Scenario 1: Logging

As for a logging server, I will choose MERN to implement it. MERN refer to MongoDB, Express, React and Nodejs. One of collections of MongoDB can be used to store log entries through JSON. When user sign up their first account, I can hash their password and store it as well as lowercase username into MongoDB user collection. Then, I will use Express to create my web server as backend and user can request a specific end of server. When user log in their account and sumbit login form. They will send a request to the server with their password information. Thus, in middleware function, I can check whether their input password is equal to stored hash password with corresponding username. If so, I authorize user getting to corresponding end of server. Otherwise, I will show their error message and redirесct them back to log in page. I can use React as frontend to create html form component for user to input their login information. If they log in successfully into main page, I can show their log in entries on main page through HTML.

Scenario 2: Expense Reports

As to Expense Reports, I will store the expense reports in MongoDB, and when user input their information in the form of reporting web applications. I will check whether the field of "isReimbursed" for that corresponding "Id" is true in middleware function. If "isReimbursed" is true, I will generate a PDF by LaTex to translate form data into PDF format. LaText is effectively the language I can write PDFs in and compile LaText to a PDF. In addition, because every user will input their own email, I can get user input email to send emails by Google Firebase. I can use handlebar library as template to handle all the templating for the web application, since all of the data structure that use input is the same, which I can have same 'key' in template.

Scenario 3: A Twitter Streaming Safety Service

For this scenario, I will use MERN stack with Twitter API v2 and MongoDB. I will create a server through Express and we can use router.get(), router.post(), router.put() and router.patch() for handling CRUD applications with querying specific end of sever and form submissions. I will use "powerTrack API" to query tweets based on keywords and geolocation. I can use Express

middleware function with Regex to query dangerous tweets and report them if there are keywords showing in tweets. Then, I can also use Python Spider and Beautiful Soup library to mine tweets and use Regex to find keyword, and then write script to automatically monitor the contents of the Tweet. If there are keywords caught by regex, python script can automatically send email to inform officers for critical triggers. Meanwhile, I can record this alert message into MongoDB with time, tweets information, who tweets it and investigation status in boolean. Officer can retroactively search through all historical log by MongoDB and see specific data in MongoDBCompass. I will also use Apache Hadoop which can perform big data analysis across the data set. It can help me analyse the tweets that I stored in database and see their threat level, which will update in real time.

Scenario 4: A Mildly Interesting Mobile Application

For this scenario, I will use python, kivy library, tinker library, folium library, GeoPandas library, pillow library, OpenCV library, buildozer and MongoDB. We can use kivy and tinker to make GUI for our mobile application and make a login or signup form by kivy. Then, we store use information into MongoDB. Pillow and OpenCV library can be used to manipulate images. It can help user modify and change pictures. GeoPandas is an open source project to make working with geospatial data in python easier. GeoPandas extends the datatypes used by pandas to allow spatial operations on geometric types. With folium and GeoPandas, we can handle user's geo information and store it with their pictures into MongoDB. Buildozer is a tool that aim to package mobiles application easily. It automates the entire build process, download the prerequisites like python-for-android, Android SDK, NDK, etc. We can use Buildozer to make APK to release our application in Android store for user to download. As for user account, we can add firebase authorization into our application to make every user have their own account.