

# CS554 Web Programming2: Lab5

Mijeong Ban

I pledge my honor that I have abided by the Stevens Honor System.

## Choosing Technical Stacks

For this lab, you will not be programming, but rather analyzing several scenarios and determining what technologies would work well together to solve these problems. For each answer, back up your reasoning with why you chose each technology, strategy, etc.

### Scenario 1: Logging

In this scenario, you are tasked with creating a logging server for any number of other arbitrary pieces of technologies.

Your logs should have some common fields, but support any number of customizable fields for an individual log entry. You should be able to effectively query them based on any of these fields.

#### Answer

I would use Amazon S3(Simple Storage Service) to store log entries wit. The service maximizes benefits of scale and passes those benefits on to developers, meaning it will be beneficial to store the huge amount of log entries.

Amazon S3 has the feature for server access logging which provides detailed records for the requests that are made to a bucket. This feature also includes enabling logs for request and querying logs for requests. Also, it allow us to customize access log information with various fields such as Time, Remote IP, Request ID, HTTP status, Error code and so on.

### Scenario 2: Expense Reports

In this scenario, you are tasked with making an expense reporting web application. Users should be able to submit expenses, which are always of the same data structure: `id`, `user`, `isReimbursed`, `reimbursedBy`, `submittedOn`, `paidOn`, and `amount`.

When an expense is reimbursed you will generate a PDF and email it to the user who submitted the expense.

#### Answer

- How would you store your expenses?
  - I would use mongoDB because it will be only one collection with the same data entries.

- What web server would you choose, and why?
  - I would use Express to manage a server and routes because I am making a simple web service. I only need to have few routes to handle user's request to submit expenses.
- How would you handle the emails?
  - I would use Gmail API to handle the emails since it allows us to create draft emails and send email to the user. Also The Gmail API allows to upload attachment as well which will be a PDF file in this scenario.
- How would you handle the PDF generation? How are you going to handle all the templating for the web application?
  - I would use PDF Generator API(link)[<https://www.programmableweb.com/api/pdf-generator>] since it provides templates that can be merged with data from REST API to create PDF documents.

### Scenario 3: A Twitter Streaming Safety Service

In this scenario, you are tasked with creating a service for your local Police Department that keeps track of Tweets within your area and scans for keywords to trigger an investigation.

This application comes with several parts:

- An online website to CRUD combinations of keywords to add to your trigger. For example, it would alert when a tweet contains the words (fight or drugs) AND (SmallTown USA HS or SMUGS).
- An email alerting system to alert different officers depending on the contents of the Tweet, who tweeted it, etc.
- A text alert system to inform officers for critical triggers (triggers that meet a combination that is marked as extremely important to note).
- A historical database to view possible incidents (tweets that triggered an alert) and to mark its investigation status.
- A historical log of all tweets to retroactively search through.
- A streaming, online incident report. This would allow you to see tweets as they are parsed and see their threat level. This updates in real time.
- A long term storage of all the media used by any tweets in your area (pictures, snapshots of the URL, etc).

#### Answer

There is a PowerTrack API which allows to filter the full Twitter firehose to receive the only data(tweets which contains trigger words) that we want to get. We can set various attributes including geo-location and language, which means it can be used beyond the specific location. This API also covers the real-time updating feature to stream incident reports. For the text alerting system, there is the service called plivo to quickly send updates to users(officers). For the email alerting system, we can use Gmail API in this scenario as well. MongoDB will be also useful in this case since it should be easily searchable and status-changable

through the historical tweets. Last, I would S3 bucket for the long term storage to store media because we can upload any number of objects to the bucket and easily manage them.

#### **Scenario 4: A Mildly Interesting Mobile Application**

In this scenario, you are tasked with creating the web server side for a mobile application allows users to see mildly interesting pictures in their geographical location.

Users must have an account to use this service. Your backend will effectively amount to an API and a storage solution for CRUD users, CRUD ‘interesting events’, as well as an administrative dashboard for managing content.

##### **Answer**

For geographical location, I would store the sets of longitude and latitude in MongoDB database to easily access to them. For the storage part, Amazon S3 bucket would be the best option since it is a highly-scalable and secure solution. This option is perfect for general purpose files that are frequently accessed, for files that are required to be archived for the long-term but are rarely accessed. Also, it is extremely secure because Amazon S3 can be used alone.