LAB – 8

Submitted By-: Yugaank Arun Sharma

CS 554

Date of Submission-: 11/22/2017

Scenario 1: Logging

To store log entries- I would like to store log records as a file on the file system. It could be distributed cloud and synchronization could take place from time to time.

How can users submit log entries? -: We can use Linux syslog and then sync them with the cloud.

How can users query? -: since we are using file system to store log entries, we can use elastic search plus Logstash to query the records.

How can they see? -: That depends, we can let them login, and use some tools to create table or they can just directly download the log file.

Web server? -: I suppose apache is common these days to use and easy to set up. Plus, it provides better functionality on static resources

Scenario 2: Expense Reports

How will we store expenses? -: We can set up mongo DB to store the data in the respective collection as it doesn't have any relationship with other collections. All data structures to be used are known beforehand.

Web server? -: express could do it I guess. It's easy to set up with mongo DB and could be used with other independent technologies conveniently.

How to handle emails? -: We can use NodeJS package called Nodemailer. It makes sending email in NodeJS easy as cake.

Generate PDF? -: We can use PDFKit. It is a JavaScript PDF generation library for Node and the browser. It allows making complex, multi-page, printable documents easy. Its API allows chain ability, and includes both low level functions as well as abstractions for higher level functionality.

Template? -: Since I've worked with handlebars before, I would use it again.

Scenario 3: A Twitter Streaming Safety Service

API? -: I would use twitter standard search API. It allows you to search for tweets matching the query specified.

How will I build this? -: I can have a separate admin panel though which we can add more functions or update the existing ones.

Make sure it remains stable: By having a tool that monitors the system like run automated tests every day, have backup resources, look for critical messages in logs.

Web Server? -: Web server like we used recently in class. That allows for a worker to run separately and the server that has a message queue.

Database? -: mongo DB, as it is convenient to store document type data in to this database.

Historical log of tweets? -: have a collection that stores the old tweets like an archive. Update them time to time.

Real time, streaming incidents reports? -: store them in a file in the file system and consistently sync them time to time with distributed systems in cloud.

Media? -: Use CDN like Akamai.

Scenario 4: A Mildly Interesting Mobile Application

handle the geospatial nature of your data? -: Use google places API.

How will I store? -: long term- store them in a cloud. Cheap storage provided by likes of Cloudinary etc. Amazon also provides services like this. Short term- use google drive, drop box, retrieve them and store them in memory using Redis.

What would you write your API in? -: we can use REST architecture to build this API. Use firebase or OAuth for authentication and use Independent technologies like ImageMagisk to modify or edit images.

Database -: MySQL would be nice and easy to use here. Store images in cloud and store their link in the tables.