Name – Arinjay Panwar

CWID – 20013410

Course – CS 554

Lab 3

**Scenario 1: Logging**

**Q. How would you store your log entries?**

I would store log entries in MongoDB as documents with some required fields like, timestamp, source, text and then the customizable fields. MongoDB being a NoSQL database would be best for customizable fields.

**Q. How would you allow users to submit log entries?**

User would submit log entries via POST request to an API endpoint api/logs which can be accessed via HTML form or API request. The API can be secured with authentication and authorization mechanisms to prevent unauthorized access.

**Q. How would you allow them to query log entries?**

To allow users to query log entries, I would use a search engine such as Elasticsearch. Elasticsearch is an advanced indexing platform that focuses on searching, rather than storage.

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

To allow users to see their log entries, I would use a web-based dashboard Kibana. These dashboards can be configured to display log entries in various formats such as tables, charts, and maps.

**Q. What would be your web server?**

I would use Node JS web server framework as it will suffice this requirement.

**Scenario 2: Expense Reports**

**Q. How would you store your expenses?**

I would store the expenses as records in a SQL database i.e. a relational database, where the table Expenses would look like

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Id** | **User** | **IsReimbursed** | **ReimbursedBy** | **SubmittedOn** | **PaidOn** | **Amount** |
|  |  |  |  |  |  |  |

**Q. What web server would you choose, and why?**

I would choose Django web server as it provides easy integration with SQL databases with its models framework. They also offer powerful templating engines to render HTML templates.

**Q. How would you handle the emails?**

To handle the emails, I would use a third-party email service such as SendGrid. These services offer easy integration with web applications and provide features such as email templates, tracking, and analytics.

**Q. How would you handle the PDF generation?**

I would use wkhtmltopdf. It is a library that uses Webkit to take HTML and print it to PDF

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

Django provides handling of HTML templates via its templates framework which is another reason to use Django. HTML templates can be stored in a folder and loaded via the Django template function.

**Scenario 3: A Twitter Streaming Safety Service**

**Q. Which Twitter API do you use?**

I would use the Twitter Streaming API to collect and process tweets in real-time. This API provides a continuous stream of tweets that can be filtered by keywords, geolocation, and other criteria. It also allows for real-time processing and analysis of tweets.

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

I would design the system to be modular and scalable. This would involve separating the different components of the system into microservices that can be independently deployed and scaled. This would also involve using cloud-based services for storage and processing to ensure high availability and scalability.

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

To make sure the system is constantly stable, I would implement a combination of monitoring and testing strategies. For monitoring, I would use tools such as Nagios to monitor the system's performance, availability, and errors. For testing, I would use a combination of unit tests, integration tests, and load tests to ensure the system is functioning as expected and can handle high volumes of data and traffic.

**Q. What would be your web server technology?**

I would use Django web server. These frameworks provide built-in support for handling HTTP requests, user authentication, and database interactions. They also offer powerful templating engines to render HTML templates.

**Q. What databases would you use for triggers?**

I would use influxdb for triggers as it is highly recommended for real-time data like sensor data, or time series data.

**Q. For the historical log of tweets?**

MongoDB as the twitter tweets are non structured and not fixed in size. Moreover we would need related tweets and comments for some tweets as well which is why MongoDB is suitable for this.

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

I would use Django celery beat a background job scheduler and task queue which would run jobs and triggers to notify about the incidents.

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

I would something like Amazon S3 or Google Cloud Storage to store the media on the cloud for fast and cheap storage in with regards to the volume of media that would be generated.

**Q. What web server technology would you use?**

I would use Django as it would integrate with lot of third-party libraries easily which would help in accomplishing this task easily.

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

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

To handle the geospatial nature of the data, I would use a geospatial database such as MongoDB with PostGIS extension. These databases support geospatial indexing and querying, allowing for efficient retrieval of interesting events within a given geographical location.

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

I would something like Amazon S3 or Google Cloud Storage to store the media on the cloud for fast and cheap storage for long term. For short term, fast retrieval, I would use a content delivery network (CDN) such as Cloudflare or Akamai. CDNs cache images at edge locations around the world, allowing for fast retrieval by users wherever they are.

**Q. What would you write your API in?**

I would write my API in Express JS as it would integrate easily with MongoDB and as we need a Mobile application, we can even use JavaScript framework for mobile application development like React Native or Ionic which makes the development easier and fast.

**Q. What would be your database?**

I would use MongoDB for the geospatial nature of the database, it can also be used to store media files in GridFS if needed. It is easy to integrate mongoDB with Express JS.