CS-554 Web Programming
Lab4

Scenario 1 : Logging

1) **How would you store your log entries?**
   To store the log entries, a NoSQL databse like MongoDB can be used. MongoDB is an object-oriented, simple, dynamic, and scalable NoSQL database. It is based on document store model. Advantages of using MongoDB are high performance, high availability and automatic scaling.

2) **How would you allow users to submit log entries?**
   Log entry can be submitted by putting the entire text of the log record into a document.

3) **How would you allow them to query log entries?**
   To query the log entries mongodb can be installed on the NodeJS application as a dependency. To extract the data from the collections, mongodb functions can be used for each specific query.

4) **How would you allow them to see their log entries?**
   To see the log entries, a simple front-end application can be developed where the data can be fetched from the database and displayed in the form of table. This can be done using front-end library like ReactJS.

5) **What would be your web server?**
   I would prefer to create a web server using NodeJS and ExpressJS.

Scenario 2: Expense Reports

1) **How would you store your expenses?**
   I would use MongoDB to store the expenses since it is a document oriented database where different type of data can be stored in the same object using the JSON format.

2) **What web server would you use and why?**
   I would use NodeJS and ExpressJS for the webs server as its easy to configure and customize. One of the main reason for using ExpressJS is it allows you to define routes the application based on HTTP methods and URL. It also includes various middleware modules which can perform additional task on the requests and response.

3) **How would you handle emails?**

To handle emails, modules like Nodemailer and xoauth2 can be used. Nodemailer is the module used for sending emails whereas xoauth2 is used for generating tokens, sending and receiving emails.

4) **How would you handle PDF generation?**
   To handle PDF generation, a package called pdfmake can be used which can be installed using the npm manager.

5) **How are you going to handle all templating for the web applications?**
   Templating for the web application can be done using express-handlebars which is a lightweight templating system for NodeJS. Handlebars allows us to avoid repetitive code by compiling the final DOM structure of our site via logic, typically compiled by task runners like GULP.

Scenario 3: A Twitter Streaming Safety Service

1) **Which Twitter API would you use?**
   I would use Filter Realtime Tweets API as it allows us to filter the tweets using keywords.

2) **How would you build this so its expandable to beyond your local precinct?**
   I would use geolocation package to expand the service beyond my local precinct.

3) **What would you do to make sure the system is constantly stable?**
   To make the system constantly stable, I would make sure code is always clean and there is no malicious or duplicate code being used.

4) **What would be your web server technology?**
   I would use NodeJS and ExpressJS for the webs server as its easy to configure and customize. One of the main reason for using ExpressJS is it allows you to define routes the application based on HTTP methods and URL. It also includes various middleware modules which can perform additional task on the requests and response.

5) **What database would you use for triggers?**
   I would use MongoDB for triggers as it is an object-oriented, simple, dynamic, and scalable NoSQL database.

6) **For historical log of tweets?**
   I would use Redis for historical log of tweets. Redis is an in-memory data structure project implementing a distributed, in-memory key-value database with optional durability.

7) **How would you handle the real time, streaming incident report?**
   I would handle real time, streaming incident report using Web Socket.

8) **How would you handle storing all the media that you have to store as well?**

   I would store all media in the MongoDB database as unlike file systems, this will not have any problem in storing millions of objects.

9) **What web server technology would you use?**
   I would use NodeJS and ExpressJS for the webs server as its easy to configure and customize. One of the main reason for using ExpressJS is it allows you to define routes the application based on HTTP methods and URL. It also includes various middleware modules which can perform additional task on the requests and response.

Scenario 4: A Mildly Interesting Mobile Application

1) **How would you handle geospatial nature of your data?**
   I would use Google Maps API to handle the geospatial nature of the data.

2) **How would you store images, both for long term, cheap storage and for short term, fast retrieval?**
   For long term and cheap storage, images can be stored in cloud services like AWS, Google Cloud Platform etc. and for the short term and fast retrieval, images can be store locally using a MongoDB database.

3) **What would you write your API in?**
   I would write an API in Javascript.

4) **What would be your database?**
   I would use MongoDB for the database. MongoDB is an object-oriented, simple, dynamic, and scalable NoSQL database. It is based on document store model. Advantages of using MongoDB are high performance, high availability and automatic scaling.