TEAM PROJECT KICKOFF: "ADELE MUSIC SEARCH ENGINE"

Project Overview:

We're building a Java-based data processing system with a built-in search engine, using Adele's discography as our dataset. Our app will let users upload, search, modify, and explore Adele's songs with advanced keyword-based search.

Dataset Theme:

- Topic: Music
- Genre: Adele (songs across albums like 21, 25, 30)
- Each record = 1 Adele song
- Fields:

  🎤 Title (unique key)

  🎤 Album

  🗓️ Year

  🎼 Lyrics (main searchable field)

📚 Core Classes:

- Song (holds song data)
- HashTable (stores songs by title)
- WordID (maps unique words to IDs)
- BST (used for inverted index structure)
- SearchEngine (builds ArrayList<BST>)
- UIHandler (menu options & interaction)

🔍 Search Engine Features:

✅ Remove stopwords (like "the", "and", "to")

✅ Assign integer ID to each keyword

✅ Insert each Song into the right BST inside an ArrayList[wordID]

✅ Search by keyword to list matching song titles

🛠 Menu Options:

[1] Upload new song

[2] Delete a song

[3] Search menu:

- by title
- by keyword

[4] Update a song record

[5] View song stats

[6] Save & Quit

📊 Required Structures:

- HashTable<Song> with separate chaining
- ArrayList<BST<Song>> as inverted index
- HashTable<String, Integer> for word-to-ID mapping

🧩 UML Needed:

We'll create a diagram showing:

- Classes, variables, methods

- Arrows for inheritance, composition, associations

👬 Team Roles (to discuss today):

1️⃣ Coordinator: Leads, integrates, testing

2️⃣ HashTable Dev: Insert, delete, search

3️⃣ Search Dev #1: WordID map + index build

4️⃣ Search Dev #2: BST insertion + keyword logic

5️⃣ UI & File Dev: Menu system, file reading/writing

6️⃣ (Optional) Stats + Slides: For final demo & 3 statistics

🎬 Goal:

A fully functional Adele music search system with a polished video demo and UML class diagram by June 21st!

Let's lock in our roles, brainstorm any features we want to add, and get this done!