Instructions:

**Submission guidelines:**

* To be done individually or in a group of maximum 2 students.
* **Due Date:**Sunday, December 24 EOD (11:59:59PM). No extensions. Delayed submissions will NOT be graded.
* Pick ONE of the two questions – only one question to be submitted.
* Only one submission per group if submitting from a 2 person group.
* Using Canvas, only the first member of the group has to upload the file. No submission over email will be considered.
* Submit the code and a document as a zip file. The document should be a PDF describing your setup, how to run your code and results.
* The document in the zip file should have full names of the group members along with the BITS Registration no. of each group member.
* Make sure that you upload the file well ahead of the deadline. At the last moment, we have seen several groups have faced issues while doing the submissions.
* If doing as a group, only one submission is expected from each group. Unnecessarily don’t upload the solution on individual basis. If it’s observed, then the penalty of 50% will be set.
* Plagiarism will be strictly dealt with and if found will result in cancellation of the Assignment and 0 marks being awarded to all the group members.
* The last date of submission will not be extended in any case.

Option 1: Spotify Recommendations

**Dataset:**<https://www.kaggle.com/datasets/thedevastator/spotify-tracks-genre-dataset>

This dataset provides comprehensive information about Spotify tracks encompassing a diverse collection of 125 genres. It has been compiled and cleaned using Spotify's Web API and Python. Presented in CSV format, this dataset is easily accessible and amenable to analysis. The dataset comprises multiple columns, each representing distinctive audio features associated with individual tracks.

**Problem: Spotify Music Recommendation Platform**

With multiple audio features such as danceability, energy, and valence, this dataset can be used to build a music recommendation system. By analyzing the preferences of users for certain genres and characteristics of tracks, the system can suggest similar tracks or even recommend new genres that users might enjoy.

**End-User experience:**As a user searches or plays a particular song in Spotify, your Music Recommendation Platform will be able to return other music recommendations for the user. And if the user selects or ignores the recommendation, that data is fed back to improve the recommendation algorithm.

**Requirements:**

1/ **Design:** Design the backend services, APIs, data models and the databases you will use for this. The design should support both realtime stream processing and batch analytics use cases. Explain the rationale for your design especially tradeoffs wrt CAP theorem, describe the tech choices for the databases and other components. You must use a NoSQL database as part of your design (MongoDB, Neo4j, DynamoDb, HBase, anything appropriate for your solution).

**Solution notes:**Submission for this part MUST include an Architecture diagram (you can use excalidraw) with bullet points or a brief description of the design choices, rationale, tech choices. No more than 2 pages.

2/ **OLTP Queries:** Execute at least 3 key read/write queries against the NoSQL database as part of the solution with the optimizations and the consistency levels set in your queries.

**Solution notes:**Submission for this part MUST include a pdf of the queries and a video of their successful execution (you can use loom for the video). No more than 2 pages for the queries and no more than 1 minute for the loom video.

3/ **OLAP Analytics Queries:**Within the broader context of this problem (Spotify) you need to perform analytics queries from the dataset to driver better Sales / Growth. For this, setup a big data batch processing environment (Hadoop or a Spark), share the config. Then perform at least 3 queries in the form of Map-Reduce jobs or Pig / HiveQL / SparkSQL queries and share the responses.

**Solution notes:**Submission for this part MUST include a pdf of the environment configuration, the queries and a video of their successful execution (you can use loom for the video). No more than 2 pages for the config and queries and no more than 1 minute for the loom video.

**\* Include links to the code in github and the loom video links in the pdf.**

**\* Make sure that any included links are publicly accessible. Any link that is broken will not be evaluated.**

**Notes:**Use appropriate assumptions for this problem to increase or decrease the scope. No further clarifications will be provided on the problem.

Option 2: Netflix Ads Platform

**Dataset:**[**https://www.kaggle.com/datasets/thedevastator/netflix-imdb-scores**](https://www.kaggle.com/datasets/thedevastator/netflix-imdb-scores)

The Netflix TV Shows and Movies dataset provides comprehensive information about various titles available on the popular streaming platform. The dataset includes details such as the title's name, its type (whether it is a TV show or a movie), a brief description of the content, the year it was released, age certification rating, runtime (for TV shows: length of episodes; for movies: duration), IMDb score, and IMDb votes.

**Problem: Netflix Ads Platform** (See: <https://help.netflix.com/en/node/126831/us)>

Netflix movies and shows will play ads in between for a lower-price-tier users. Analyzing the distribution of IMDB scores and ratings for TV shows and movies on Netflix can help identify trends and patterns in audience preferences. Netflix Ads platform can use this data to get advertisers to bid for ads and serve the relevant ads to their viewers in their lower-price-tier subscription.

**End-User experience:**As a user is watching the shows from the Netflix catalog, they get served relevant ads, sometimes with choices between more than 1 and the ability to skip the ad after a few seconds. The user behavior wrt watching / skipping the ad is all tracked, that data is fed back to improve the ads recommendation algorithm.

**Requirements:**

1/ **Design:** Design the backend services, APIs, data models and the databases you will use for this. The design should support both realtime stream processing and batch analytics use cases. Explain the rationale for your design especially tradeoffs wrt CAP theorem, describe the tech choices for the databases and other components.You must use a NoSQL database as part of your design (MongoDB, Neo4j, DynamoDb, HBase, anything appropriate for your solution).

**Solution notes:**Submission for this part MUST include an Architecture diagram (you can use excalidraw) with bullet points or a brief description of the design choices, rationale, tech choices. No more than 2 pages.

2/ **OLTP Queries:** Execute at least 3 key read/write queries against the NoSQL database as part of the solution with the optimizations and the consistency levels set in your queries.

**Solution notes:**Submission for this part MUST include a pdf of the queries and a video of their successful execution (you can use loom for the video). No more than 2 pages for the queries and no more than 1 minute for the loom video.

3/ **OLAP Analytics Queries:**Within the broader context of this problem (Netflix Ads Platform) you need to perform analytics queries from the dataset to driver better Sales / Growth. For this, setup a big data batch processing environment (Hadoop or a Spark), share the config. Then perform at least 3 queries in the form of Map-Reduce jobs or Pig / HiveQL / SparkSQL queries and share the responses.

**Solution notes:**Submission for this part MUST include a pdf of the environment configuration, the queries and a video of their successful execution (you can use loom for the video). No more than 2 pages for the config and queries and no more than 1 minute for the loom video.

**\* Include links to the code in github and the loom video links in the pdf.**

**\* Make sure that any included links are publicly accessible. Any link that is broken will not be evaluated.**

**Notes:**Use appropriate assumptions for this problem to increase or decrease the scope. No further clarifications will be provided on the problem.