CS 554 – WEB PROGRAMMING 2

HOMEWORK 5

Scenario 1: Logging

- **How would you store your log entries?**

  To store the log entries, I will suggest using MongoDB or any NOSQL database. The reason behind this is at some point in future we may need to add new attributes in our log entries for better analysis purpose and a SQL database would be difficult to manage. NOSQL database provides consistency as well as partition tolerance which helps in partitioning large data sets.

- **How would you allow users to submit log entries?**

  Log entries can be submitted by saving a document in NOSQL.

- **How would you allow them to query log entries?**

  For querying them, we can use NodeJS or even .NET. But I would prefer NodeJS, as it is clean, light, and easy to use. To query any log entry, we just write similar command as used in MongoDB.

- **How would you allow them to see their log entries?**

  To view the log entry, we can create a simple frontend to display all fields of log using AngularJS or ReactJS.

- **What would be your web server?**

  NodeJS and ExpressJS

Scenario 2: Expense Reports

- **How would you store your expenses?**

  To store the expenses**,** I will use MongoDB here so that I can get a clean JSON data which is easy to read and handle.

- **What web server would you choose, and why?**

  I will use ExpressJS as it is very helpful in routing to different parts of the web application. It also supports middleware to handle incoming data.

- **How would you handle the emails?**

  We can use “emailjs” for sending email and “node-imap” for receiving emails.

- **How would you handle the PDF generation?**

  For this we can use “pdf-creator-node” npm package.

- **How are you going to handle all the templating for the web application?**

  To handle all the templating there are many templating engines provided by NodeJS. For example – Handlebars, PUG, EJS. I suggest using handlebars – “express – handlebars” module

Scenario 3: A Twitter Streaming Safety Service

- **Which Twitter API do you use?**

  I will use Filter Realtime tweets – PowerTrack API as it gives all fields related to the tweet as the scenario given.

- **How would you build this so its expandable to beyond your local precinct?**

  I would use Cloud for storage and security for extended build and would have dedicated servers.

- **What would you do to make sure that this system is constantly stable?**

  To make the system constantly stable, I would make sure code is always clean, is less complex and do not have duplicate code as it might slow down the running time.

- **What would be your web server technology?**

  I will use ExpressJS as it is very helpful in routing to different parts of the web application. It also supports middleware to handle incoming data.

- **What databases would you use for triggers?**

  I will use Database trigger for MongoDB as database.

- **For the historical log of tweets?**

It requires each tweet to be cached so I would use Redis as it is an in-memory data structure. MemCached is also suggested.

- **How would you handle the real time, streaming incident report?**
  I will use Socket.io as it is much compatible with NodeJS.

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

  I would use MongoDB because it has a feature to handle all types of media data which is called GridFS.

- **What web server technology would you use?**

  I will use ExpressJS as it is very helpful in routing to different parts of the web application. It also supports middleware to handle incoming data.

Scenario 4: A Mildly Interesting Mobile Application

- **How would you handle the geospatial nature of your data?**

  I would use Google Maps API, as it is very easy to use and reliable.

- **How would you store images, both for long term, cheap storage and for short term, fast retrieval?**

  I would use GridFS functionality of MongoDB to store images as it stores it in binary format which saves a lot of memory.
  And for long term I would use AWS services.

- **What would you write your API in?**

  I would write it in NodeJS

- **What would be your database?**

  It would be NOSQL database (MongoDB)