Faraz Pathan

Scenario 1:
We could store the log status by storing the info into a cookie and have the server running Express. Have the log as a string and push it into the array of strings. So anytime a user does an action such as a button or the user goes on a web page, it would be a string added to an array describing what the user did. The array of strings of action of the user will be put in a MongoDB server when the cookie expires, user logs off, or if the user goes to the page to view all activity. This will clear the cookie but put all the info in the MongoDB and display all the actions the user did. The logs can be queried by having a search form to search up specific actions or a filter to filter only button presses or web page visits. The result of the query can be displayed by a table or rendered into the web page.

Scenario 2:
We could have the servers running in python and have mongoDB as the databases. I would store the info into an array of objects and each object has all the data structures that are needed to generate the pdf. I decided to use python as a web server because it would be easier to generate a pdf and email it to the user with python. I would use SMTP_SSL() to send the email. It is easy and simple to send an email. All you need is the login info and it will be able to send a message with very few lines of code. We could also generate a pdf using python fpdf which will be able to create an excel and then convert it to a pdf which is exactly what needs to be done in this project.

Scenario 3
I would use Twitter API V2 as a Api to receive the tweets that contain the trigger words. I would use Apache Spark as the web server to go beyond our local precinct. Spark will give the results fast and stable. I would be able to store the trigger words by putting it in a Redis due to it having relatively low data and it could cause our program to go faster. I would also use ElasticSearch to handle and store the big data such as historical tweets and all the media. I would be able to handle a real time streaming incident report by organizing the trigger word with higher danger possibility and within the same trigger word organize by the time the post was posted.

Scenario 4

I would create a field with the image to store where the image has taken place. This would attach a location to every image and make it easy to find the location of every image. I would use a MySQL database to achieve long term and cheap storage. MySQL is very good at storing large amounts of image data. However, if you want short term and fast retrieval I would use MongoDB. Getting an image would be very fast but it is not very sustainable for a big user base and better in a smaller amount of traffic. I would write my own API using Javascript and I would use Express for the web server.