

LAB – 8: Choosing Technical Stack

Scenario 1: Logging

Server:

I will be using **Express Server** in Node as a backend server to allow user to submit log on remote server.

Database:

I will be using **MongoDB or any NoSQL** database because as per requirement log will have some common fields, but it also support any number of customizable fields for an individual log entry, So storing data as JSON format with user as collection.

For example:

USER1 Collection:

```
{
  Name: username,
  Password: hashPwd
  Log: [
    {
      TIMESTAMP: Data&Time,
      SomeCommonAttribute: Log
      customizableAttribute: Log
    },
    {
      TIMESTAMP: Data&Time,
      SomeCommonAttribute: Log
      customizableAttribute: Log
    }
  ]
}
```

Each time new log appends the log array with timestamp.

Front-end Technology:

I would prefer ReactJS for this because log data continues submit through user experience from worldwide. So, Admin can monitor runtime data and QA team will analyze each and every error occur to user and test team can forward particular issue to assign team.

Admin can see the statistics of the user error by querying in database data.

Scenario 2: Expense Reports

Web Server:

I would like to use Apache server because for this scenario the data field is fixed so I would prefer to use MySQL database, as per my knowledge MySQL works fine with apache server.

This application is related to finance so we have to assign one buffer for the database operation.

Suppose your current balance is \$100 and at same time you and your friend spend \$20 from New York City and, Hoboken respectively.

Then initially application take balance \$100 for both and it reduce \$20 from main balance so it remains \$80 for both and both update the database of current balance with \$80 instead of \$60.

To overcome this problem, we have to use common buffer for database operation so only one operation will do at one time.

For PDF Generation, we suppose to use master slave concept with the help of publish subscribe methodology, Main web server request LaTeX worker to generate PDF file of particular template with given data. Worker obey the command of webserver and after generating PDF worker publish message to webserver that I put file in this folder with file name XXX.pdf after that web server takes this files and send to requester over HTTP protocol.

For Email, Php language has already inbuilt function call mail() to send emails.

Database:

I would prefer to use MySQL database because field is fixed and this is related to finance and with the help of MySQL database we can provide any additional facilities also like what are highest expense for first quarter and like any other.

Front-end Side:

I would like to prefer AngularJS in frontend side because It give us display live data through runtime AJAX method.

Scenario 3: A Twitter Streaming Safety Service

Web Server:

I would like to use Express Server from NodeJS because it will handle multiple request at same time.

I would like to use master with multiple slave's methods. There is one server running for entire country and each slave server running for each city.

Master server provide CRUD operation and slave will use that services. For instance each slave server have python script to scan the twitter data for particular city through twitter APIs: <https://developer.twitter.com/en/docs/api-reference-index> and scan for fight, drug and many more word search. Each slave server maintains history database in MongoDB which contains history of tweets.

If slave server found something bad in assigned city than it triggers to master server with dangerous level assign through machine learning algorithm through CRUD operation of master slave.

Then master slave has database of all officers with city assigned and designation by priority. So after slave server post request for something bad happen then master server identify the all officers related to this act and send mail using mail() function and/or it will send text message through Way2SMS API to all related officer.

Database:

Master server has MongoDB database which contains all police officers information like duty city, designation and assigned dangerous level.

Each city slave server has MongoDB database which keep tracks of every tweet in the particular city and also maintains the history of tweets. History of tweets uses for machine learning training data to identify level of serious incidence.

Scenario 4: A Mildly Interesting Mobile Application

How would you handle the geospatial nature of your data?

Firstly, I will take users current location by Google API, Wifi Location and IP location based on the location we will identify the area and map with our data.

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

Long term/Cheap storage: I would like to use online storage like Amazon AWS, Microsoft Azure. This gives a fast, scalable, and cheap storage option for our images.

Short Term/Fast retrieval: I would like to maintain a cache of images on the user's device for fast retrieval. A lot of popular mobile applications do this, and it helps to serve content faster than requesting from the database.

What would you write your API in?

We write API to upload a picture with raw data through HTTP protocol, location data from multiple sources like Wi-Fi location, IP address and mobile location services in NodeJS server.

What would be your database?

I would like to use relational database to store users information and events information, and store photos into Amazon AWS, or Microsoft Azure, and just we have to insert access information or link in database.