CS-554 Lab 8

Scenario 1: Logging:

- **For this scenario, I prefer to use MongoDB, Node JS, Express, HTML5, CSS3, BootStrap4. So Once HTML form is ready then we set it to post data using AJAX request and then data is stored into MongoDB using POST API of Express Server and then we handle operations for querying data using (PUT, GET, DELETE) and then set appropriate results into HTML Format.**

1. How would you store your log entries?

- We store log entries using MongoDB in the form of JSON document.

2. How would you allow users to submit log entries?

- In Node JS, we can set JQuery-Ajax request for form submission and then using POST API, we submit log entries and then data module store into Mongo DB Database.

3. How would you allow them to query log entries?

- In MongoDB CRUD operations with Express Server are used to query log entries so that we can manage POST, DELETE, GET, PUT requests.

4. How would you allow them to see their log entries?

- In Node JS, Get the data from MongoDB using GET API of Express Server and set it in HTML format or preferable format of different frameworks.

5. What would be your web server?

- Express Web Server.

Scenario 2: Expense Reports:

- **For this scenario, I prefer to use HTML5, CSS3, MySQL, Latex, PHP, Apache Server. So Once HTML form is ready then we insert data using AJAX request and then data is stored into MySQL. IF field isReimbursed is set to true then It will generate PDF using Latex and then mail to user using mail () function in PHP.**

1. How would you store your expenses?

- I will use MySQL to store data with fix fields into database.

2. What web server would you choose, and why?

- I will prefer XAMP-Apache server with MySQL Database. MySQL database works comfortably with apache server.

3. How would you handle the emails?

- I will use mail () function in PHP to handle emails.

4. How would you handle the PDF generation?

- For the PDF generation, I will use Latex which will convert the HTML content into PDF.

5. How are you going to handle all the templating for the web application?

- For templating of PDF, I will prefer to use Latex.

Scenario 3: A Twitter Streaming Safety Service:

- **For this scenario, I prefer to use MongoDB, Node JS, Express, HTML5, CSS3, BootStrap4, Redis pub-sub for trigger, Socket.io for streaming of data, Elastic Search to handle request fast.**

1. Which Twitter API do you use?

- We can use Twitter Public streaming API.

2. How would you build this so its expandable to beyond your local precinct?

- We can build system in such that we can modify the triggers to be used accordingly for other precinct, so we can use master slave concept to change something that affects to other data.

3. What would you do to make sure that this system is constantly stable?

- We have mainly three functions
  - Collect and Analyze Tweets
  - Email Alert
  - Backend

  So Here we can use daemon to monitor this script so that stability can be maintained. We can also maintain error_log and status log.

4. What would be your web server technology? What databases would you use for triggers?

- Express Web Server would be preferable and Redis pub-sub for triggers and as a storage MongoDB is used to store data in JSON Format.

5. For the historical log of tweets.

- Elasticsearch is used for historical log of tweets.

6. How would you handle the real time, streaming incident report?

- We can use Socket.io for streaming reports real time and MapReduce to handle real time streaming incidents.

7. How would you handle storing all the media that you have to store as well?

- AWS would be used to store media. We can use CDN to access media speedly.

Scenario 4: A Mildly Interesting Mobile Application:

1. How would you handle the geospatial nature of your data?

- First, we can use google location API and then tag our data. For increasing speed, we can use CDN. There are also different ways to find current location using IBMS and then do sorting using elastic search and get appropriate data.

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

- For short term and Fast retrieval, I prefer to use Mongo DB GridFS as it allow to store Binary data in BSON format. For Cheap storage and Long term, I prefer to use Amazon S3 as It gives a fast, scalable and cheap option to store data.

3. What would you write your API in?

- Django can be useful to write API in server side of mobile application.

4. What would be your database?

- I would like to use MongoDB Grid FS.