VISVESVARAYA TECHNOLOGICAL UNIVERSITY
BELGAUM-590014

An Internship Report on

"MP3 PLAYER IN JAVA"

Submitted in partial fulfillment of the requirements for the award of the degree of
Bachelor of Engineering in Artificial Intelligence and Machine Learning

Submitted by:
OORJA SAXENA
(1DT21AI043)
ANUSHRI J
(1DT21AI005

Guide & Coordinator

Mrs. Sowbhagya M P
Asst. Professor
DSATM

HOD

Dr. Sandhya
Professor & Head
DSATM

DAYANANDA SAGAR ACADEMY OF TECHNOLOGY AND MANAGEMENT

Department of Artificial Intelligence and Machine Learning

(Accredited by NACC with A+ Grade)

Udayapura, Kanakapura Road, Bangalore-560082

2022-2023

DAYANANDA SAGAR ACADEMY OF TECHNOLOGY AND MANAGEMENT

Department of Artificial Intelligence and Machine Learning

(Accredited by NACC with A+ Grade)

Udayapura, Kanakapura Road, Bangalore-560082

2022-2023

CERTIFICATE

Certified that the Internship work entitled "**Projects in Java"** carried out by **OORJA SAXENA, (1DT21AI043) & ANUSHRI J (1DT21AI005)** are bonafide students of **Dayananda Sagar Academy of Technology and Management** in partial fulfilment for the award of Bachelor of Engineering in **Artificial Intelligence and Machine Learning** of the Visvesvaraya Technological University, Belgaum during the year 2022-23. It is certified that all corrections/suggestions indicated for InternalAssessment have been incorporated in the Report deposited in the departmental library. The Internship report has been approved as it satisfies the academic requirements in respect of Internship work prescribed for the said Degree.

.

Signature of Guide & Coordinator
Mrs. Sowbhagya M P
Asst. Prof.
Dept. of AIML

Signature of HOD
Dr.Sandhya N
Professor & Head
Dept. of AIML

STUDENT DECLARATION

We, Student **ANUSHRI J, 1DT21AI005 & OORJA SAXENA, 1DT21AI043** hereby declare that the presented report of internship titled "**MP3 PLAYER IN JAVA"** isuniquely prepared by me after the completion of "one month" Internship work carried out at "**Infosys Spring board**".

We also confirm that the report is prepared for my academic requirement, not forany other Purpose.

…….

OORJA SAXENA

1DT21AI043

ANUSHRI J

1DT21AI005

Department of Artificial Intelligence and

Machine Learning

DSATM, Bangalore

Table of Contents

CHAPTER 1

INTRODUCTION

In this project, we are building a MP3 Player in Java. Java music player is a simple classic mp3 player which has features like playing selected mp3 music files, pausing the music, resuming the music, and stopping the music. The music player is used daily by all types of users. Music helps users to create a fresh mind, inspire life, and also boost the mind of the user. We have used java.swing and java.awt packages to create the GUI of the application and jaco-mp3-player-0.9.3 jar file to play, pause, resume, stop and adjust volume of mp3 files and in design of GUI. MP3 player can be combined and used to make media players.

Objectives of Summer Internship:

- Understanding about object-oriented programming and Gain knowledge about the capability to store information together in an object.
- Understand the capability of a class to rely upon another class and functions.
- Understand about constructors which are special type of functions.
- Create and process data in files using file l/O functions
- Use the generic programming features of Java including Exception handling.
- Understand the use of Eclipse/NetBeans IDE to create Java Applications.

Using java programming to develop programs for solving real-world problems.

CHAPTER 2

ROLE OF THE STUDENT

In this project, Java is used for all the features of the MP3 player. Each individual student gained the knowledge of Java. Mainly, two Java packages are used to build the MP3 player namely java.awt and java.swing and thee jloc java jar

- The basic introduction to Java.
- Basic introduction to creating GUIs.
- Knowledge about classes, nesting and inheritance
- Basic knowledge of constructors and methods
- Basic Knowledge about jawa.awt and its classes and methods
- Basic knowledge about java.swing and its classes and methods
- Using instance methods for volumeup and volumedown in MP3 player.

CHAPTER 3

SCHEDULE OF INTERNSHIP

Projects in Java (8hr 40 mins Infosys spring board certification course). Develop a project using the knowledge gained.

Date: 25/10/2022 to 29/10/2022

Time: 9.30 am to 4.00pm (online)

- Identify a problem statement
- Solve using Java
- Design and Implementations
- Prove Accuracy towards problem statement
- Project Review
- Project Report

After developing the mentor need to evaluate the project and also collect certificate

CHAPTER 4

TOOLS AND TECHNOLOGY USED

4.1 JAVA

Java is a high-level, class-based, object-oriented programming language that is designed to have as few implementation dependencies as possible. It is a general-purpose programming language intended to let programmers *write once, run anywhere* (WORA), meaning that compiled Java code can run on all platforms that support Java without the need to recompile. Java applications are typically compiled to bytecode that can run on any Java virtual machine (JVM) regardless of the underlying computer architecture. The Java runtime provides dynamic capabilities (such as reflection and runtime code modification) that are typically not available in traditional compiled languages.

4.2 JAVA.AWT PACKAGE

Java AWT (Abstract Window Toolkit) is *an API to develop Graphical User Interface (GUI) or windows-based applications* in Java. Java AWT components are platform-dependent i.e., components are displayed according to the view of operating system. AWT is heavy weight i.e., its components are using the resources of underlying operating system (OS). The java.awt package provides classes for AWT API such as TextField, Label, TextArea, RadioButton, Checkbox, Choice, List, etc.

4.3 JAVA.SWING PACKAGE

Java Swing is a part of Java Foundation Classes (JFC) that is used to create window-based applications. It is built on the top of AWT (Abstract Windowing Toolkit) API and entirely written in java. Unlike AWT, Java Swing provides platform-independent and lightweight components. The javax.swing package provides classes for java swing API such as JButton, JTextField, JTextArea, JRadioButton, JCheckbox, JMenu, JColorChooser etc.

4.4 NETBEANS IDE

The NetBeans Platform is a generic framework for simplifying the development of Java Swing desktop applications. It provides reusable services and components that are common to desktop applications, such as user interface management (windows, menus, toolbars…), settings management, storage management, etc - allowing developers to focus on coding business logic specific to their applications, without having to code the “plumbing”. Therefore, it reduces a lot of development time and increase productivity. And NetBeans allows you to develop applications based on NetBeans Platform right inside the IDE. For Java development, you can use NetBeans to develop almost any kind of Java applications, from Java desktop apps (Swing & JavaFX) to Java web apps (Servlet, JSP, and EJB). For Swing development, the GUI Builder helps you visually design layout and drag and drop user interface components. Likewise, the SceneBuilder helps you design JavaFX applications quickly and easily. For developing Java web applications, NetBeans supports two types of servers: GlassFish and Apache Tomcat.

Jaco MP3 Player (Jloc Java Jar File)

JACo MP3 Player is a cross platform java mp3 player.
- very low CPU usage (~2%)
- incredible small library (~90KB)
- doesn't need JMF (and his system dependent plug-ins)
- easy to integrate in any application (for the simplest case you need to write only one line of code)
- easy to integrate in any web page (as a java applet)

CHAPTER 5

REFLECTIONS

This was the project which was completed under the domain of Java. The objective of this project was to build a text MP3. The project features are as follows :

- User may open Folder to add and play various mp3 files within it.
- User may see track lists and play desired ones accordingly.
- Supports the music formats .mp3
- Interactive GUI
- Consists of Pause/Play/Stop/Repeat Features
- Consists of a Volume controller

OUTCOMES:

fig 5.1 Interface

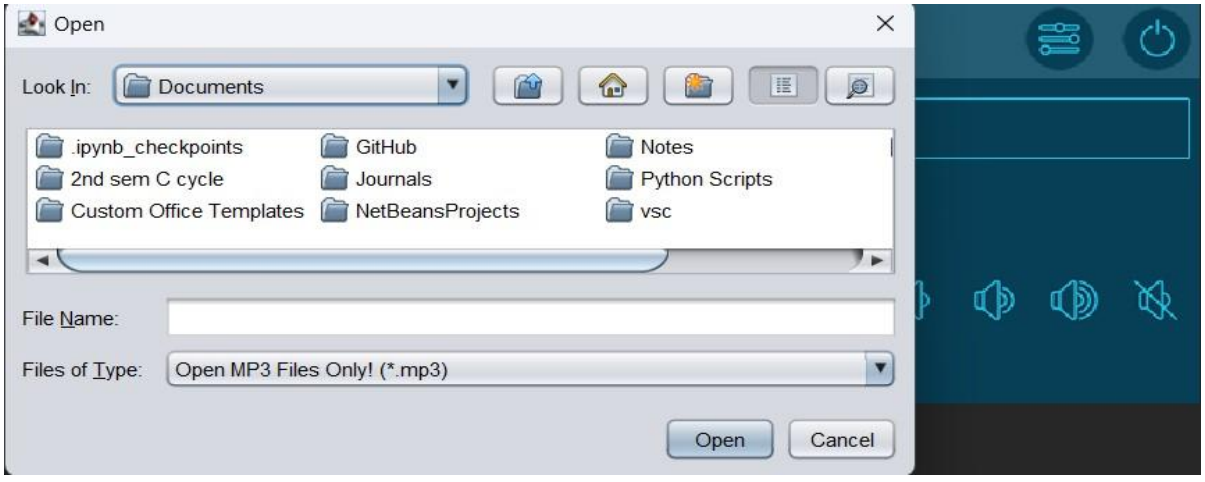

fig 5.2 File chooser

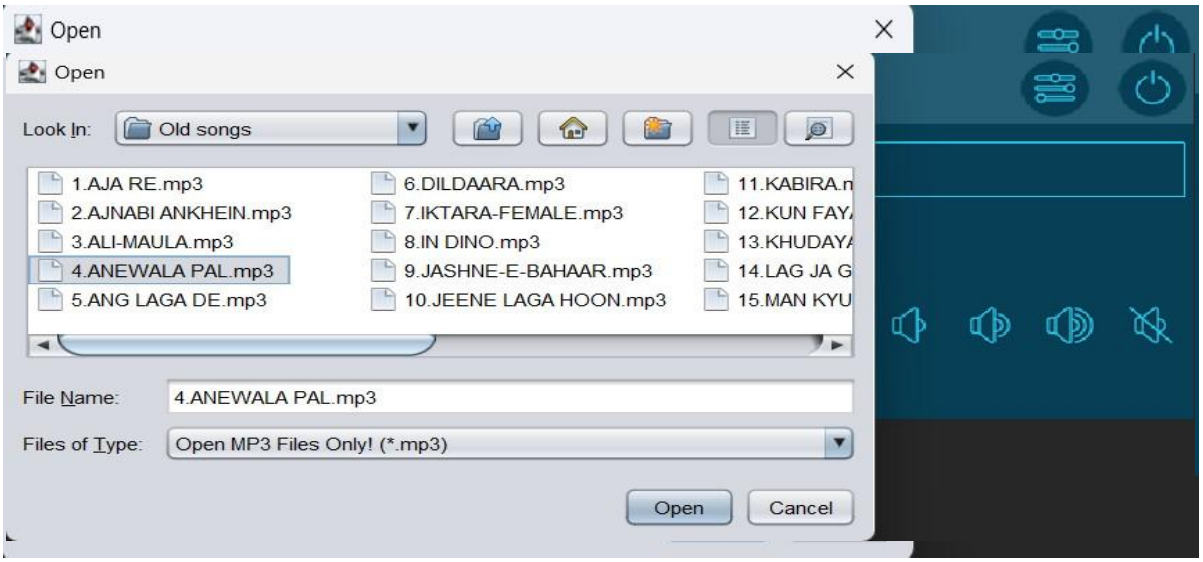

fig 5.3 File chooser

fig 5.4 Output

WORKING CODE

```
package music.pk;

import jaco.mp3.player.MP3Player;
import java.awt.Dimension;
import java.awt.Font;
import java.io.File;
import java.nio.file.Paths;
import javax.sound.sampled.AudioSystem;
import javax.sound.sampled.Clip;
import javax.sound.sampled.FloatControl;
import javax.sound.sampled.Line;
import javax.sound.sampled.LineUnavailableException;
import javax.sound.sampled.Mixer;
import javax.swing.ImageIcon;
import javax.swing.JFileChooser;
import javax.swing.JOptionPane;


public class PlayerFrame extends javax.swing.JFrame {

    MP3Player player;
    File songFile;
    String currentDirectory = "home.user";
    String currentPath;
    String imagePath;

    String appName = "MP3 Player";
    boolean repeat = false;
    boolean windowCollapsed = false;
    int xMouse, yMouse;
```

```
    public PlayerFrame() {
        initComponents();
        appTitle.setText(appName);

        songFile = new File("\"C:\\Users\\Oorja Saxena\\Music\\Pinoy Pop\\Wag Kang Umalis-
Nobrvnd.mp3.mp3\"");


        String fileName = songFile.getName();
        songNameDisplay.setText(fileName);
        player = mp3Player();
        player.addToPlayList(songFile);
        currentPath = Paths.get(".").toAbsolutePath().normalize().toString();
        imagePath = "\\images";
    }

@SuppressWarnings("unchecked")

private void playBtnMouseClicked(java.awt.event.MouseEvent evt) {
        player.play();
    }

    private void stopBtnMouseClicked(java.awt.event.MouseEvent evt) {
        player.stop();
    }

    private void pauseBtnMouseClicked(java.awt.event.MouseEvent evt) {
        player.pause();
    }

    private void repeatBtnMouseClicked(java.awt.event.MouseEvent evt) {
        if(repeat == false){
            repeat = true;
            player.setRepeat(repeat);
```

```
            String image = currentPath+imagePath+"\\repeat_enabled.png";
            repeatBtn.setIcon(new ImageIcon(image));
        }else if(repeat == true){
            repeat = false;
            player.setRepeat(repeat);

            String image = currentPath+imagePath+"\\repeat.png";
            repeatBtn.setIcon(new ImageIcon(image));
        }
    }

    private void appTitleMousePressed(java.awt.event.MouseEvent evt) {
        xMouse = evt.getX();
        yMouse = evt.getY();
    }

    private void appTitleMouseDragged(java.awt.event.MouseEvent evt) {
        int x = evt.getXOnScreen();
        int y = evt.getYOnScreen();

        this.setLocation(x - xMouse, y - yMouse);
    }

    private void quitBtnMouseClicked(java.awt.event.MouseEvent evt) {
        this.dispose();
    }

    private void settingsBtnMouseClicked(java.awt.event.MouseEvent evt) {
        JOptionPane.showMessageDialog(this, "Your settings dilog will be popup here!");
    }

    private void openBtnMouseClicked(java.awt.event.MouseEvent evt) {
        JFileChooser openFileChooser = new JFileChooser(currentDirectory);
```

```
    openFileChooser.setFileFilter(new FileTypeFilter(".mp3", "Open MP3 Files Only!"));
    int result = openFileChooser.showOpenDialog(null);
    if(result == JFileChooser.APPROVE_OPTION){

        songFile = openFileChooser.getSelectedFile();
        player.addToPlayList(songFile);
        player.skipForward();
        currentDirectory = songFile.getAbsolutePath();
        songNameDisplay.setText("Playing Now... | " + songFile.getName());
    }
}

private void appTitleMouseClicked(java.awt.event.MouseEvent evt) {
    if(evt.getClickCount() == 2){
        if(windowCollapsed == false){
            windowCollapsed = true;
            this.setSize(new Dimension(this.getSize().width, 50));

            appTitle.setFont(new Font("Nirmala UI", 0, 12));
            appTitle.setText("Playing Now... | " + songFile.getName());
        }else if(windowCollapsed == true){
            windowCollapsed = false;
            this.setSize(new Dimension(this.getSize().width, 250));

            appTitle.setFont(new Font("Nirmala UI", 0, 18));
            appTitle.setText(appName);
        }
    }
}

private void volumeDownBtnMouseClicked(java.awt.event.MouseEvent evt) {
    volumeDownControl(0.1);
}
```

```
    private void volumeUpBtnMouseClicked(java.awt.event.MouseEvent evt) {
        volumeUpControl(0.1);
    }
    private void volumeFullBtnMouseClicked(java.awt.event.MouseEvent evt) {
        volumeControl(1.0);
    }
    private void muteBtnMouseClicked(java.awt.event.MouseEvent evt) {
        volumeControl(0.0);
    }

public static void main(String args[]) {

java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new PlayerFrame().setVisible(true);
            }
        });
    }

    private javax.swing.JLabel appTitle;
        private javax.swing.JPanel controlPanel;
        private javax.swing.JPanel headerPanel;
        private javax.swing.JLabel jLabel7;
        private javax.swing.JPanel mainPanel;
        private javax.swing.JLabel muteBtn;
        private javax.swing.JLabel openBtn;
        private javax.swing.JLabel pauseBtn;
        private javax.swing.JLabel playBtn;
        private javax.swing.JLabel quitBtn;
        private javax.swing.JLabel repeatBtn;
        private javax.swing.JLabel settingsBtn;
        private javax.swing.JLabel songNameDisplay;
        private javax.swing.JPanel songNameMainPanel;
        private javax.swing.JPanel songNameSubPanel;
```

```
    private javax.swing.JLabel stopBtn;
    private javax.swing.JLabel volumeDownBtn;
    private javax.swing.JLabel volumeFullBtn;
    private javax.swing.JLabel volumeUpBtn;

  private MP3Player mp3Player(){
      MP3Player mp3Player = new MP3Player();
      return mp3Player;
    }

  private void volumeDownControl(Double valueToPlusMinus){
      Mixer.Info[] mixers = AudioSystem.getMixerInfo();
      for(Mixer.Info mixerInfo : mixers){
        Mixer mixer = AudioSystem.getMixer(mixerInfo);
        Line.Info[] lineInfos = mixer.getTargetLineInfo();
          Line line = null;
          boolean opened = true;
          try{
            line = mixer.getLine(lineInfo);
            opened = line.isOpen() || line instanceof Clip;
            if(!opened){
              line.open();
            }
            FloatControlvolControl=(FloatControl)
line.getControl(FloatControl.Type.VOLUME);
            float currentVolume = volControl.getValue();
            Double volumeToCut = valueToPlusMinus;
            float changedCalc = (float) ((float)currentVolume-(double)volumeToCut);
            volControl.setValue(changedCalc);

          }catch (LineUnavailableException lineException){
          }catch (IllegalArgumentException illException){
          }finally{
            if(line != null && !opened){
```

```
                line.close();
            }
        }
    }
}
private void volumeUpControl(Double valueToPlusMinus){
    Mixer.Info[] mixers = AudioSystem.getMixerInfo();
    for(Mixer.Info mixerInfo : mixers){
        Mixer mixer = AudioSystem.getMixer(mixerInfo);
        Line.Info[] lineInfos = mixer.getTargetLineInfo();
        for(Line.Info lineInfo : lineInfos){
            Line line = null;
            boolean opened = true;
            try{
                line = mixer.getLine(lineInfo);
                opened = line.isOpen() || line instanceof Clip;
                if(!opened){
                    line.open();
                }
                FloatControlvolControl=(FloatControl)
line.getControl(FloatControl.Type.VOLUME);
                float currentVolume = volControl.getValue();
                Double volumeToCut = valueToPlusMinus;
                float changedCalc = (float) ((float)currentVolume+(double)volumeToCut);
                volControl.setValue(changedCalc);

            }catch (LineUnavailableException lineException){
            }catch (IllegalArgumentException illException){
            }finally{
                if(line != null && !opened){
                    line.close();
                }
            }
```

```
    Internship Report on" Projects in Java"
        }
      }
    }

  private void volumeControl(Double valueToPlusMinus){
      Mixer.Info[] mixers = AudioSystem.getMixerInfo();
      for(Mixer.Info mixerInfo : mixers){
        Mixer mixer = AudioSystem.getMixer(mixerInfo);
        Line.Info[] lineInfos = mixer.getTargetLineInfo();
        for(Line.Info lineInfo : lineInfos){
          Line line = null;
          boolean opened = true;
          try{
            line = mixer.getLine(lineInfo);
            opened = line.isOpen() || line instanceof Clip;
            if(!opened){
              line.open();
            }
            FloatControl volControl = (FloatControl)
            float currentVolume = volControl.getValue();
            Double volumeToCut = valueToPlusMinus;
            float changedCalc = (float) ((double)volumeToCut);

            volControl.setValue(changedCalc);
          }catch (LineUnavailableException lineException){
          }catch (IllegalArgumentException illException){
            if(line != null && !opened){
              line.close();
            }
          }
        }
      }
    }
  }
```

CONCLUSION

With this internship we got to learn about Java and various important concepts such as classes, nesting, inheritance, constructors and methods.

We also learnt about the applications of JAVA in making functional projects with the help of external libraries, packages and jar files such as java.awt, java.swing and jloc java jar. With the help of all this information and resources we successfully made a MP3 player.

REFERENCES

1. https://www.geeksforgeeks.org/ml-principal-component-analysispca
2. http://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html
3. https://ieeexplore.ieee.org/document/6005082
4. https://machinelearningmastery.com/convolutional-layers-for-deep-learning-neural-networks/
5. GitHub resources and tutorials
6. Some downloaded tutorials