Industrial Internship Report on

”Music Player App”

Domain:Core Java

Prepared by

[K.Nivedha]

Executive Summary
This report provides details of the Industrial Internship provided by upskill Campus and The IoT Academy in collaboration with Industrial Partner UniConverge Technologies Pvt Ltd (UCT). This internship was focused on a **Music Player App** provided by UCT. We had to finish the project including the report in 6 weeks’ time. **My project title: Music Player App** **Domain:Core Java** **Problem Statement:**

Description:

Develop a music player application that allows users to play, manage, and enjoy their music collection. The music player should provide essential functionalities for organizing and playing music files in various formats.

Scope:

The scope of a music player app includes core functionalities like music playback, playlist management, and library organization, along with user-friendly UI/UX design. Advanced features may include streaming integration, lyrics display, and offline mode, all optimized for performance and cross-platform compatibility.

TABLE OF CONTENTS

1.Preface

Summary of the whole 6 weeks' work.

- The primary focus of first week is to understand the one of thecore java projects from multiple project lists.
- This week, my focus was on the project's role, its achievements, user usage, and its usefulness to the user.
- The primary focus of second week is to **1. Set Up Development Environment. 2. Create a New Android Project. 3. Design the User Interface.**
- The primary focus of Third week is to **Request Permissions and Implement the MainActivity.java,MusicAdapter.java, MusicList.java.**
- The primary focus of Fourth week is to **complete the entire code of MainActivity.java, MusicAdapter.java, MusicList.java.**

Need of relevant Internship in career development.

1.Practical Experience:

Internships provide hands-on experience in a real-world work environment. This practical experience allows interns to apply theoretical knowledge gained in academic settings to actual projects and tasks.

2.Skill Development:

Internships offer opportunities to develop and enhance valuable skills relevant to a specific industry or field. These skills may include technical abilities, communication skills, problem-solving skills, time management, and teamwork.

3.Industry Exposure:

Internships provide insights into the workings of a particular industry or profession. Interns gain exposure to industry practices, trends, challenges, and opportunities, which can help them make informed career decisions.

4.Career Exploration:

Internships provide opportunities for career exploration and self-discovery. Interns can explore different roles, industries, and work environments to determine their interests, strengths, and areas for growth.

5.Professional Development:

Internships offer opportunities for professional development and growth. Interns receive feedback, guidance, and mentorship from experienced professionals, helping them develop professionally and refine their career goals.

Brief about project/problem statement.

Project Titlle: **Music Player App**

Project Overview:

The music player app is designed to provide users with an intuitive platform for organizing and playing their music. It supports various audio formats, offers playlist creation and management, and includes features like streaming integration and offline playback, all within a sleek, user-friendly interface.

Project Objective:

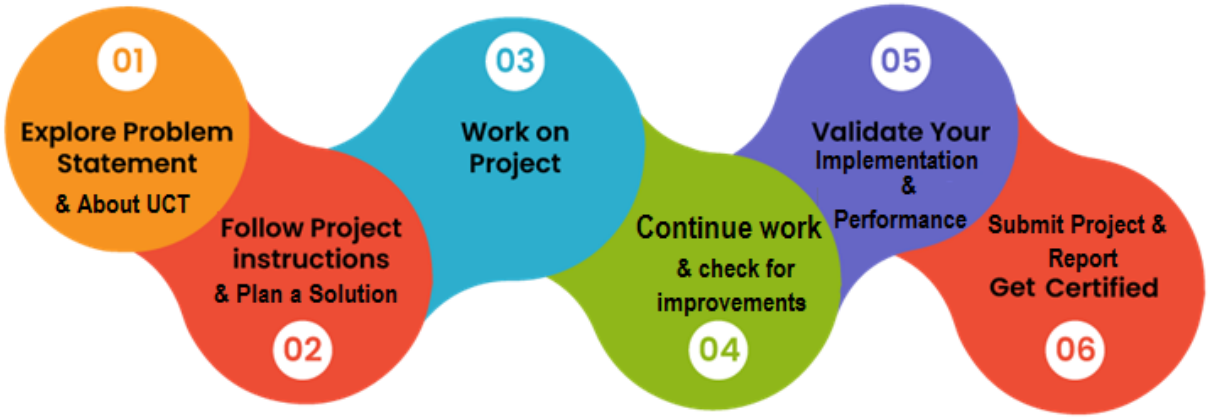
The objective of the music player app is to deliver a seamless and enjoyable music listening experience by enabling users to easily play, organize, and manage their music collections. The app aims to support multiple audio formats, provide robust playlist and library management, and integrate advanced

features like streaming services and offline playback, all while maintaining optimal performance and a user-friendly interface.

Opportunity given by USC/UCT.

I would like to extend my sincere appreciation for providing me with the opportunity to intern at [**UniConverge Technologies Pvt Ltd**]. My internship experience has been incredibly valuable, and I am grateful for the knowledge, skills, and insights I have gained during my time with your team.

How Program was planned



My Learnings and overall experience.

- Overall, the project serves as a valuable tool for users seeking to challenge their knowledge and engage in interactive learning experiences.
- Thank to all , who have helped you directly or indirectly.

My message to my juniors and peers.

UCT has offered an exceptional internship opportunity. We encourage you to enroll for the internship and take full advantage of this valuable opportunity.

2.Introduction

2.1About UniConverge Technologies Pvt Ltd

A company established in 2013 and working in Digital Transformation domain and providing Industrial solutions with prime focus on sustainability and RoI.

For developing its products and solutions it is leveraging various **Cutting Edge Technologies e.g. Internet of Things (IoT), Cyber Security, Cloud computing (AWS, Azure), Machine Learning, Communication Technologies (4G/5G/LoRaWAN), Java Full Stack, Python, Front end** etc.

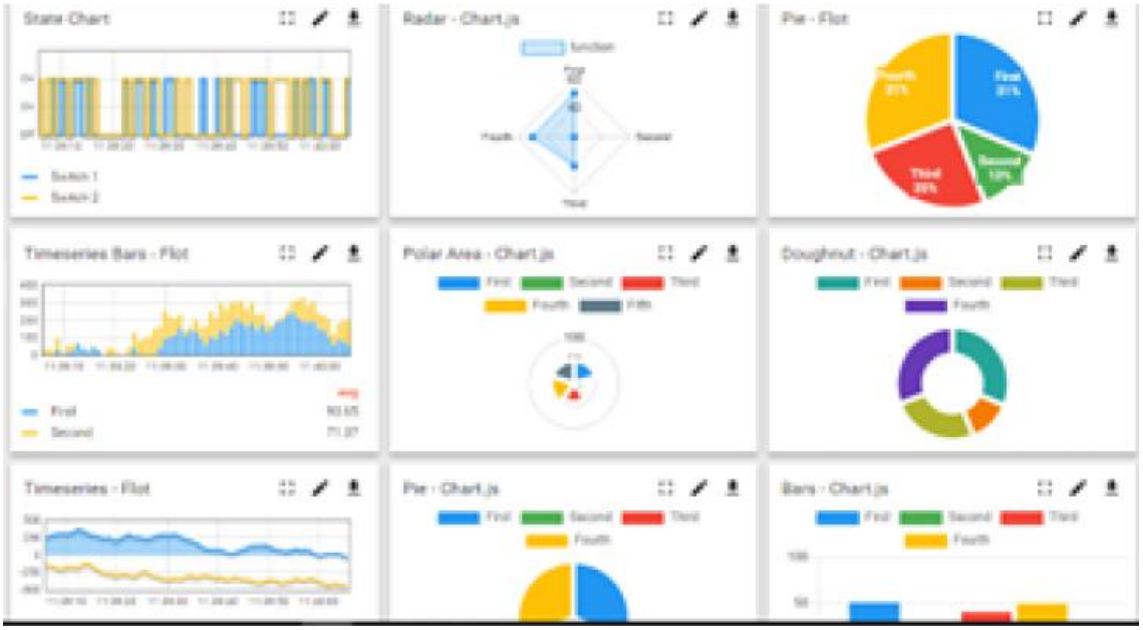
i. UCT IoT Platform (UCT Insight)

UCT Insight is an IOT platform designed for quick deployment of IOT applications on the same time providing valuable "insight" for your process/business. It has been built in Java for backend and ReactJS for Front end. It has support for MySQL and various NoSql Databases.

- It enables device connectivity via industry standard IoT protocols - MQTT, CoAP, HTTP, Modbus TCP, OPC UA
- It supports both cloud and on-premises deployments.

It has features to

- Build Your own dashboard
- Analytics and Reporting
- Alert and Notification
- Integration with third party application(Power BI, SAP, ERP)
- Rule Engine



ii. Smart Factory Platform (FACTORY WATCH)

Factory watch is a platform for smart factory needs.

It provides Users/ Factory

- with a scalable solution for their Production and asset monitoring
- OEE and predictive maintenance solution scaling up to digital twin for your assets.
- to unleased the true potential of the data that their machines are generating and helps to identify the KPIs and also improve them.
- A modular architecture that allows users to choose the service that they what to start and then can scale to more complex solutions as per their demands.

Its unique SaaS model helps users to save time, cost and money.

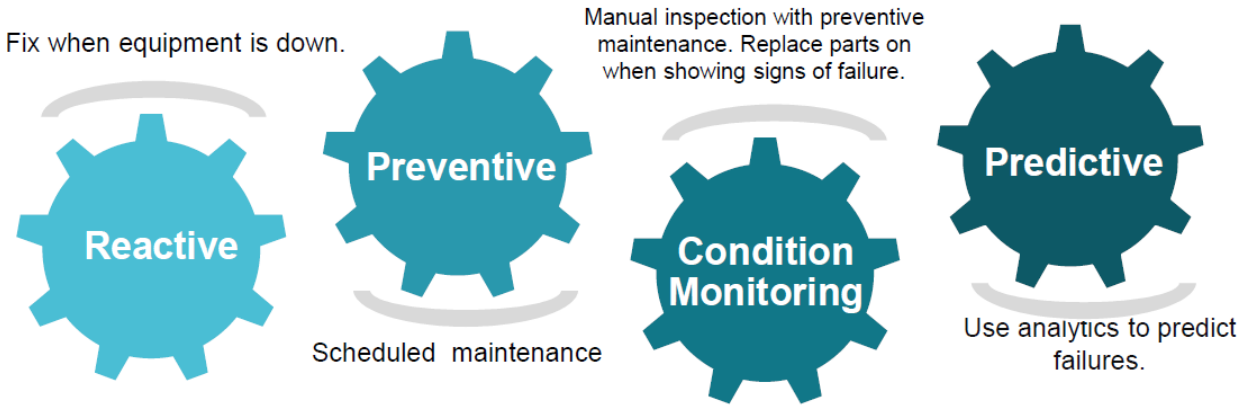
Machine	Operator	Work Order ID	Job ID	Job Performance	Job Progress		Output		Rejection	Time (mins)				Job Status	End Customer
					Start Time	End Time	Planned	Actual		Setup	Pred	Downtime	Idle		
CNC_S7_81	Operator 1	WO0405200001	4168	58%	10:30 AM		55	41	0	80	215	0	45	In Progress	i
CNC_S7_81	Operator 1	WO0405200001	4168	58%	10:30 AM		55	41	0	80	215	0	45	In Progress	i

iii. LoRaWAN based Solution

UCT is one of the early adopters of LoRAWAN teschnology and providing solution in Agritech, Smart cities, Industrial Monitoring, Smart Street Light, Smart Water/ Gas/ Electricity metering solutions etc.

iv. Predictive Maintenance

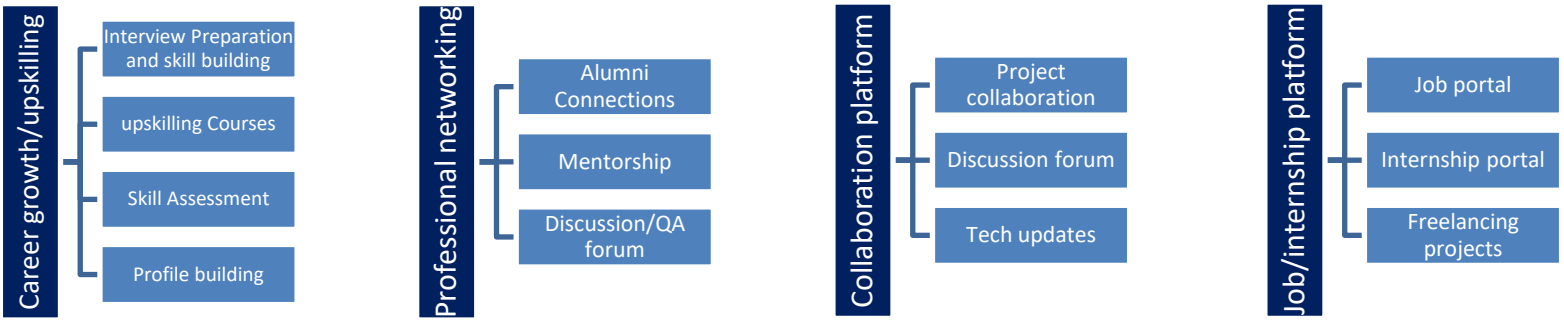
UCT is providing Industrial Machine health monitoring and Predictive maintenance solution leveraging Embedded system, Industrial IoT and Machine Learning Technologies by finding Remaining useful life time of various Machines used in production process.



2.2About upskill Campus (USC)

upskill Campus along with The IoT Academy and in association with Uniconverge technologies has facilitated the smooth execution of the complete internship process.

USC is a career development platform that delivers **personalized executive coaching** in a more affordable, scalable and measurable way.



2.3The IoT Academy

The IoT academy is EdTech Division of UCT that is running long executive certification programs in collaboration with EICT Academy, IITK, IITR and IITG in multiple domains.

2.4 Objectives of this Internship program

The objective for this internship program was to

- ☛ get practical experience of working in the industry.
- ☛ to solve real world problems.
- ☛ to have improved job prospects.
- ☛ to have Improved understanding of our field and its applications.
- ☛ to have Personal growth like better communication and problem solving.

2.5 Reference

[1] GeeksforGeeks- https://www.geeksforgeeks.org/

[2] youtube videos

2.6 Glossary

Terms	Acronym
UI	User Interface
API	Application Programming Interface
SDK	Software Development Kit
MP3	MPEG Audio Layer III
UX	User Experience

3 Problem Statement

Description:

Develop a music player application that allows users to play, manage, and enjoy their music collection. The music player should provide essential functionalities for organizing and playing music files in various formats.

Scope:

The project will focus on building a music player application with the following core functionalities:

- Playback of various audio formats.
- Playlist creation, editing, and management.
- Offline playback and music file management.
- Customizable user interface and settings.
- Metadata editing and automatic organization features.
- Integration with streaming services (optional)

4 Existing and Proposed solution:

Existing solution:

Existing music player applications often fall short in one or more of the following areas:

- Format Support: Limited support for various audio formats (e.g., FLAC, AAC, MP3, etc.).
- User Experience: Lack of intuitive design and ease of use, leading to user frustration.
- Customization: Limited options for users to personalize their listening experience, such as equalizer settings, themes, and playlist management.
- Performance: Issues with lag, buffering, or inefficient resource usage, especially on low-end devices.
- Cross-Platform Consistency: Inconsistent user experience across different operating systems (e.g., Android, iOS, Windows).

Proposed Solution:

Objective: To develop a new music player application that overcomes the limitations of existing solutions by providing a seamless, customizable, and high-performance music playback experience across multiple platforms.

Key Features of the Proposed Solution:

1. **Comprehensive Format Support:**
    - ❖ Support for both lossy (e.g., MP3, AAC) and lossless (e.g., FLAC, ALAC) audio formats.
    - ❖ Integration with popular streaming services for both online and offline playback, ensuring users have access to a broad range of music content.
2. **Enhanced User Experience:**
    - ❖ A clean, intuitive, and aesthetically pleasing user interface that is easy to navigate.
    - ❖ Quick access to essential features such as playback controls, playlists, and equalizer settings.

- Customizable themes and layouts to match user preferences, allowing for a more personalized user experience.

3. **Optimized Performance:**

- Efficient resource usage to ensure smooth performance even on low-end devices.
- Fast loading times and minimal lag, with a focus on reducing memory and CPU usage, ensuring that the app runs efficiently on a wide range of devices.

4. **Cross-Platform Consistency:**

- A unified experience across Android, iOS, Windows, and macOS, with consistent features and design.
- Cloud sync options to maintain playlists, settings, and metadata across devices, ensuring users have access to their music library no matter where they are.

4.4 Code submission (Github link)

MusicPlayer/MainActivity.java:

https://github.com/nivedhak2003/upskillcampus/blob/master/app/src/main/java/com/example/musicplayer/MainActivity.java

MusicPlayer/activity_main.xml:

https://github.com/nivedhak2003/upskillcampus/blob/master/app/src/main/res/layout/activity_main.xml

4.5 Report submission (Github link) :

https://github.com/nivedhak2003/upskillcampus/blob/master/MusicPlayerApplication_K.Nivedha_USC_UCT.pdf

5 Proposed Design/ Model

1.Requirements Gathering

2.Conceptualization and Ideation

3. Architecture Design

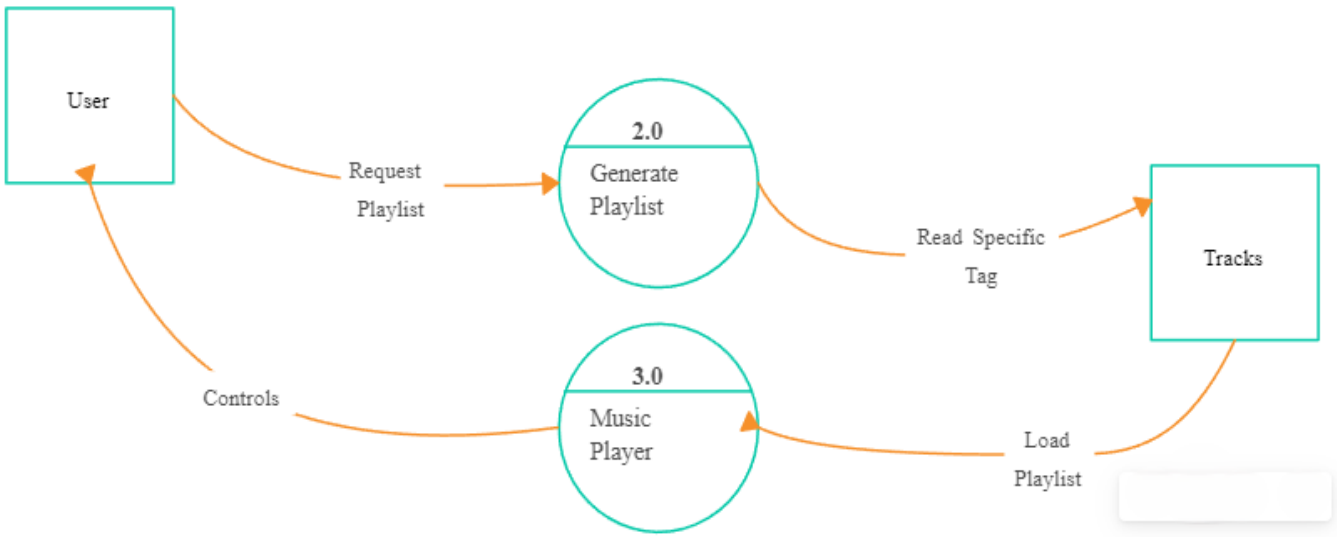
4. User Interface (UI) Design

5. Development

6. Feature Implementation

7. Integration and Testing

5.1 Interfaces (if applicable)



6 Performance Test.

6.1Test Plan/ Test Cases

Test Plan:

1 Introduction

- Objective: Ensure that the Music Player app functions correctly according to the requirements.
- Scope: Testing will cover functionality, performance, user interface, security, and compatibility.
- Tools: Android Studio, JUnit for unit testing.

2 Environment:

- Operating System: Windows, macOS, Linux
- Java Version: Java 8 or higher
- Dependencies: JDK, Maven/Gradle, any required libraries
- Devices: Desktop, Mobile (optional based on application)

Test Scenarios:

1 Installation and Launching

- Verify that the app installs correctly on all supported platforms.
- Verify that the app launches without errors.

2 User Interface

- Test all UI elements (buttons, sliders, lists) for correct visibility, functionality, and responsiveness.
- Verify that the UI adapts correctly to different screen sizes.

3 Audio Playback

- Test playback of various audio formats (MP3, WAV, AAC, etc.).
- Verify that play, pause, stop, next, and previous controls work correctly.
- Check for smooth transition between tracks.
- Verify volume controls and mute functionality.

4 Playlist Management

- Create, edit, and delete playlists.
- Add and remove songs from playlists.
- Verify the order of songs in playlists.

5 Search Functionality

- Search for songs by title, artist, and album.

- Test different search criteria (partial matches, case sensitivity).

6 Library Management

- Verify that the app correctly imports and displays songs from the device library.
- Check that sorting and filtering options work as expected.

7 Audio Effects

- Test the functionality of equalizers and other audio effects.
- Verify that changes in settings are applied in real-time.

8 Performance

- Measure app startup time.
- Test for smooth operation under heavy load (e.g., large playlists).
- Verify that there are no memory leaks or crashes.

6.2 Test Procedure

Test Cases:

Test Case 1: App Launch

- **Objective**: Verify that the application launches correctly.
- **Preconditions**: The app is installed.

- **Steps**:

    1. Launch the application.

- **Expected Result**: The application should launch without errors and display the main screen.

Test Case 2: Play Audio File

- **Objective**: Verify that the app can play an audio file.
- **Preconditions**: The app is installed, and a valid audio file is present.
- **Steps**:

    1. Open the app.
    2. Select an audio file.
    3. Click on the play button.

- **Expected Result**: The selected audio file should play without any issues.

Test Case 3: Create Playlist

- **Objective**: Verify that a user can create a new playlist.
- **Preconditions**: The app is installed.

- **Steps**:

    1. Open the app.
    2. Navigate to the playlist section.
    3. Click on the "Create New Playlist" button.
    4. Enter a name for the playlist and save.

- **Expected Result**: A new playlist with the given name should be created.

Test Case 4: Search Song

- **Objective**: Verify that the search functionality works correctly.
- **Preconditions**: The app is installed, and the library is populated with songs.
- **Steps**:

1. Open the app.
2. Enter a song title in the search bar.
3. Press enter or click the search button.

- **Expected Result**: The search results should display all matching songs.

Test Case 5: Adjust Volume

- **Objective**: Verify that the volume can be adjusted.
- **Preconditions**: The app is installed, and a song is playing.
- **Steps**:
    1. While the song is playing, adjust the volume slider.
- **Expected Result**: The volume should increase or decrease according to the slider movement

6.3 Performance Outcome

1. Application Startup Time

- **Metric**: Time taken from launching the app to the point where the user interface is fully loaded and responsive.
- **Expected Outcome**: The application should start up within a few seconds (typically under 5 seconds) on most devices.
- **Assessment**: Measure using time logging or profiling tools within the code, or by using external tools like JProfiler or VisualVM.

2. Audio Playback Latency

- **Metric**: The delay between the user pressing the play button and the audio starting to play.
- **Expected Outcome**: Latency should be minimal, ideally under 100ms, to provide a seamless experience.
- **Assessment**: Use a high-resolution timer to measure the time from user input to the start of playback.

3. Memory Usage

- **Metric**: The amount of RAM consumed by the application during idle, active playback, and when managing large playlists.
- **Expected Outcome**: The application should efficiently manage memory, avoiding leaks and keeping usage within reasonable bounds (e.g., under 200MB during normal operation).
- **Assessment**: Monitor memory usage using tools like VisualVM, JConsole, or Java's Runtime.getRuntime().totalMemory() method.

4. CPU Utilization

- **Metric**: The percentage of CPU resources consumed during audio playback, playlist management, and UI interactions.

- **Expected Outcome**: CPU usage should be low during audio playback (ideally under 10% on a modern processor), even with multiple tracks or effects applied.

- **Assessment**: Use system monitoring tools or Java profiling tools to measure CPU usage during different app states.

5. Responsiveness

- **Metric**: The responsiveness of the user interface, especially under load (e.g., when a large playlist is being processed or multiple songs are being played).

- **Expected Outcome**: UI interactions should remain smooth, with no noticeable lag, even when the app is under heavy load.

- **Assessment**: Subjectively test UI responsiveness and back this up with profiling data showing frame rates or event handling delays.

6. Battery Consumption (for mobile devices)

- **Metric**: The rate at which the application drains the battery during continuous usage.

- **Expected Outcome**: The app should be optimized to minimize battery consumption, especially during playback and idle states.
- **Assessment**: Measure battery usage on mobile devices using tools like Android Studio Profiler or Xcode Instruments.

7. Real-time Audio Effects Processing

- **Metric**: The time required to apply audio effects (like equalizers, reverb, etc.) in real-time without causing playback interruptions.
- **Expected Outcome**: Effects should be applied seamlessly, with no noticeable delays or stuttering in audio playback.
- **Assessment**: Measure the processing time for effect application using performance timers and listen for any audio anomalies during testing.

7 My learnings

1.Technical Skills Acquired

2. Software Development Practices

3. Problem-Solving and Debugging

4. Project Management

5. Understanding of Music Technology

6. Reflection on Challenges

8 Future work scope

1.Advanced Features

- ❖ Music Streaming Integration
- ❖ Cloud Syncing and Backup
- ❖ Lyrics and Metadata Display
- ❖ Smart Playlists

2.Enhanced User Experience

- ❖ Customizable UI Themes
- ❖ Gestures and Shortcuts
- ❖ Advanced Equalizer and Audio Effects

3. Performance Optimization

- Low Latency Audio Processing
- Resource Management
- Scalability

4. Cross-Platform Support

- Mobile and Web Versions
- Cross-Platform Synchronization

The future scope of your music player app project is vast, with opportunities for innovation in user experience, performance, AI integration, and more. By strategically prioritizing these enhancements based on user needs and technological trends, the app can evolve into a feature-rich platform that meets the demands of a diverse and growing user base.