A SEMINAR REPORT ON

MEAN STACK

SUBMITTED TO THE SAVITRIBAI PHULE PUNE UNIVERSITY

IN THE PARTIAL FULFILLMENT OF THE REQUIREMENTS

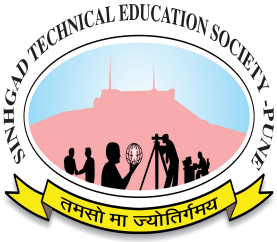
FOR THE AWARD OF THE DEGREE

OF

THIRD YEAR-COMPUTER ENGINEERING

BY

Zingade Shraddha Bhanudas — Exam No : T120364503



Sinhgad Institutes

DEPARTMENT OF COMPUTER ENGINEERING

STES'S SMT KASHIBAI NAVALE COLLEGE OF ENGINEERING

VADGAON BK,OF SINHGAD ROAD

PUNE 411041

MARCH 2015

Sinhgad Institutes

CERTIFICATE

This is to certify that the project report entitles

”MEAN STACK”

Submitted by

Zingade Shraddha Bhanudas Exam No : T120364503

This is to certify that Ms. Zingade Shraddha Bhanudas has successfully completed her seminar work on “MEAN STACK” at Smt. Kashibai Navale College Of Engineering,Pune in the partial fulfillment of the Graduate Degree course in B.E.(Computer Engineering) at the department of Computer Engineering, in the academic Year 2014-2015 Semester VI as prescribed by the Savitribai Phule Pune University.

Prof. M. R. Patil
Guide
.

Dr. P. N. Mahalle
Head of Department
Department of Computer Engineering

Dr.A.V.Deshpande
Principal

Place:Pune
Date:

Acknowledgement

I owe my gratitude and respect to every person who in smallest way has taken efforts for successful completion of the seminar. Firstly, I would like to thank Prof. M. R. Patil, my seminar guide, for not only helping me choose the topic precisely but also for guiding me at every step on the way. She has invested a lot of time and efforts and has always encouraged me to move forward. I owe all my success to her. I am also grateful to Dr. P. N. Mahalle for guiding and motivating us whenever in doubt. He always prompted us to give our best.

I also give my regards to my seminar practical guide, Mr. Raghib. Nasri, for giving his inputs and remarks. They were really valuable.

At last, I wish to thank all my colleagues and friends for providing help and support whwnever needed.

-Shraddha Zingade

Abstract

Traditional methods of creating web applications needed us to create a back-end server separately, a front-end separately and then establish a connection between them using Database Connectivity API. Also, the language and data format used at both the ends varies, making it a necessity for us to learn and understand their various structures and design framework. The whole process is extensive, tedious and time-consuming. Also, the the creation, deletion and updation of data is slow and needs a lot of work.

In order to overcome these drawbacks we use a stacks of four web technologies called - MEAN STACK, abbreviated as, MongoDB, Express.js, Angular.js, Node.js.

This seminar, thus focuses on using these four technologies together as a stack to create web applications which are easy to code, update as well as require less cost if compared to others. The reason it is easier to code is because it has - one language throughout, i.e JavaScript in both back-end and front-end, eradicating the hardwork of learning two or more formats for one application.

List of Figures

Contents

Chapter 1

INTRODUCTION

The MEAN (MongoDB, Express, AngularJS, Node.js) stack is a modern stack for building professional websites and web applications with open source software. It represents a major shift in architecture and mental models from relational databases to NoSQL and from server-side Model-View-Controller to *client-side, single-page applications.*

1.1 Description

MEAN stack refers to a collection of JavaScript based technologies used to develop web applications. MEAN is an acronym for MongoDB, ExpressJS, AngularJS and Node.js. From client to server to database, MEAN is full stack **JavaScript**

Node.js is a server side JavaScript execution environment. It helps in building highly scalable and concurrent web applications rapidly.

Express is lightweight framework used to build web applications in Node. It provides a number of robust features for building single and multi page web application.

MongoDB is a schemaless NoSQL database system. MongoDB saves data in binary JSON format which makes it easier to pass data between client and server as *the whole stack has JavaScript throughout*.

AngularJS is a JavaScript framework developed by Google. It provides some awesome features like the two-way data binding. Its a complete solution for rapid and awesome front end development.

1.2 Overview

In this chapter, we got a brief introduction about the four web technologies combined together in a stack and their descriptions.

In Chapter 2, we study the server environment i.e Node.js in detail, its functionality and its packet manager called NPM. It stresses on one important thing, which is, Single Threaded Approach.

In Chapter 3, we learn about the Express framework and use of its additional packages and features inside Node.js.

In Chapter 4, we understand the database used, MongoDB, its data format and the additional plugin it offers called Mongoose.

In Chapter 5, we study Angular.js which is the front-end framework used and its two unique features called Single Page Application and Two-Way Binding.

In Chapter 6, we learn the architecture of these four technologies as a whole stack.

In Chapter 7, we compare the other existing technologies and stacks and understand why MEAN stack is better compared to others.

In Chapter 8, we read about the existing user of MEAN stack and go through their wide range of applications.

Chapter 9 being the references, the study material is listed along with their authors.

Figure 1.1: Components Of Stack

The MEAN stack represents a thoroughly modern approach to web development: one in which a single language (JavaScript) runs on every tier of your application, from client to server to persistence.

Chapter 2

NODE.JS

Node.js is the foundation of stack i.e the platform on which the whole application is built. It has one language used throughout the stack which is the JavaScript language. It creates its own web server and builds web application on top of this environment. It has built-in HTTP server library, due to which we don't have to run a separate web server program eg: In LAMP stack we run a separate Apache server and then establish the connection.

Node.js is scalable, fast and makes efficient use of server resources. Due to less resources being used, it requires less cost.

2.1 NPM Manager

NPM is Node.js Packet Manager which is installed along with node.js. The function of this manager is to download packages to extend functionality of node.js. It has around

46,000 packages providing various features and functions. It is easy to create, acquire and modify these packages.

Example of packages :

1. colors $\longrightarrow$ $textcolor, backgroundcoloretc.$
2. mocha $\longrightarrow$ $testingJavaScriptframework.$

2.2 Single Threaded Approach

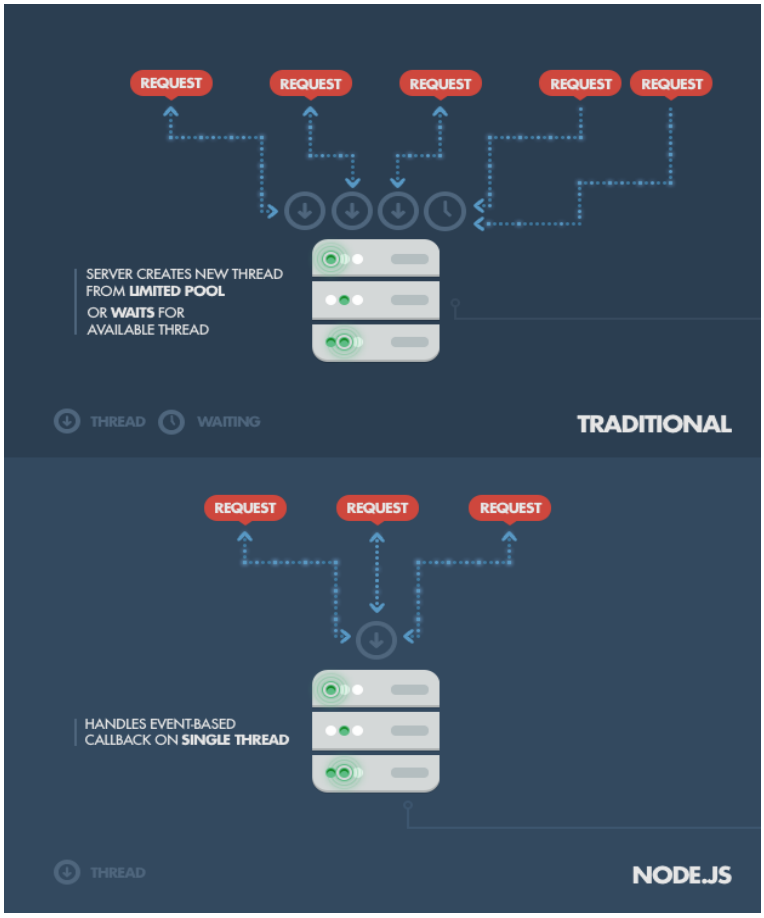


Figure 2.1: Single Page Application

In the traditional approach, most of the current mainstream web servers are multi-threaded, including Apache, in which every new visitor(or session)is given a separate thread and associated amount of RAM, often around 8MB. Due to this excess RAM being used, webservers are often overpowered and have so much RAM, even though for 90

Node.js server is single threaded and works differently to the multithreaded way. Rather than giving each visitor a unique thread and a separate silo of resources, every

visitor joins the same thread. The visitor and thread only interact when needed, when the visitor is requesting something or the thread is responding to a request. Duee to this the resources used and the cost required is reduced remarkably.

Chapter 3

EXPRESS

Express is a web application framework for Node.js. There are a few common tasks that need doing every time while creating a web applications. Express.js does these tasks in well tested and easier way.

3.1 Use and Advantage

3.1.1 Easing Server Setup

Express sets up a webserver to listen to incoming requests and return relevant responses. On top of this it also defines a directory structure making it easier to find the required webpage and send it to display. This can be done directly in Node.js, but Express does it for you.

3.1.2 Routing URL's to Response

It directs the incoming URL's to certain piece of code.Even if it is going to serve a static HTML page, read from a database or write to a database, the interface is simple and consistent. Also, it is easier to code and maintain if compared to node.js.

3.1.3 Supports Template Engine

Express provides support for a number of different templating engines that make it easier to build HTML pages in an intelligent way, using reusable components as well as data

from your application. Express complies these together and serves them to the browser as HTML.

3.1.4 Remembering Visitors with Session Support

Being single threaded, Node.js doesnt remember a visitor from one request to the next. It doesnt have a silo of RAM set-aside just every user.

On the other hand, Express comes with ability to use sessions, so that you can identify individual visitors through multiple requests and pages.

Chapter 4

MONGODB

MongoDB is a NoSQL document database where each row is a document and multiple documents together form a collection. A column defines what should be in the row where each row is a document, and this document both defines and holds the data itself. In MongoDB, data is stored in an unstructured way. It holds the data in key-value pair i.e describing the data(key) and defining data(value).

MongoDB stores document in BSON - Binary JSON format, where JSON is JavaScript object notation.

The following snippet shows a very simple sample MongoDB document :

”firstName” : ”Simon”,
”lastName” : ”Holmes”,
¡*d* : *ObjectId*(”52279*effc*62*ca*8*b*0*c*1000007”)

4.1 Mongoose

Mongoose is an object modeling package for node.js that allows us to add structure to the application. As well as modeling data, Mongoose adds an entire layer of features on top of MongoDB that are useful when building web apps. Mongoose makes it easier to

manage the connections to your MongoDB database, as well as making it easier to save data and read data. Well use all of this later.

Mongoose enables you to add data validation at the schema level, making sure that you only allow valid data to be saved in the database.

MongoDB is a great choice of database for most web applications, as it provides a balance between the speed of pure document databases and the power of relational databases. That the data is effectively stored in JSON makes it the perfect datastore for the MEAN stack.

The reason for using Mongoose in MEAN stack is for conversion of data format i.e BSON(data format of MongoDB) to JSON(data format used all over stack). Mongoose translates data in the database to JavaScript objects for use in your application.

Chapter 5

ANGULAR.JS

In simple terms AngularJS is a JavaScript framework for working with data directly in the front-end. The traditional way of doing things is to have all of the data processing and application logic on the server, which then passes HTML out to the browser. AngularJS enables you to move some or all of this processing and logic out to the browser, sometimes leaving the server just passing data from the database.

AngularJS makes use of MVC model.

$$\text{M} \longrightarrow responsible for maintaining data.$$
$$\text{V} \longrightarrow responsible for displaying required data.$$
$$\text{C} \longrightarrow controls interaction between model.$$

So when data is called at the front-end, the M-block provides the necessary data, the V-block looks after its display format whereas the C-block puts both the data together and controls their interraction.

The best part of Angular is that we don't need to split an application into parts according to webpages and then link them together. Angular.js does this for you by serving as the pipeline that connects these webpages.

5.1 Single Page Application

In real terms a Single Page Application runs everything inside the browser, and never does a full page reload. What this means is that all application logic, data processing, user flow

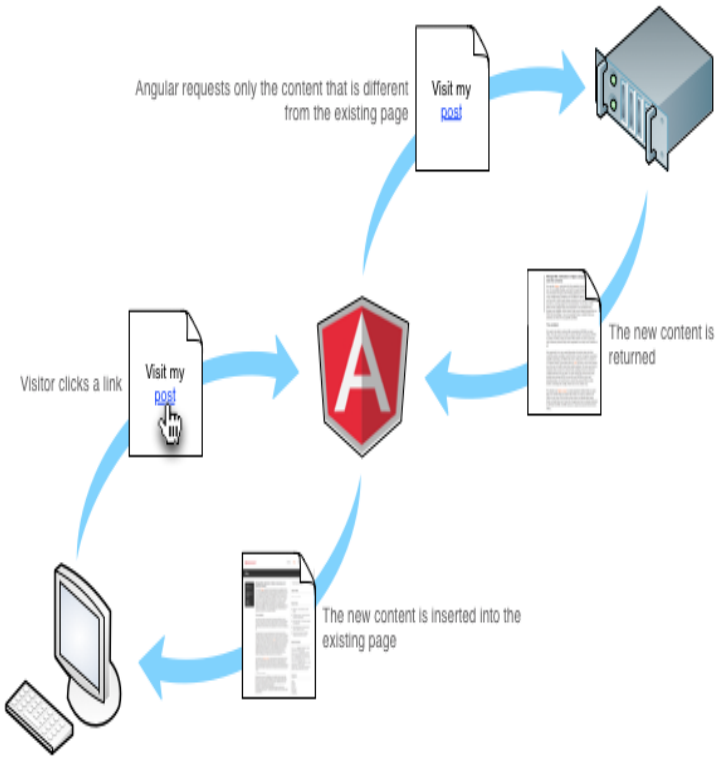


Figure 5.1: Working of Single Page Application

and template delivery can be managed in the browser. Example of this application is Gmail. Thats a Single Page Application. Different views get shown in the page, along with a whole variety of data sets, but the page itself never fully reloads.

This approach can really reduce the amount of resources you need on your server, as you are essentially crowd-sourcing the computational power. Each persons browser is doing the hard work, and your server is basically just serving up static files and data on request.

The user experience can also be greater when using this approach. Once the application is loaded there are fewer calls to be made to the server, reducing the potential of latency.

5.2 Two-Way Binding

This one-way model is the basis for most database-driven websites. In this model most of the hard work is done on the server, leaving the browser to just render HTML and run any JavaScript interactivity.

Two-way data binding is different. Firstly, the template and data are sent indepen-

ONE-WAY BINDING	TWO-WAY BINDING
Node.js gets data from MongoDB	Template and data sent separately.
Express.js uses template to compile it into HTML and sends it to browser.	View is live.
Hardwork by server, Browser only runs JavaScript.	Server only sends data.

Figure 5.2: Types of Binding

dently to the browser. The browser itself compiles the template into the view and the data into a model. The real difference is that the view is live. The view is bound to the model, so that if the model changes the view changes instantly. On top of this, if the view changes then the model also changes. This is the two-way binding, and is illustrated in Table 5.1

Chapter 6

ARCHITECTURE

MongoDB stores data in binary JSON, which through Mongoose is exposed as JSON. The Express framework sits on top of Node.js, where the code is all written in JavaScript. In the front-end we have AngularJS, which again is JavaScript.

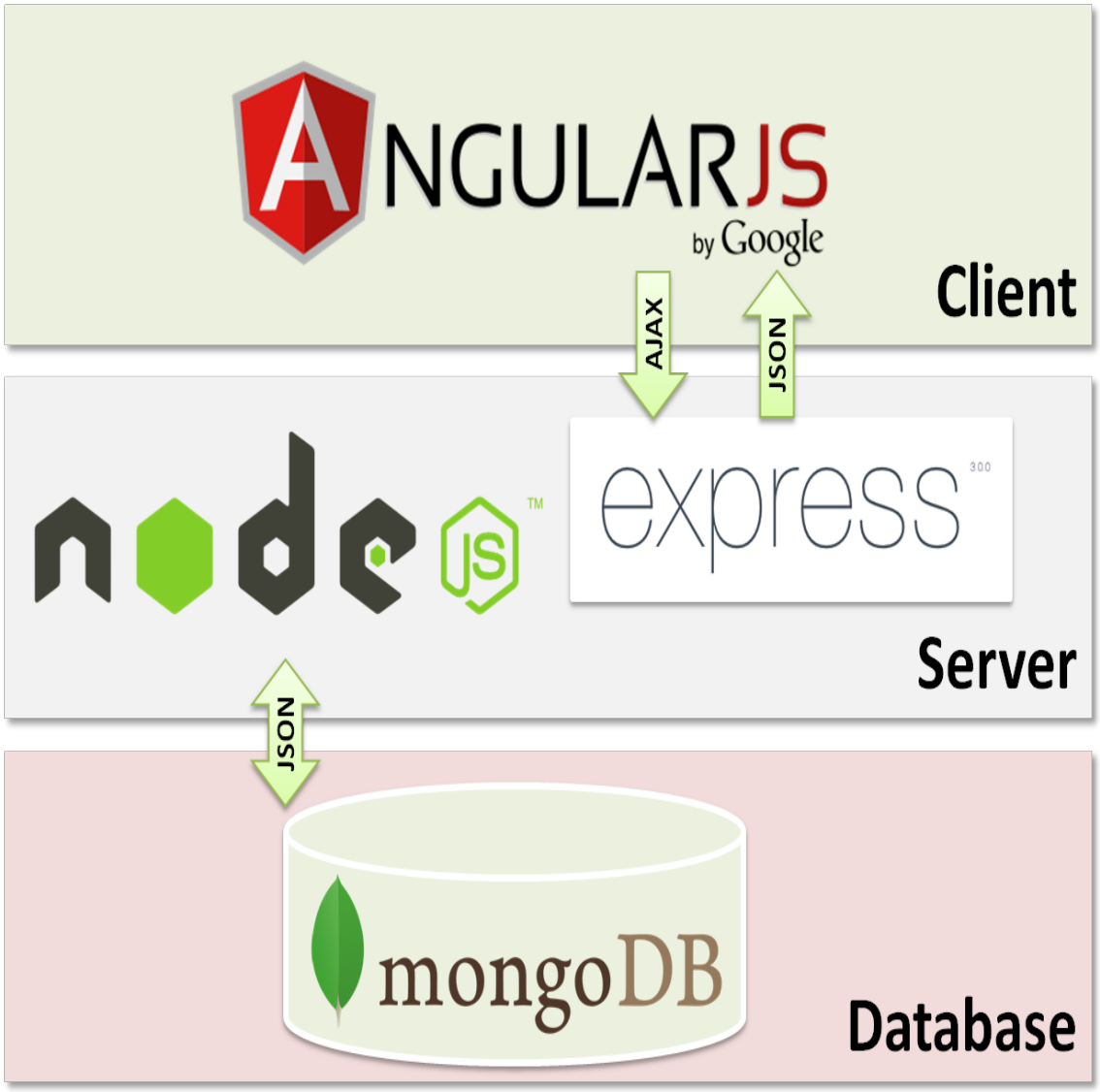


Figure 6.1: Architecture of stack

The above figure shows the architecture of the stack.

The below figure shows the proper working of stack.

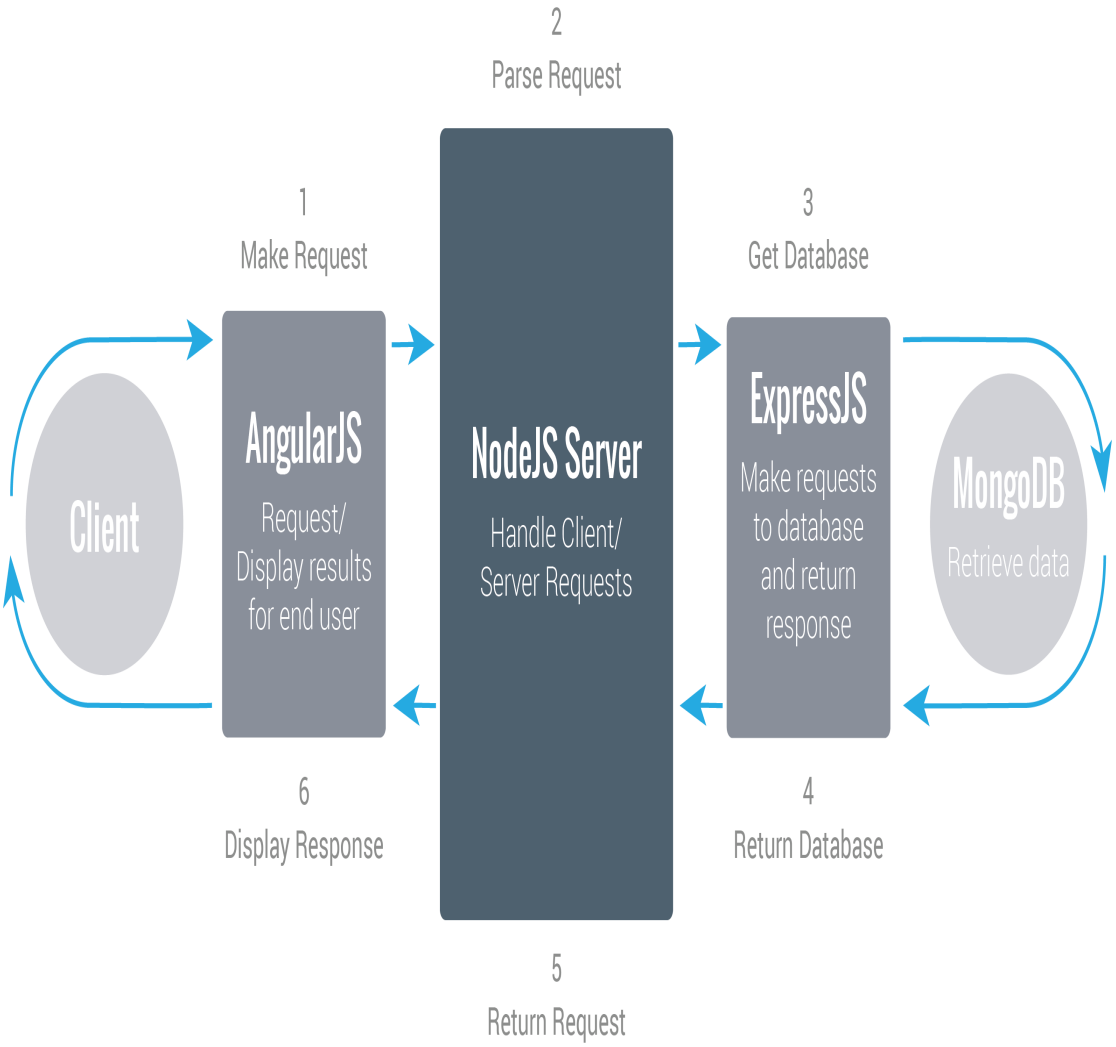


Figure 6.2: Working of stack

Chapter 7

COMPARISON WITH OTHER TECHNOLOGIES

Given below are the advantages and disadvantages of using other technologies available. Studying all these stacks showed that MEAN stack is better compared to existing technologies in web development.

LAMP STACK	.NET STACK	MEAN Stack
▪**Linux** : Can be used only with one OS.	▪**OS** : Microsoft Windows.	▪**OS (V8)** : Any OS compatible with node.js.
▪**Apache** : Back-end server without plug-in feature.	▪ ASP.NET Web API, is still **heavier** than NodeJS	▪**Node.js** : Back-end server with active plug-in feature.
▪**MySQL** : Relational DB with less efficiency to create, read, update, delete data.	▪ No **plug-ins**	▪**MongoDB** : Non-relational DB with efficiency to create, read, update, delete data.
▪**PHP or Python** : Scripting language	▪ The transitions it must go through from every MVC model to JSON, and back again i.e server side MVC	▪**Express and Angular.js** : Web page presentation.
▪Various data formats.	▪ Higher cost	▪JSON format

Figure 7.1: Comparison of Technologies for Web Development

Chapter 8

EXISTING APPLICANTS

Node.js

Microsoft, Google, PayPal, E-Bay, LinkedIn etc.

Express

MySpace, LearnBoost, Storify, Klout etc etc.

MongoDB

The NewYork Times, GitHub, BuzzFeed etc.

Angular.js

Google, NetFlix, YouTube etc. etc.

Chapter 9

CONCLUSION

Thus we can conclude the following points from the above analysis :

1. We can create Single page applications easily using this stack.
2. The only pre-requisite to use this stack, is to know JavaScript.
3. It blurs the lines between front-end and back-end.
4. No need to learn additional languages as it has JavaScript throughout !
5. We can have lots of plug-ins and thus better features.
6. MEAN stack is easy to code, run and update.
7. It has low cost.

REFERENCES

[1] Getting MEAN with Mongo, Express, Angular, and Node, Version 2 by Manning Publications.

[2] Write Modern Web Apps with the MEAN Stack by Jeff Dickey.

[3] http://www.nodebeginner.org/

[4] http://blog.scottlogic.com/2014/11/21/introducing-the-mean-stack.html

[5] http://thejackalofjavascript.com/mean-stack-hands-on-tutorial/