

“Vimeo Traffic Analytics Engine”

A

Project Report

*Submitted in partial fulfilment of the
requirements for the award of the degree of*

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE & ENGINEERING

by

Name

Roll No.

Paritosh Gupta

R110211052

Divya Punetha

R110211064

Under the guidance of

Dr. Neelu J. Ahuja



Department of Computer Science & Engineering

Centre for Information Technology

University of Petroleum & Energy Studies

Bidholi, Via Prem Nagar, Dehradun, UK

January – 2015



The innovation driven
E-School

CANDIDATE'S DECLARATION

I/We hereby certify that the project work entitled “**Vimeo Traffic Analytics Engine**” in partial fulfilment of the requirements for the award of the Degree of BACHELOR OF TECHNOLOGY in COMPUTER SCIENCE AND ENGINEERING with specialization in Cloud computing and virtualization technology) and submitted to the Department of Computer Science & Engineering at Center for Information Technology, University of Petroleum & Energy Studies, Dehradun, is an authentic record of my/ our work carried out during a period from **January, 2015** to **April, 2015** under the supervision of **Dr. Neelu J. Ahuja** , .

The matter presented in this project has not been submitted by me/ us for the award of any other degree of this or any other University.

Paritosh Gupta
Roll No.R110211052

Divya Punetha
Roll No.R10211064

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Dr. Neelu J. Ahuja
Project Guide

Dr. Amit Agarwal

Program Head – Cloud computing and virtualization technology
Center for Information Technology
University of Petroleum & Energy Studies
Dehradun – 248 001 (Uttarakhand)

ACKNOWLEDGEMENT

We wish to express our deep gratitude to our guide **Dr. Neelu J. Ahuja**, for all advice, encouragement and constant support he has given us throughout our project work. This work would not have been possible without his support and valuable suggestions.

We sincerely thank to our respected Program Head of the Department, **Dr. Amit Agarwal**, for his great support in doing our project in **Vimeo Traffic Analytics Engine** at **CIT**.

We are also grateful to **Dr. Manish Prateek**, Associate Dean and **Dr. Kamal Bansal** Dean COES, UPES for giving us the necessary facilities to carry out our project work successfully.

We would like to thank all our **friends** for their help and constructive criticism during our project work. Finally we have no words to express our sincere gratitude to our **parents** who have shown us this world and for every support they have given us.

Name	Divya Punetha	Paritosh Gupta
Roll No.	R110211064	R110211052

ABSTRACT

To provide insights concerning the video of users/channel to the customers through visual image. Vimeo Traffic Analytics Engine could be a project that goes to influence analysis of Vimeo traffic so it will offer insight concerning the video to the users. Vimeo attracts over a hundred million distinctive guests per month and over twenty two million registered users. Fifteen % of Vimeo's traffic comes from mobile devices. As of Gregorian calendar month 2013, Vimeo accounted for zero.11% of all net information measure, following fellow video sharing sites YouTube and Facebook.

Vimeo is a web portal during which users will transfer, share and look at videos. Analysis of traffic is critical for full optimisation. Users don't have any plan on that a part of the globe their uploaded videos area unit acknowledged or what form of users area unit a lot of inquisitive about their channel. Analysis of Vimeo traffic will offer insight concerning the video to the users.

The goal of the project is to research sizable amount of video connected knowledge points across classes, geographies, and sources so offer insights concerning the video of users/channel to the customers through visual image.

Vimeo is already operating in providing advanced statistics to its users on a paid subscription enclosed within the premium feature, we tend to conceive to develop a tool which might facilitate users gain insights concerning their videos while not their ought to subscribe premium plans and therefore serving to the scholars greatly.

CONTENTS

Declaration

Acknowledgement

Abstract

Contents

List of figures and tables

1. Introduction	7
1.1 History	7
1.2 Background	7
1.3 Current Problem	7
1.4 Objectives	7
2. Literature Review	8
2.1 Mean Stack	9
3. System Requirement Specification	11
4. Project Methodology	12
5. Conclusion	19
6. References	20

1. INTRODUCTION

1. Vimeo

Video sharing sites are getting common. New networking services bridge users to share video contents through video sharing sites like YouTube, Vimeo. Vimeo is a web portal known for being the high-quality home for ad-free HD videos over twenty million folks and businesses across the planet use Vimeo one amongst the foremost important specialty of Vimeo is it offers Stats regarding your videos. Analysation of Vimeo traffic will offer insight regarding the video to the users. Vimeo Traffic Analytics Engine target serving to the users to achieve elaborate insights and facilitate them reach a stronger audience by providing them details by analysing their video stats.

1.1. History

Vimeo may be a video sharing web site during which users will transfer, share and examine videos. Vimeo is changing into very talked-about among web users. Video Sharing has been observe for as long as web has been pervasive in our culture. Before 2005, once users wished to share video they were primarily listed through FTP sites or individual's web site. Some firms place their video in CD's and DVD's and gave them to customers though video sites were plentiful they didn't reside publically accessible repositories. There was no cohesive thanks to retrieve them however things modified with emergence of Vimeo. The emergence of Vimeo enabled folks to themselves be administrators and creators. These sites offer folks a free stage to point out their own merchandise and a platform for advertising through embedding and sharing in alternative places intrinsically there are incalculable video uploads to Vimeo daily.

1.2. Current Problem

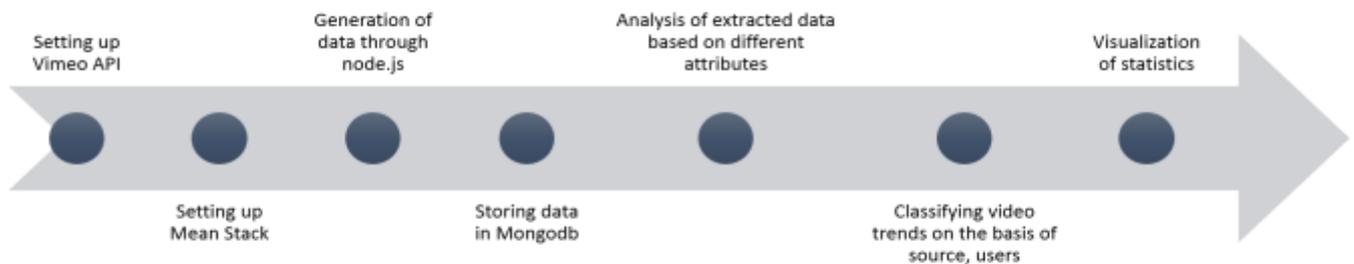
The market had been flourishing with video-sharing web site during which users will transfer, share and examine videos however analysation of traffic is critical for full improvement. Users haven't any plan on that a part of the planet their uploaded videos are acknowledged or what kind of users are additional curious about their channel. Analysation of Vimeo traffic will offer insight regarding the video to the users.

1.3. Main Objective

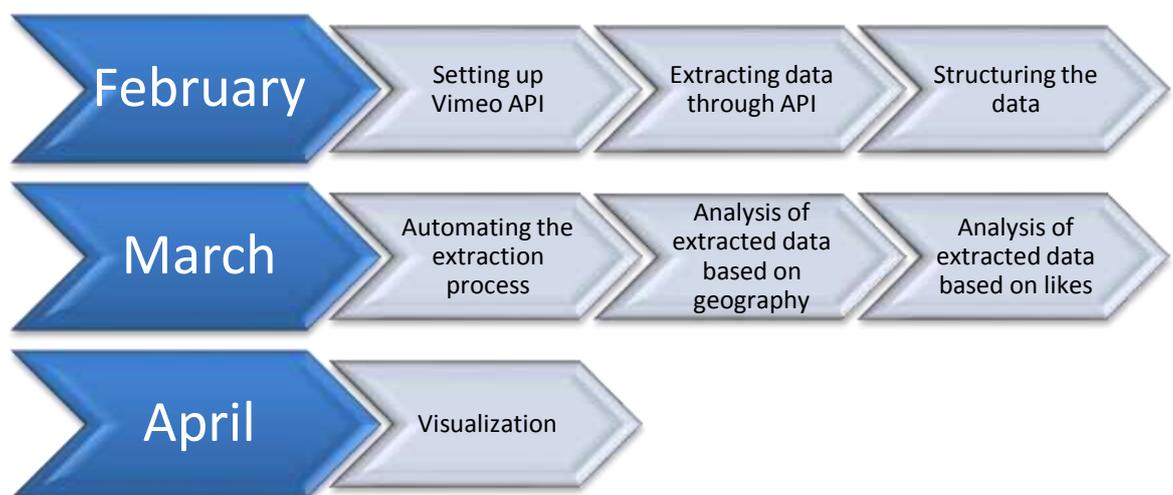
Vimeo Traffic Analytics Engine concentrate on serving to the users to achieve elaborated insights and facilitate them reach a higher audience by providing them details by analysing their video stats.

1.4. Sub Objectives

- To analyse sizable amount of video connected knowledge points across classes, geographies, and sources.
- To give insights regarding the video of users/channel to the shoppers through visualisation.



1.5. Pert Chart Legend



2. LITERATURE REVIEW

The project may be a sensible implication of the business theory that a product are often created higher by analysing the info points from the info generated for the videos uploaded/downloaded by users/channels, improve and take away its flaws, therefore this project is largely a sensible tool to assist the business to more enhance the videos. Vimeo is already operating in providing advanced statistics to its users on a paid subscription enclosed within the premium feature we have a tendency to attempt to develop a tool which might facilitate users gain insights regarding their videos while not their got to subscribe premium plans and therefore serving to the scholars greatly.

2.1 MEAN STACK

MEAN encompass the four elements it's a free and ASCII text file answer stack for building dynamic websites. MEAN.JS helps in obtaining started and it avoids useless grunt work and customary pitfalls, whereas keeping your application organized. MEAN may be a full-stack JavaScript framework, which may assist you quickly develop your applications. The elements of the MEAN stack ar as follows:

- MongoDB: it's a NoSQL database;
- Express.js: it's an online applications framework;
- Angular.js: it's a JavaScript MVC framework for internet apps;
- Node.js: it's a package platform for ascendable server-side and networking applications.

2.1.1 MongoDB

It is a document-oriented info. Mongo, modify the utilization of one language the entire method through it's classified as a NoSQL info. Mongolian monetary unit could also be an honest work here. In different cases, you will have a lot of advanced transactions concerned several separate logical units. MongoDB eschews the normal table-based electronic database structure in favor of JSON-like documents with dynamic schemas

(MongoDB calls the format BSON), creating the mixing of knowledge in bound sorts of applications easier format BSON), making the integration of data in certain types of applications easier and faster. NoSQL but is weak written, creating only a few makes an attempt at information validation so responsibility is placed back on the developer. The technical variations between SQL databases vs. no SQL extremely return right down to an exchange between performance and dependableness free beneath a mix of the antelope General Public License and therefore the Apache License.

Key Features:

High Performance

MongoDB provides high performance information persistence. It provides supports for embedded information models reduces I/O activity on information system. Indexes in MongoDB support quicker queries they will embody keys from embedded documents and arrays.

High Availability

To provide high availability, Mongo DB's replication facility, known as replica sets, provide:

- Automatic failover.
- data redundancy

Automatic Scaling

MongoDB provides horizontal scalability as part of its *core* practicality.

- Automatic sharing distributes information across a cluster of machines.
- Replica sets will give eventually-consistent reads for low-latency high throughput deployments.

2.1.2 Express.js

Express could be a utility belt for authoring internet apps with Node. Its provides functions to make servers and utility functions for parsing the POST request body. Some of the options of specific are:

Web Applications

Express could be a versatile Node.js internet application framework that has sort of strong set of options for internet and mobile applications

APIs

With the assistance of HTTP utility strategies and middleware, making a sturdy API is fast and straightforward.

Performance

Express provides economical performance and skinny layer of basic internet application options, while not obscuring.

2.1.3 Angular.js

AngularJS, conjointly referred to as Angular, is associate degree ASCII text file framework created and maintained by Google and a community of individual developers. AngularJS is intended to make single-page applications with the assistance of the MVC design. AngularJS applications area unit a mixture of HTML and JavaScript. AngularJS helps to increase HTML vocabulary for your application. Its goal is to change each development and testing of applications by providing a framework for client-side model-view-controller (MVC) design, two-way information binding. Binding could be a core feature of AngularJS that synchronizes between models and views beside parts unremarkably employed in made net applications. The models and controllers area unit specified via JavaScript objects and JavaScript functions. The setting is communicatory, readable, and fast to develop. AngularJS is totally protractile and works well with alternative libraries during this each feature may be changed or replaced to suit distinctive development progress and have wants of the user.

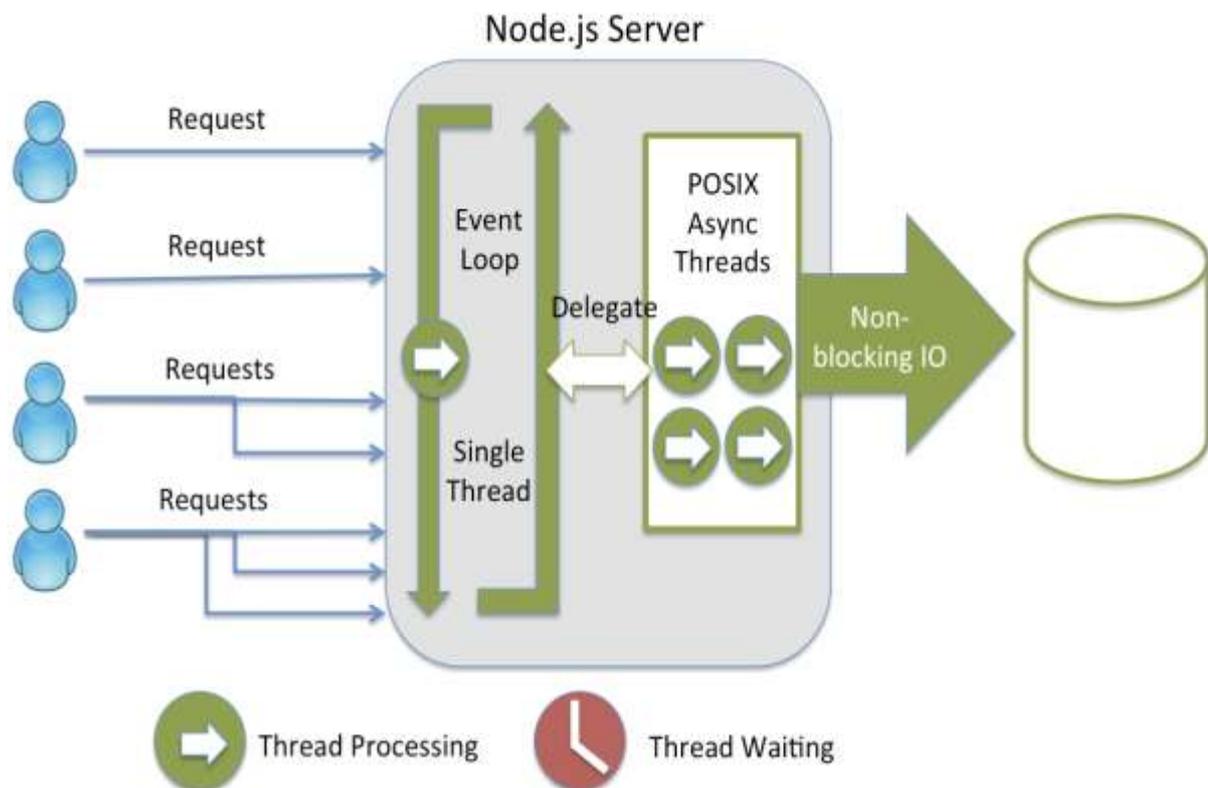
It this library works by presentation the HTML page, that has embedded into it further custom tag attributes. Those attributes area unit taken as directives telling Angular to bind input or output elements of the page to a model that's drawn by normal JavaScript variables. The values of these JavaScript variables may be manually set among the code, or retrieved from static or dynamic JSON resources.

2.1.4 Node.js

Node.js is an open source, runtime environment for server-side and networking applications. Node is single-threaded and uses a concurrency model based on an event loop. Node.js is a headless JavaScript runtime and can be run within the Node.js runtime on OS X etc.

Node.js provides an event-driven architecture and a non-blocking I/O API that optimizes an application's throughput and scalability. It is non-blocking, so it doesn't make the program wait instead it registers a callback and lets the program continue. These technologies are commonly used for real-time web applications.

Node.js uses the Google V8 JavaScript engine to execute code, and a large percentage of the basic modules are written in JavaScript. In Node, all of the libraries have been designed from the ground up to be non-blocking, but the same cannot be said for the others. Node.js contains a built-in library to allow applications to act as a Web server without software such as Apache HTTP Server or IIS.



2.1.5 Tableau

Tableau software system is Associate in Nursing yank pc software system company headquartered in point of entry, WA, USA. It produces a family of interactive knowledge image merchandise targeted on business intelligence. In 2010, Tableau rumored revenue of \$34.2 million greenbacks.

It is terribly simple to use:

You don't ought to understand programming of any kind, all you would like is a few knowledge and tableau to make reports that square measure visually captivating and that tells a story that you would like to inform our managers or impress your faculty member in school.

With its revolutionary drag and drop feature u will simply produce stories or reports mistreatment simply your mouse and a trifle imagination. All this can be potential as a result of the revolutionary VizQL a visible source language

VizQL

VizQL could be a visual command language that interprets drag-and-drop actions into knowledge queries and so expresses that knowledge visually. VizQL delivers dramatic gains in people's ability to examine and perceive knowledge by abstracting the underlying complexities of question and analysis. The result's associate degree intuitive user expertise that lets individuals answer queries as quick as they'll consider them we tend to believe that VizQL represents a foundational advancement within the space of knowledge analysis and image.

The fundamental innovation could be a proprietary command language that interprets your actions into an information question and so expresses the response diagrammatically attributable to VizQL, quick analytics and image are reality individuals with very little or no coaching will see and perceive knowledge quicker than ever and in ways that like ne'er before. And that's the largest distinction of all.

Advanced In-Memory Technology- The Data Engine

Most of the information analytic software package have lots of fancy options however the majority of them fail once it involves operational with giant amounts of knowledge, this can be wherever the advanced in memory technology of tableau could be a saviour to any or all of these WHO ought to get reports from ever increasing knowledge. The tableau knowledge engine could be a breakthrough in-memory analytics information designed to beat the constraints of existing knowledge bases and data silos

Tableau desktop

Tableau Desktop is predicated on breakthrough technology from Stanford that allows you to drag & drop to analyse information. You'll hook up with information in a very few clicks, then visualize and build interactive dashboards with a couple of a lot of.

Tableau server

Tableau Server could be a business intelligence application that has browser-based analytics anyone will use. It's a rapid-fire various to the slow pace of ancient business intelligence package.

Tableau on-line

Tableau on-line could be a hosted version of Tableau Server. It makes rapid-fire business analytics easier than ever. Tableau on-line will proportion the maximum amount as you would like and you'll use it where you would like it. you do not got to purchase, found out or manage any infrastructure.

Tableau public

Tableau Public is for anyone United Nations agency needs to inform stories with interactive information on the online. It's delivered as a service that enables you to be up and running long. With Tableau Public you'll produce wonderful interactive visuals and publish them quickly, while not the assistance of programmers or IT. The Premium version of Tableau Public is for organizations that wish to boost their websites with interactive information visualizations. There square measure higher limits on the dimensions of information you'll work with. And among alternative premium options, you'll keep your underlying information hidden.

Strengths

Speed

The greatest strength of Tableau is its speed with that it analyses many rows and provides the desired answers in seconds. This quality comes in terribly handy once quick business selections got to be taken.

Ease of use

Another strength of Tableau is that it's terribly straightforward to use. It's simply basic drag and drop. One will begin exploitation Tableau even with no previous programming expertise. With simply basic MS stand out skills one will simply learn Tableau.

Beautiful and interactive dashboard

The Dashboard of Tableau is extremely interactive and provides dynamic results. Wealthy visualizations are often created terribly simply. The graphics and charts square measure good and exquisite. Images, web content and documents are often extra into the dashboard for simple story telling. All this ends up in a far insight into the information.

Direct affiliation

Tableau permits the users to directly hook up with databases, cubes, and information warehouses etc. {the information |the info| the information} access is really easy with none advanced setup and therefore the data is live that keeps obtaining updated on its own. One will decide tables from spreadsheets to information from Hadoop to make an ideal mash-up and obtain desired ends up in no time this can be easy unintended business analytics.

Easy publication and sharing

After analysing the information, the results are often shared stomach simply a couple of clicks. The dashboard are often revealed to share it survive internet and mobile devices.

Growing market

Tableau is comparatively new within the Business Intelligence market however its market share is growing on a commonplace it's being nearly all industries, from transportation to tending.

Weakness

Not comprehensive resolution, focus on Bi. Tableau package is a simple to use Bi application, however still not a platform for developing analytic applications that may be loosely shared it's not designed for enterprise wide deployments that have broad business and technical user populations.

No prognosticative analytical capabilities

Tableau package permits users to solely analyze what is going on. And, Tableau package doesn't have any applied mathematics or prognosticative capabilities (they don't support SAS, S+, R, etc. like Spotfire does) thus users cannot model or project the most effective cross-selling or upselling opportunities really Tableau package is datumally “challenged” and is extraordinarily limited; providing users with less statistic functions than Microsoft stand out.

Customization and Integration with alternative apps

Due to its not ASCII text file based mostly nature and market goods, Tableau package can't be made-to-order or integrated with alternative applications (such as Salesforce.com) essentially what comes within the box is what you get. No thanks to extend it. However, their official web site has Forum section, users may vote for his or her ideas, whenever reaching the brink, the corporate would response and doubtless add such options within the following version and update for purchased packages.

Expandability for analytics

Tableau package doesn't support expanded analytics like Box plots, network graphs, tree-maps, heat-maps, 3D-scatter plots, Profile Charts or information relationships tool

which permit users to mine information for relationships like another information image package will.

Social media integration

Tableau package doesn't integrate cooperative capabilities into social platforms (like Twitter, Microsoft SharePoint, etc.). a awfully hot trend. it's for Business users to share their findings and progress think about it like – email meets Twitter meets Facebook. Due the present trend and users' would like, Tableau has integrated Google Analytics into its platform.

2.1.6 Chance.js Generator

It is a minimalist generator of random strings, numbers, etc. to assist cut back some monotony significantly whereas writing machine-driven tests. Chance.js are often utilized in browser or node.js, and it additionally support AMD module so you'll simply import in need.js. Chance could be a minimalist generator of random strings, numbers, etc. to assist cut back some monotony significantly whereas writing machine-driven tests or anyplace else you would like something random. Chance.js are often utilized in browser or node.js, and it additionally support AMD module so you'll simply import in need.js

Chance is open supply package and is discharged beneath the developer and business-friendly Massachusetts Institute of Technology license. Chance is loaded on this website therefore you'll simply open the console on your browser and play!

Chance instantiates itself onto the window this implies that within the simplest case you'll simply embrace the script tag then use associate degree instance of likelihood straight off

3.SYSTEM REQUIREMENT SPECIFICATION

Requirement Specification:

Hardware:

- ✓ System with quad core INTEL or AMD processor
- ✓ 8 GB RAM
- ✓ High Bandwidth Internet Connection

Software:

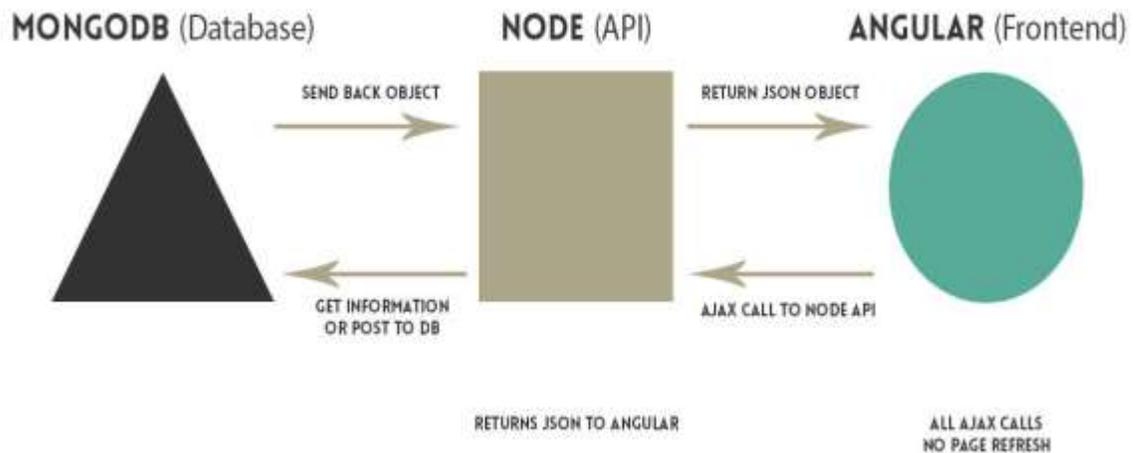
- ✓ Python 2.7
- ✓ Mean Stack
- ✓ Express.js
- ✓ Node.js
- ✓ Angular.js
- ✓ Database: MongoDB
- ✓ Faker.js
- ✓ Choice.js
- ✓ Tableau

4.PROJECT METHODOLOGY

4.1 Setting up MEAN Stack:

In MEAN Stack there are 4 main parts:

- MongoDB which acts as the database
- Express which acts as the web framework
- AngularJS which acts as the frontend framework, and
- Node.js which acts as the server platform



4.2 Installing MongoDB

Download MongoDB, from the official website.

The installation process is as follows:

- Download and extract the archive file of MongoDB
- After that, run mongod binary
- Then, create the folder to store the database files (the default folder is /data/db).
- Lastly, make sure port 27017 is free to use

4.2.1 Installing MongoDB on a Windows machine

After downloading archive version of MongoDB, unpack the archive file, and move the folder to C:\mongodb. MongoDB uses a default folder to store its files. On Windows machine the default location is:

- **C:\>mkdir data\db**

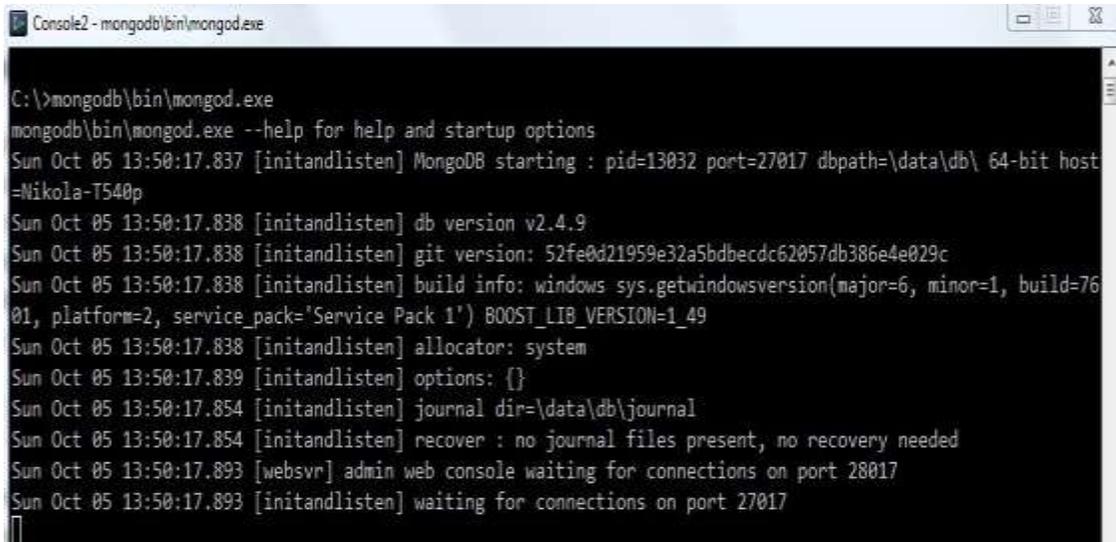
There are two options for running a MongoDB service:

Running manually

Run the mongod binary by executing the following command in command prompt:

- **C:\>mongodb\bin\mongod.exe**

This started the MongoDB service on a default 27017 port, and if everything went well you should see a console output like the one on the screenshot below:



```
C:\>mongodb\bin\mongod.exe
mongodb\bin\mongod.exe --help for help and startup options
Sun Oct 05 13:50:17.837 [initandlisten] MongoDB starting : pid=13032 port=27017 dbpath=\data\db\ 64-bit host
=Nikola-T540p
Sun Oct 05 13:50:17.838 [initandlisten] db version v2.4.9
Sun Oct 05 13:50:17.838 [initandlisten] git version: 52fe0d21959e32a5bdbecdc62057db386e4e029c
Sun Oct 05 13:50:17.838 [initandlisten] build info: windows sys.getwindowsversion(major=6, minor=1, build=76
01, platform=2, service_pack='Service Pack 1') BOOST_LIB_VERSION=1_49
Sun Oct 05 13:50:17.838 [initandlisten] allocator: system
Sun Oct 05 13:50:17.839 [initandlisten] options: {}
Sun Oct 05 13:50:17.854 [initandlisten] journal dir=\data\db\journal
Sun Oct 05 13:50:17.854 [initandlisten] recover : no journal files present, no recovery needed
Sun Oct 05 13:50:17.893 [websvr] admin web console waiting for connections on port 28017
Sun Oct 05 13:50:17.893 [initandlisten] waiting for connections on port 27017
```

Running as a Windows service

This may be an additional favorable answer as a result of you won't get to begin the MongoDB service manually anytime you boot your laptop, instead Windows service can begin it mechanically once boot so as to try to therefore, you initially got to produce a log file by executing:

```
C:\>mkdir C:\mongodb\log
```

Open up a new command prompt window with administrative privileges.

Finally, to install the MongoDB as a service execute:

```
sc.exe create MongoDB binPath= "\"C:\mongodb\bin\mongod.exe\" --service --
config= "\"C:\mongodb\mongod.cfg\""" DisplayName= "MongoDB 2.6" start=
"auto"
```

After installing MongoDB service, you can run it by executing the following command in the administrative command prompt window:

```
Net start MongoDB
```

4.3 Installing Node.js

Download Node.js from the official website

Step 1: Install Node.js



Running Node.js

After you successfully installed Node.js, following command is executed from command line tool:

- `node`

This will start the Node.js command-line interface (CLI). In order to test it execute the following command:

- `console.log('Hello World from Node.js');`

which will output:

Hello World from Node.js

4.4 Installing Express

To install express execute following command:

```
npm install express
```

4.5 Setting up AngularJS

AngularJS is a frontend JavaScript framework designed to build single-page applications using the MVC architecture. It's built and maintained by Google so this should give you a peace of mind.

Few of the cool features are:

- two-way data binding, which synchronizes between models and views
- extended HTML with additional attributes, which bind the JavaScript objects with HTML elements
- improved code structure
- easier testing through dependency injection

Download AngularJS from official website.

Bower is a package manager which makes it easier to download and maintain your frontend libraries.

Once the Bower is installed then install Angular:

Bower places all of its files within the `bower_components` folder within the root folder of your application, particularly within the folder wherever you dead the bower install command to put in your frontend packages in an exceedingly completely different location, attend the basis folder of your application and build a file named `.bowerrc` Bower, like Node.js, uses a config file named `bower.json` wherever you'll outline your dependencies.

4.5 Setting up Vimeo API

The first step in order to create a service was to successfully setup the Vimeo API. This API provides various methods for fetching the data based upon different attributes. We chose Python to setup the Vimeo API in order to fetch the relevant data of all the public videos.

Vimeo offers us its service to access its API and help us build different kind of applications on top of their data. Vimeo provides two kinds of API – “Simple API” and “Advanced API”. We have setup the Advanced API which required us to first create a developer account to fetch the appropriate authentication tokens. These will further be used for read/write requests on videos, categories, channel, users, and albums or uploads. The Advanced API used OAuth for authentication, thus is the need to sign up.

While using Simple API, we have the option of accessing it directly and we do not need to create any account. The biggest limitation of Simple API is that we have the restriction of accessing only the public and read-only data. Also, only 20 items per page is the response limit in Simple API. Vimeo API provides us different response formats and they all return same results.

4.5.1 Common Vimeo Requests

- Making an Album Request
- Making an User Request
- Making an Video Request
- Making an Channel Request
- Making an Activity Request

4.5.2 Using the Vimeo API to get data

```
import requests
import json
r = requests.get("http://vimeo.com/api/v2/video/48082757.json")
r.text
data = json.loads(r.text)
'do something with the data'
```

```

1 import vimeo
2 import json
3
4 v = vimeo.VimeoClient(
5     token="48d86b945868402de6ece6712651d81f",
6     key="17b844d3d441ff4b423b0357dd1084a3505fcc3e",
7     secret="8a582eb892bd6b909bc455c5ff9f0b54e1bdb16e")
8
9 # Make the request to the server for the "/me" endpoint.
10 about_me = v.get('/categories/animation/videos/94502406')
11
12 assert about_me.status_code == 200 # Make sure we got back a successful response.
13 print json.dumps(about_me.json(), indent=1) # Load the body's JSON data self.

```

```

1 import vimeo
2
3 v = vimeo.VimeoClient(
4     key="17b844d3d441ff4b423b0357dd1084a3505fcc3e",
5     secret="8a582eb892bd6b909bc455c5ff9f0b54e1bdb16e")
6
7 try:
8     token = v.load_client_credentials()
9     print token
10 except vimeo.auth.GrantFailed:
11     print exception
12     # Handle the failure to get a token from the provided code and redirect.

```

```

1 import requests
2
3 import json
4
5 r = requests.get("http://vimeo.com/api/v2/video/68083055.json")
6
7 r.text
8
9 data = json.loads(r.text)
10 print json.dumps(data)

```

4.6 Generation of data using node.js

When we start creating an application, we often come across the challenge of showcasing that how it will work. It is totally useless if all the tables in our application are blank. When we start creating our application, we do not start by thinking about the data but instead we first start by focusing on the logic that our application requires. After the creation of the logic, we need to focus on creating a database and populating it with simple users, which is often an annoying step. This step is very important as we cannot proceed without having sufficient data. Thus comes the need of creating fake data.

Generated massive amount of data in node.js. The overall category list includes name, address, phone etc. Each of those categories has various options within itself. The major problem faced while working with a specific API of any company is that, whenever a company updates its API, then the complete service is broken down as the data or schema in which data was originally being returned might have been changed, which makes our existing software unable to work on that particular software. Thus after detailed research and guidance, we came up with the idea of generating our own data, i.e. fake data, which would be similar to the form of data that would have been returned by the original website or API of the company.

We started with exploring various API's which returned video JSON records when requested by a client or user. Each and every different API had its own unique set of attributes which could help the uploader gain more insights about their uploaded videos. After careful comparison between all the different JSON fetched by the API, we came up with up very own schema which according to us would be the best available schema for a user to gain insights about his/her videos.

In order to generate fake data, we tried a lot of different softwares available. We had the option of using any one out of the following:

- Faker.js
- Choice.js

4.6.1 Generation of Data using Faker.js

Let us begin by creating a simple user with very few details like *name*, *email*, *address*, *bio*, and even an *image*.

In Node, using Faker.js all we need to do is :

```
var user = {  
  
  name: faker.name.findName(),  
  
  email: faker.internet.email(),  
  
  address: faker.address.streetAddress(),  
  
  bio: faker.lorem.sentence(),  
  
  image: faker.image.avatar()  
  
};
```

The output is all the required fields in a JSON format.

```
{  
  
  name: 'June Klein',  
  email: 'Vicky9@yahoo.com',  
  address: '0560 Micheal Shoal',  
  bio: 'veniam adipisci voluptates maxime nesciunt ea eaque quaerat ut',  
  image: 'https://s3.amazonaws.com/uifaces/faces/twitter/timpetricola/128.jpg'  
  
}
```

The biggest challenge we faced while using faker.js was the inability to create fake data using custom values. While creating fake data, there was often a need to create custom dictionaries which would be further used while generating fake data.

4.6.3 Generation of Data using Casual.js

We overcame the problem faced by Faker.js by shifting to Casual.js. The advantage of Casual.js is that it gives us the option of using custom dictionaries.

The following are the advantages of Casual.js:

random element

Get random array element

```
var item = casual.random_element(['ball', 'clock', 'table']);
```

random value

Extract random object value

```
var val = casual.random_value({ a: 1, b: 3, c: 42 });
```

```
// val will be equal 1 or 3 or 42
```

random key

Extract random object key

```
var val = casual.random_key({ a: 1, b: 3, c: 42 });
```

```
// val will be equal 'a' or 'b' or 'c'
```

populate

Replace placeholders with generators results

```
casual.populate('{{ email }} {{ first_name }}');
```

```
// 'Dallin.Konopelski@yahoo.com Lyla'
```

populate one of

Pick random element from given array and populate it

```
var formats = ['{{ first_name }}', '{{ last_name }} {{ city }}'];
```

```
casual.populate_one_of(formats);
```

```
// Same as
```

```
casual.populate(casual.random_element(formats));
```

numerify

Replace all # in string with digits

```
var format = '(##)-00-###-##';
```

```
casual.numerify(format); // '(10)-00-843-32'
```

Using the advantages of Casual.js it gave us the feasibility to create fake data the schema decided by us.

The schema of json being generated by us is:

```
casual.define('videoRecord', function() {
  var username = casual.username;
  var video_id = casual.integer(from = 1000000, to = 9999999);
  return
  {
    "title": casual.title,
    "description": casual.description,
    "id": video_id,
    "video_url": "http://www.vimeo.com/"+video_id,
    "ip": casual.ip,
    "author": {
      "author_name": username,
      "url": "http://www.vimeo.com/"+username,
      "email": casual.email,
      "location":
      {
        "city": casual.city,
        "state": casual.state,
        "country": casual.country
      },
      "followers": casual.integer(from = 0, to = 1000),
      "total_uploads": casual.integer(from = 0, to = 500),
```

```

        "company":casual.company_name,
        "following":casual.integer(from = 0, to = 300),
        "shares":casual.integer(from = 0, to = 100)
    },
    "created_date": casual.date(format = 'DD-MM-YYYY'),
    "created_time": casual.time(format = 'HH:mm:ss'),
    "plays": array_of(casual.integer(from = 0, to = 100),casual._details),
    "likes": array_of(casual.integer(from = 0, to = 50),casual._details),
    "shares": array_of(casual.integer(from = 0, to = 20),casual._details),
    "downloads": array_of(casual.integer(from = 0, to = 20),casual._details),
    "comments":array_of(casual.integer(from = 0, to = 20),casual._comments),
    "privacy": {
        "downloadable":casual.random_element(['yes']),
        "embed": casual.random_element(['private','public']),
        "commentable":
casual.random_element(['anybody','friend_of_friend','none','custom']),
        "view":
casual.random_element(['anybody','friend_of_friend','none','custom']),
    },
    "status": casual.random_element(['available','unavailable']),
    "tags": casual.array_of_words(casual.integer(from = 1, to = 15)),
    "language": casual.language_code,
    "content_rating": casual.random_element(['safe','13+','18+']),
    "playbar": casual.random_element(['yes','no']),
    "buttons": {
        "fullscreen":casual.random_element(['yes','no']),
        "like":casual.random_element(['yes']),
        "share":casual.random_element(['yes']),
        "download":casual.random_element(['yes']),
        "embed":casual.random_element(['yes']),
        "hd":casual.random_element(['yes','no']),
    }

```

```

        "watch_later":casual.random_element(['yes','no'])
    },
    "subtitles":casual.random_element(['yes','no']),
    "traffic_source":
casual.random_element(['vimeo_search','google_search','direct']),
    "category":
casual.random_element(['animation','arts&design','motivation','comedy','documentart','ex
perimental','fashion','food','instructionals','music','personal','sports','travel','technology','tal
ks'])
    };
});

```

4.6.3 Example JSON

```

{
  "title":"Aut deleniti",
  "description":"Saepe voluptatem ullam doloribus accusamus quis
nulla tenetur incidunt. Velit iure nam vero perferendis. Esse sit
exercitationem. Voluptatem rem vitae et culpa nam fugiat assumend
a pariatur.",
  "id":7717607,
  "video_url":"http://www.vimeo.com/7717607",
  "ip":"21.199.122.173",
  "author":{
    "author_name":"Ryan.Eldora",
    "url":"http://www.vimeo.com/Ryan.Eldora",
    "email":"Burdette.Pollich@Buckridge.tv",
    "location":{
      "city":"Port Estrella",
      "state":"New Hampshire",
      "country":"Norway"
    }
  }
}

```

```

    },
    "followers":711,
    "total_uploads":316,
    "company":"Borer PLC",
    "following":61,
    "shares":0
  },
  "created_date":"06-06-1975",
  "created_time":"15:55:10",
  "plays":[
    {
      "name":"Valentin.Greenfelder",
      "url":"http://www.vimeo.com/Valentin.Greenfelder",
      "city":"Susiechester",
      "state":"Indiana",
      "country":"Niger",
      "lat":"17.8918",
      "lon":"137.2555",
      "date":"04-12-1978",
      "time":"22:30:39",
      "user_agent":"Mozilla/5.0 (Windows NT 6.3; WOW64; Trident/7.0; rv:11.0) like Gecko"
    },
    {
      "name":"Elwin.Robel",
      "url":"http://www.vimeo.com/Elwin.Robel",
      "city":"West Eudora",
      "state":"Georgia",
      "country":"Botswana",
      "lat":"39.7745",
      "lon":"-165.7865",

```

```
    "date": "23-09-2010",
    "time": "09:31:28",
    "user_agent": "Mozilla/5.0 (Windows NT 6.3; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/39.0.2171.95 Safari/537.36"
  }
],
"likes": [
  {
    "name": "Jessy.Reinger",
    "url": "http://www.vimeo.com/Jessy.Reinger",
    "city": "Emanueland",
    "state": "Connecticut",
    "country": "Indonesia",
    "lat": "-68.3085",
    "lon": "-177.6757",
    "date": "17-10-1986",
    "time": "15:20:45",
    "user_agent": "Mozilla/5.0 (Windows NT 6.0) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/39.0.2171.95 Safari/537.36"
  },
],
"shares": [
  {
    "name": "Marks.Orlando",
    "url": "http://www.vimeo.com/Marks.Orlando",
    "city": "East Johnny",
    "state": "Indiana",
    "country": "Jersey",
    "lat": "-80.9236",
    "lon": "-136.5042",
```

```
    "date": "30-04-1999",
    "time": "22:13:09",
    "user_agent": "Mozilla/5.0 (Windows NT 6.1; WOW64; rv:35.0) Gecko/20100101 Firefox/35.0"
  }
],
"downloads": [
  {
    "name": "Chaim.Jacobson",
    "url": "http://www.vimeo.com/Chaim.Jacobson",
    "city": "Lake Elda",
    "state": "California",
    "country": "Latvia",
    "lat": "-7.4861",
    "lon": "-105.5661",
    "date": "03-07-1998",
    "time": "20:09:55",
    "user_agent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_10_1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/40.0.2214.45 Safari/537.36"
  }
],
"comments": [
  {
    "id": 92209857,
    "name": "Darian.Christiansen",
    "message": "Neque itaque nostrum laboriosam nisi eos est non voluptas. Deserunt molestias sed ex. Voluptas maxime harum harum nemo modi reiciendis nemo quasi. Voluptas sed porro omnis ea voluptas tempora earum et sint. Perspiciatis ipsam molestiae necessitatibus inventore nemo.",
```

```
        "created_date": "05-09-1973",
        "created_time": "16:45:57",
        "city": "West Destiny",
        "state": "West Virginia",
        "country": "Iraq",
        "lat": "-31.0122",
        "lon": "98.2231"
    }
],
"privacy": {
    "downloadable": "yes",
    "embed": "private",
    "commentable": "custom",
    "view": "friend_of_friend"
},
"status": "available",
"tags": [
    "ea",
    "corporis",
    "veritatis",
    "dolorum",
    "fuga",
    "impedit",
    "dolores",
    "recusandae"
],
"language": "de",
"content_rating": "safe",
"playbar": "no",
"buttons": {
    "fullscreen": "yes",
```

```
    "like": "yes",
    "share": "yes",
    "download": "yes",
    "embed": "yes",
    "hd": "yes",
    "watch_later": "yes"
  },
  "subtitles": "no",
  "traffic_source": "google_search",
  "category": "music"
}
```

4.7. Storing Data in MongoDB

Before the generation of fake data in large amount, we also need to make sure we have a place ready to store it. We have used MongoDB as our database to store our data. The benefit of using MongoDB is that it is a NoSQL database and it allows us to store JSON records directly. It has the capability to automatically adapt to the JSON and modify the schema as per its own needs.

The sample of the script we are using to store data in our Database is :

```
// Use connect method to connect to the Server

MongoClient.connect(url, function (err, db) {

  if (err) {

    console.log('Unable to connect to the mongoDB server. Error:', err);

  } else {

    //HURRAY!! We are connected. :)
  }
}
```

```
console.log('Connection established to', url);

// Get the documents collection

var collection = db.collection('vimeo_traffic');

for (i = 1; i <= 50; i++)
{
    var videoRecord = casual.videoRecord

    // console.log(videoRecord);

    fs.appendFile('data.json',JSON.stringify(videoRecord)+'\n');

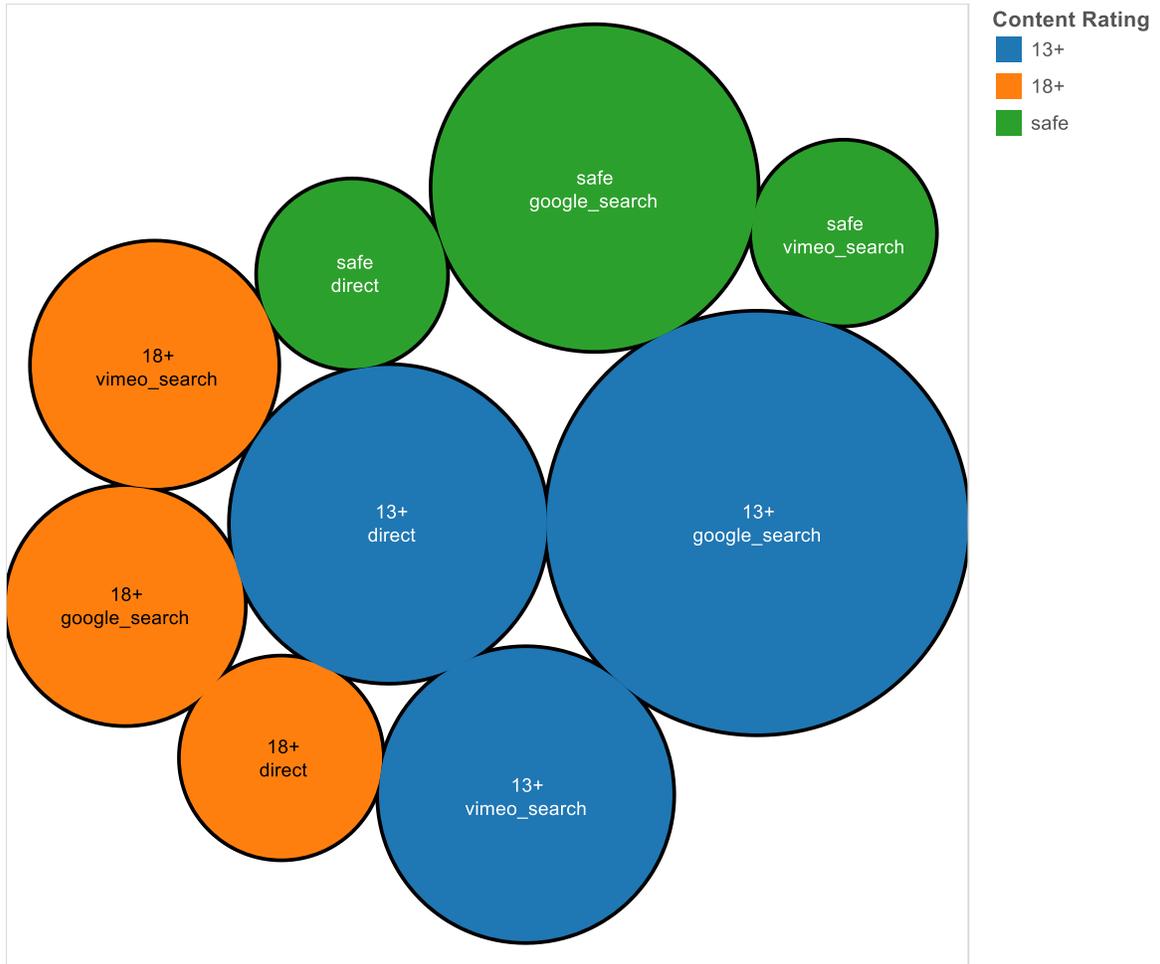
    collection.insert(videoRecord, function (err, result) {
        if (err) {
            console.log(err);
        } else {
            console.log('Inserted %d documents into the "users" collection. The
documents inserted with "_id" are:', result.length, result);
        }
    })
}
```

```
        });  
    }  
  
    db.close();  
  
}  
  
});
```

4.8. Analysis And Visualization of Data

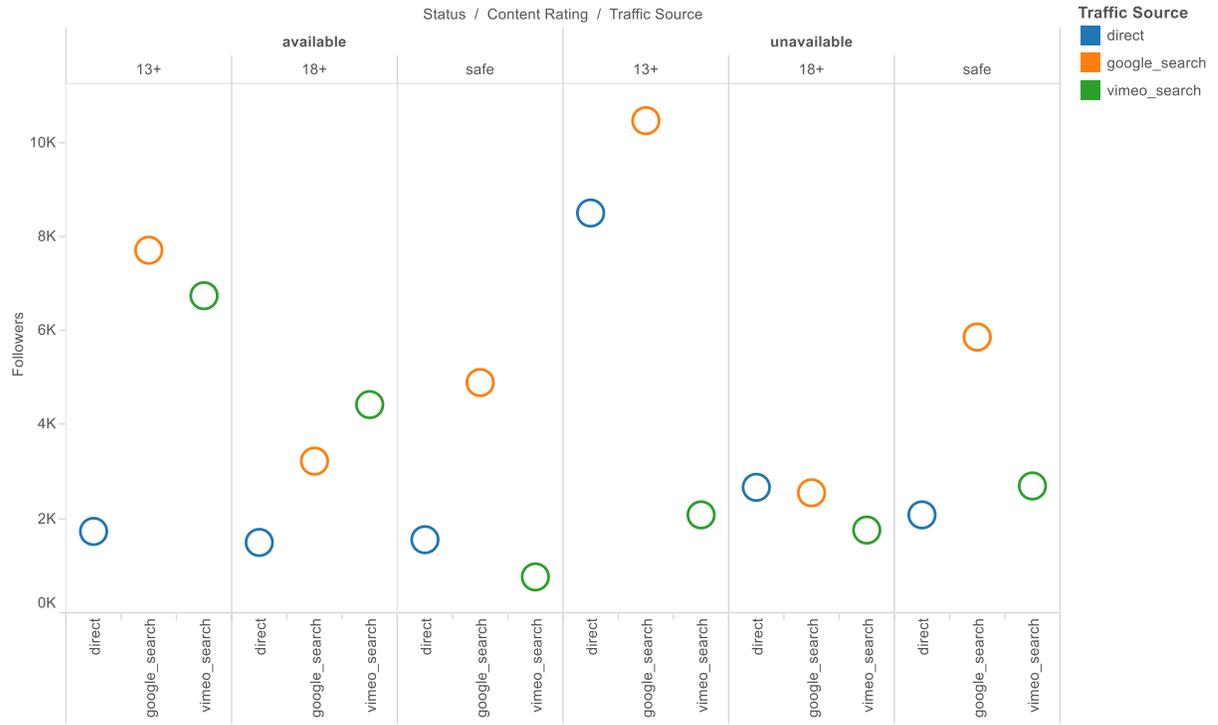
The final step that we come across after generation of fake data is that we need to perform analysis on top of the data. We need to carefully plan out different attributes from the given schema of the Video JSON, which would further help a user/channel/company for gaining meaningful information about their uploaded videos. The last step analysis of data, i.e. after deciding the various attributes which can be compared with each other and can be scaled up or down, or can be drilled down or rolled back up, is visualization of data. The visualizations can be of various forms like 2d graphs, pie charts, text tables, etc.

Sheet 1



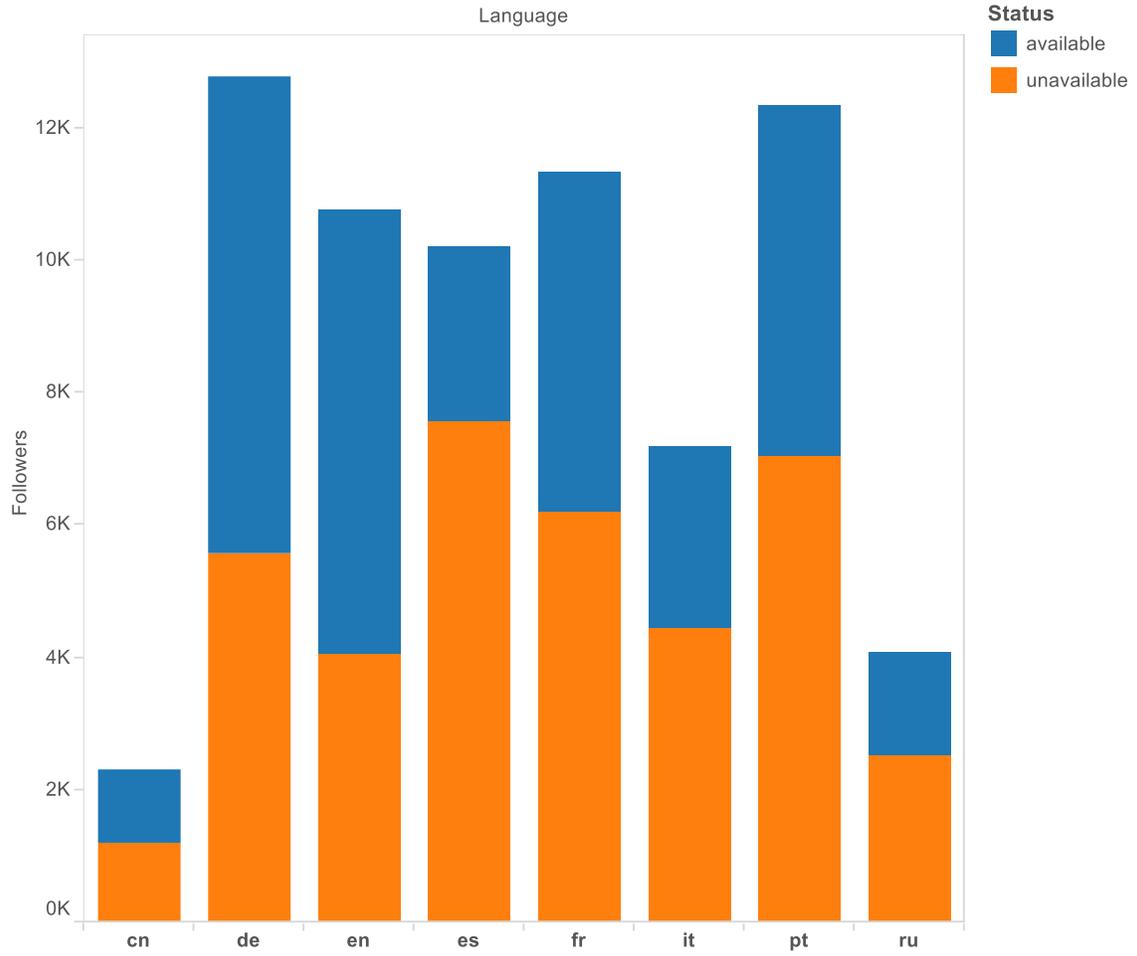
Content Rating and Traffic Source. Color shows details about Content Rating. Size shows sum of Followers. The marks are labeled by Content Rating and Traffic Source.

Sheet 1



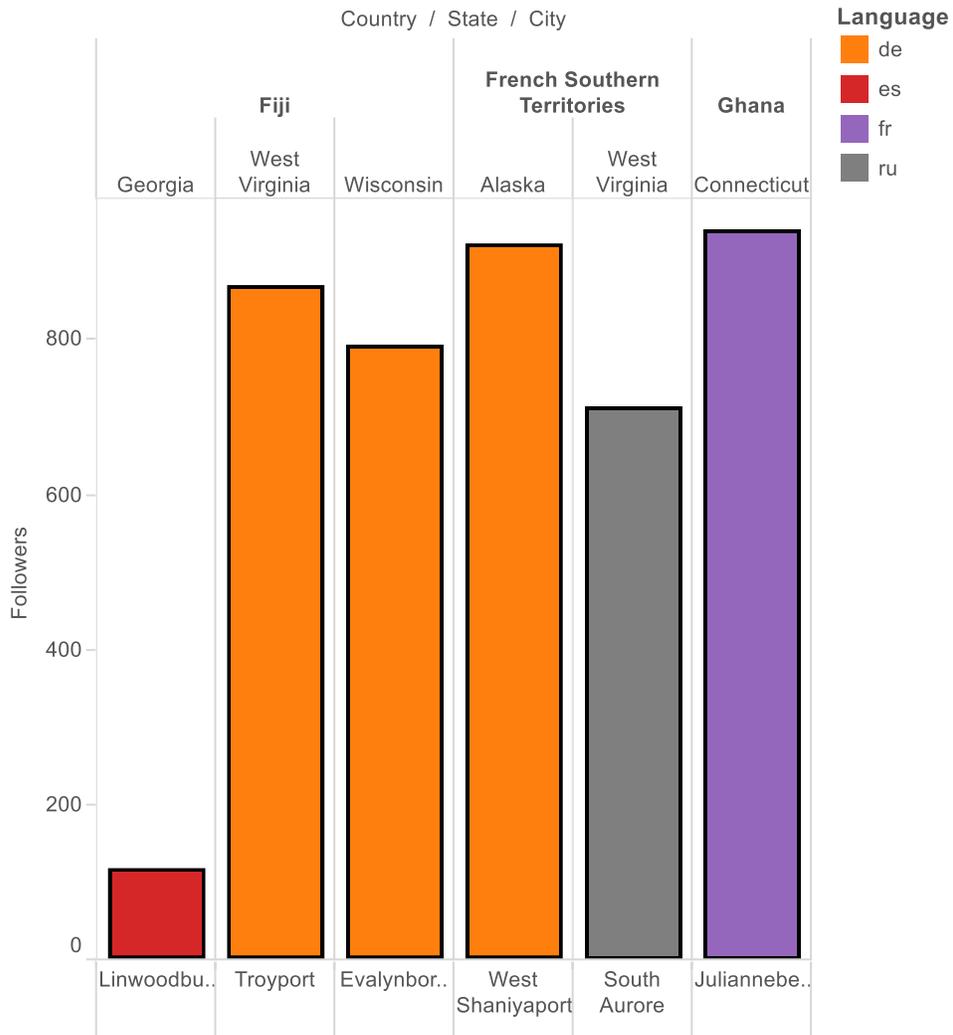
Sum of Followers for each Traffic Source broken down by Status and Content Rating. Color shows details about Traffic Source.

Sheet 1



Sum of Followers for each Language. Color shows details about Status.

Sheet 1



Sum of Followers for each City broken down by Country and State. Color shows details about Language. The view is filtered on Country, which keeps

6. CONCLUSION

Users upload videos on various portals these days, and providing the users with meaningful insights about their videos, or helping a company improve their services or audience reach with help of insights of their videos greatly helps the uploader and the viewer. Companies pay the 3rd party vendors to be able to provide them meaningful information out the large amount of data generated by their uploaded videos. The development of such a service, which has the capability to provide meaningful video traffic analytics by simply providing it with the json schema of the records being generated by the video service company will a provide a total new level of experience which will be enjoyed by all.

7. REFERENCES

<http://www.similarweb.com/website/developer.vimeo.com>

<http://www.alexa.com/siteinfo/Vimeo.com>

<https://www.quantcast.com/vimeo.com>

<http://developer.vimeo.com/api/start>

www.stat.cmu.edu/~cshalizi/ADAfaEPoV/ADAfaEPoV.pdf

Python for Data Analysis - O'Reilly Media

<https://hackhands.com/how-to-get-started-on-the-mean-stack/>

<https://www.tableau.com>

<https://www.npmjs.com/package/casual>

<http://marak.com/faker.js/>