**Assignment 1: Problem Statement:**

You are tasked with creating a console-based **Music Playlist Manager** that allows users to:

* Play next and previous songs.
* Add or remove songs.
* View all songs.
* Exit the playlist manager.

The playlist should maintain the order of songs and allow bi-directional navigation using a ListIterator.

**Tasks:**

**Task 1: Setup the Playlist and Add Default Songs**

* Create a LinkedList<String> named playlist.
* Add 4–5 default song names to it .

**Task 2: Display a Menu with Options**

Display the following menu repeatedly until the user chooses to exit

Playlist Menu:

0 - Show menu options

1 - Play next song

2 - Play previous song

3 - Add a song

4 - Remove a song

5 - Show all songs

6 - Exit playlist

**Task 3: Implement Forward Navigation**

* Use a ListIterator<String> to move forward in the list.
* Handle edge cases like reaching the end of the playlist.

**Task 4: Implement Backward Navigation**

* Use iterator.previous() to go back to the previous song.
* Handle the case when the start of the playlist is reached.

**Task 5: Add a Song**

* Ask the user to enter the name of a new song.
* Add the song to the end of the playlist.

**Task 6: Remove a Song**

* Ask the user to enter the song name to remove.
* Use LinkedList.remove(Object) to delete it.
* Reset the iterator after modification.

**Task 7: Display All Songs**

Print all the songs currently in the playlist in order.

**Task 8: Exit Option**

Allow the user to exit the program gracefully.

**JDBC CRUD Application for Product Management Using Layered Architecture + Mockito Testing**

**Objective**

Design and implement a Java application using **JDBC** to perform CRUD (Create, Read, Update, Delete) operations on a product table in a **PostgreSQL** (or MySQL) database, following **layered architecture**, and apply **Mockito-based unit testing** for the ProductService layer.

**Architecture Requirement**

Your application must strictly follow **3-layered architecture**:

1. **Repository Layer (ProductRepository)**
2. **Service Layer (ProductService)**
3. **Controller Layer (ProductController)**

**Features to Implement**

* Add New Product
* View All Products
* Search Product by ID
* Update Product by ID
* Delete Product by ID