West, Mumbai-400602

Contact: 9821681514 / 7715036251

Website: www.asterixsolution.com

Email: welisten@asterixsolution.com