**MOBILE APPLICATION DEVELOPMENT**

**PROJECT REPORT**

**Name: Ammar Ali**

**CMS-ID: 023-19-0125**

**Section:B**

**Introduction**

The purpose of this project is to create a media player using Android's MediaPlayer class. The media player allows users to select a song from their device, play/pause the song, and seek through the song using a seek bar.

The code for this project is written in Kotlin and makes use of Android's AppCompatActivity and MediaPlayer classes. The user interface for the media player is created using XML layout files.

**User Input**

The user can add a song to the media player by clicking the "Add Song" button. This will prompt the user to select a song from their device using Android's ACTION\_GET\_CONTENT intent. Once the user has selected a song, its name and duration will be displayed in the user interface.

The user can also seek through the song by sliding the seek bar left or right. The media player will pause while the user is seeking and will resume playing once the user releases the seek bar.

**MediaPlayer Functions**

The media player has play/pause functionality that is activated by clicking the "Play" button. If the media player is currently playing a song, clicking the "Play" button will pause the song and change the "Play" button's image to a play icon. If the media player is currently paused, clicking the "Play" button will resume the song and change the "Play" button's image to a pause icon.

The media player can also be stopped by calling the stop() function. This will reset the media player and the seek bar's progress will be set back to 0.

Seeking through the song is achieved by calling the seekTo() function on the media player and passing in the desired position in milliseconds. The seekStart() and seekStop() functions are used to reset the media player before and after seeking, respectively.

**Threading**

The media player's seek bar is updated using a thread. The thread runs an infinite loop that increments the seek bar's progress by 1000 milliseconds (1 second) every iteration. The loop is stopped and started using the stop() and start() functions, respectively.

**Conclusion**

In summary, this project has demonstrated the use of Android's MediaPlayer class to create a media player that allows users to select a song, play/pause the song, and seek through the song using a seek bar. The media player's seek bar is updated using a thread.

There are many potential improvements that could be made to this media player in the future. For example, the media player could be enhanced to support multiple songs, or to display album artwork for the currently playing song.