# Project Plan

By – Agampreet Singh Sihota (22204258) & Kushagra Mittal (22206254)

## Q1: Briefly introduce the dataset of your project. What are the big data challenges in terms of volume and variety?

The Spotify dataset (1921-2020) is a massive collection of music data spanning nearly a century. It includes details like song popularity, tempo, energy, danceability, and more.

Big Data Challenges:

- Volume: With millions of records, handling and processing this dataset efficiently is a challenge.
- Variety: The dataset contains different types of information (songs, artists, genres, years), which makes analysis more complex.
- Scalability: If we wanted to analyze real-time trends, handling streaming data would be another hurdle.

## Q2: What is the main goal of your analysis?

The goal is to uncover music trends over the years, how music characteristics (like tempo, energy, and danceability) have evolved over time.

Why does this matter?

- Helps understand shifts in listener preferences.
- Provides insights into how different genres have changed.
- Can be useful for musicians, producers, and streaming platforms to predict trends.

## Q3: What are the specific tasks needed to achieve this goal?

To analyze trends, we need to break the problem into smaller, manageable tasks:

1. Yearly Trends: Find the average tempo, energy, danceability, and popularity for each year.
   - This helps identify how music features have changed over time.
   - (This is a simple task, computationally light.)

2. Genre Evolution: Track how different genres have evolved by analyzing their characteristics decade by decade.
  - This will need some grouping and filtering.
  - (Moderate computational cost.)

3. Top Artists Over Time: Identify the most popular artists in each decade.
  - Useful for seeing which artists dominated and when.
  - (Sorting large datasets can be computationally expensive.)

4. Correlation Between Features: Check if features like tempo and danceability are related.
  - This helps understand how certain song attributes impact each other.
  - (Requires statistical analysis, making it one of the most computation-heavy tasks.)

## Q4: How can Big Data Technologies help improve performance?

Since some of these tasks involve large-scale computations, using **Big Data technologies** can make processing faster and more efficient:

- Apache Spark:
  - SparkSQL can efficiently query large datasets.
  - MLlib can handle correlation calculations.
  - Caching data (RDD caching) speeds up repeated queries.

- Hadoop & MapReduce:
  - Useful for distributing heavy computations across multiple machines.
  - HDFS ensures scalable storage.

- Optimizations:
  - Using Parquet (instead of CSV) for faster reading/writing.
  - Creating indexes for frequently used columns like 'year' and 'genre' for quicker lookups.

These optimizations can make handling a massive dataset like this one much smoother.