MEAN Stack

1. **Introduction**

2. **Foundation**
    a. The Node.js framework
    b. Installing Node.js
    c. Using Node.js to execute scripts

3. **Node Projects**
    a. The Node Package Manager
    b. Creating a project
    c. The package.json configuration file
    d. Global vs. local package installation
    e. Automating tasks with Grunt.

4. **HTTP**
    a. The HTTP protocol
    b. Building an HTTP server
    c. Rendering a response
    d. Processing query strings
    e. Using Representational State Transfer
    f. Configuring TLS

5. **File System**
    a. Synchronous vs. asynchronous I/O
    b. Path and directory operations
    c. __dirname and __filename
    d. Asynchronous file reads and writes

6. **Buffers, Streams, and Events**
    a. Using buffers for binary data
    b. Flowing vs. non-flowing streams
    c. Streaming I/O from files and other sources
    d. Processing streams asynchronously
    e. Configuring event handlers

7. **Modules and Unit Testing**
    a. Modularization
    b. The CommonJS and RequireJS specifications
    c. Defining modules with exports
    d. Modules are singletons

Office 41-42/A, Second Floor Shreenath Plaza,
Dnyaneshwar Paduka Chowk, Pune, Maharashtra 411005
Mob : +91-8237077325

e. Creating a package
f. Module scope and construction
g. Unit testing frameworks
h. What to test and how to test it
i. Building unit tests with Jasmine

8. Express

a. The model-view-controller pattern
b. Building a front-end controller
c. Defining routes
d. Creating actions
e. Using REST
f. Reading POST data
g. Adding middleware

9. Data Sources

a. How Node.js connects to databases
b. RDBMS databases and NoSQL databases
c. Connecting to RDBMS and NoSQL databases
d. Performing CRUD operations
e. Building client requests to web services

10. What is MongoDB?

a. The current SQL/NoSQL landscape
b. Document-oriented vs. other types of storage
c. Mongo's featureset
d. Common use-cases
e. Introduction to JSON

11. Documents and Collections

a. Creating documents
b. Managing documents in collections
c. Iterating over documents

12. Simple Queries

a. Field equality tests
b. Operators available
c. Projections
d. Limiting results and paging

13. Simple Updates and Deletes

a. Field updates
b. Field insertions and removal
c. Document deletion

14. **More Complex Types of Queries**
    a. Existential field values
    b. Aggregations and groups
    c. Aggregations and groups in hierarchical data

15. **Updates and Arrays**
    a. Altering array field elements
    b. Insertion to array fields
    c. Removing from array fields

16. **Indexing 1**
    a. The primary index and the _id field
    b. Problems requiring an index
    c. Defining secondary indexes
    d. Compound indexes

17. **Indexing 2**
    a. Index selection
    b. Index hints
    c. Covering indexes
    d. Index storage size
    e. Indexes effect insertion and update speeds

18. **Mongo RESTful API**
    a. CRUD operations through REST
    b. Using Mongoose with Node.js

19. **MapReduce**
    a. Explanation of MapReduce
    b. Types of logic that can be expressed as MapReduce declarations
    c. Mapping documents
    d. Reducing values

20. **Mongo Security**
    a. Authorization and securing collections, documents
    b. The limits of Mongo's authorization scheme
    c. Authentication
    d. Mongo in the enterprise

21. **Mongo Replication and Sharding (optional)**
    a. Configuring replication
    b. Configuring sharding
    c. Accessing clustered data from client APIs

d. Latency and consistency in replicated and sharded Mongo

22. Introduction to AngularJS

a. What does AngularJS do for me?
b. Who controls AngularJS?
c. How can I get AngularJS?

23. Our first AngularJS application

a. A basic application
b. Using angular-seed
c. The pieces of the puzzle
    i. Two-way data binding
    ii. Directives
d. How it fits together
    i. How much of the page is an Angular application?
e. Model, View, Controller from the AngularJS Perspective

24. Single Page Applications

a. What do we mean by Single Page Application?
b. Creating Angular Modules
c. Using Angular's Routing Service
    i. Routing Basics
    ii. Accessing URL Data
    iii. Using the $location Service
d. Creating a Skeleton Single Page Application

25. Controllers

a. Where Controllers fit in, and what they do, from Angular’s perspective
b. Managing Scope
c. Setting up Behavior
d. Building a basic controller
e. A more advanced controller

26. Models

a. How to create a model
b. Explicit models
c. Implicit models

27. Views

a. Angular's take on the View: a little bit different
b. Tying a View to a Controller
c. Tying a View to a model

28. Expressions

a. Expressions are lightweight code snippets
b. Expression capabilities
c. Limitations
d. The border between expressions and $eval

29. Filters

a. Standard filters
b. Writing your own filter
c. Tying filters together

30. Scopes

a. What are scopes?
b. What do scopes provide?
c. Scope lifecycle
d. Scopes as glue between controller and view
e. Scope hierarchies
f. Scope and events

31. Angular Forms

a. Angular forms vs HTML forms
b. Angular form controls
c. Form events
d. The form controller
e. Form validation
    i. CSS classes for form data
f. Ajax, Data, and Angular
    i. High level interactions with servers
    ii. Low-level server interactions with $http
    iii. The deferred/promises API
    iv. Making RESTful Service calls with $resource
g. Directives
    i. Teaching HTML new tricks
    ii. Binding text and attributes
    iii. Directive processing lifecycle
    iv. A basic directive
    v. Directives and scopes
    vi. Creating reusable directives
    vii. Turning directives into components
h. Testing in Angular
    i. Unit testing
    ii. End-to-end testing
    iii. Angular User Interfaces

32. Overview of MEAN.IO/MEAN.JS (select one)

33. **Build a Web-Based CRUD Application using MongoDB, Node.js, Express and Angular.js**
    a. Application will include the following:
        i. Use of MEAN.IO/MEAN.JS
        ii. Data Storage
        iii. User Login
        iv. Server-Side Routes
        v. CSS/JS File Combining and Minimization with Grunt Tasks
        vi. DropBox-style Drag and Drop File Uploads
        vii. Web Sockets
        viii. js
        ix. Additional components as time allows and class interest indicates

34. **Angular 2 Overview and Migration Strategy**

35. **Conclusion**