Final Project Documentation – Music Player

Project Title:

Mini Spotify – Music Player Application

Project Description:

The goal of this project is to create a fully functional graphical user interface (GUI) music player application inspired by Spotify. This application enables users to view a list of songs, search through their music library, play and pause songs, skip forward and backward through the playback history, and favorite songs to create a custom playlist. The graphical interface is built using Java Swing, and all data structures used (except Map) have been implemented from scratch.

Justification for Data Structures:

1. MyArrayList<Song>:

- **Use**: Stores the main music library.
- **Why:** Efficient access to elements by index (O(1)), great for populating the song list UI quickly.

2. Stack (RecentPlays):

- **Use**: Keeps track of the previously played songs (backward functionality).
- **Why**: Stack allows pushing and popping songs in O(1) time, making it efficient for replaying recently played tracks.

3. Queue (PlayerQueue):

- **Use**: Manages the queue for songs played after the current one (forward functionality).
- **Why**: Queue enables O(1) enqueue and dequeue operations, fitting naturally with song navigation.

4. MySet<Song>:

- **Use**: Stores favorite songs to ensure uniqueness.
- **Why**: Set prevents duplicate entries and offers quick lookup times (O(1)-O(log n) based on implementation).

5. MyMap<String, Playlist>:

- **Use**: Manages playlists (currently only Favorites).
- **Why**: Allows mapping playlist names to their song collections for fast retrieval.

Classes Implemented:

- **MusicAppGUI** – GUI logic and event handling
- **Song** – Stores song title and artist
- **Main** – Launches the application
- **MyArrayList** – Custom dynamic array
- **MySet** – Custom set using MyArrayList
- **MyMap** – Custom map using MyArrayList of key-value pairs
- **Playlist** – Handles logic for favorite playlists
- **MusicLibrary** – Stores initial list of songs (helper class)
- **Stack (RecentPlays)** – Custom stack for back-navigation
- **Queue (PlayerQueue)** – Custom queue for forward-navigation
- **Node / SNode / TreeNode** – Helper classes for data structures

Functionality:

- View all songs in a scrollable list
- Click a song to play it
- Use play/pause toggle
- Favorite songs using the star button (⭐)
- View Favorites as a playlist and return back to the main library

Error Handling:

- Checks for null songs
- Handles empty history or queue when skipping
- Avoids adding duplicate songs to Favorites

User Interface:

- Built using Java Swing
- Modern dark-themed UI with stylized buttons
- Favorites section integrated into the playlist list

Work Log:

- Day 1: Thought of an Idea and started planning
- Day 2: Finished writing the code for the main UI look and created classes for the project
- Day 3: Finish writing all the classes but decided I didnt want to use some of them
- Day 4: Started writing documentation for the project.