REPORT

LAB 5 : Xylophone

Student name: Võ Khắc Đoài

Student ID: 21IT335

Student email: doaivk.21it@vku.udn.vn

1. Introduction

The purpose of this report is to provide an overview of a simple Flutter application that implements a Xylophone. The app allows users to play different sounds by pressing buttons that correspond to various musical notes. This project is built using Flutter's Material package for the UI and the audioplayers package for sound playback.

2. Objectives

The primary objectives of this project are:

- To create a visually appealing, simple musical instrument using Flutter.
- To understand and implement audio playback using the audioplayers package.
- To develop an interactive UI that reacts to user inputs and triggers audio.

3. Methodology

- **Flutter Framework**: The app is built using Flutter, a UI toolkit for building natively compiled applications for mobile, web, and desktop from a single codebase. The MaterialApp and Scaffold widgets are used to define the app's layout.

- **audioplayers Package**: The audioplayers package is integrated into the app to handle the playback of sound files. This package allows us to load and play local sound files when a button is pressed.

- **UI Design**: The user interface consists of seven colored buttons (keys), each of which plays a different sound when pressed. The Expanded widget is used to ensure that each button occupies an equal portion of the screen. The TextButton widget captures user input, and the background color of each button corresponds to a unique sound.

- **Sound Playback**: The sound is triggered via the playSound(int soundNumber) function, which plays a specific audio file based on the number passed as an argument. Audio files are named sequentially (note1.wav, note2.wav, etc.), and are stored as assets in the app.

4. **Results**

5. Discussion

This project demonstrates the ease with which Flutter and audioplayers can be used to create interactive multimedia applications. By using the Expanded widget, the app ensures that the layout is flexible and responsive, adjusting itself to different screen sizes. The SafeArea widget prevents the app's UI from being obscured by system interfaces like the status bar or notches on mobile devices.

The integration of sound playback through the audioplayers package is straightforward. However, it's important to ensure that the audio files are properly included in the app's assets and referenced correctly in the code. Additionally, there is a slight delay when playing the sound for the first time, which could be optimized by preloading the audio files in a more advanced version of the app.

6. Conclusion

This simple xylophone app demonstrates how to build interactive apps using Flutter. By combining UI elements with audio functionality, users can experience instant feedback by playing sounds with a single tap. The project fulfills its objectives of teaching basic Flutter layout principles, audio integration, and user interaction. It also provides a foundation for further exploration into more complex apps that could incorporate advanced audio features such as looping, recording, or sound effects.