

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря
Сікорського»

Факультет інформатики та обчислювальної техніки
Кафедра інформаційних систем та технологій

Лабораторна робота №4
з дисципліни «Розробка мобільних застосувань під Android»
Тема: «Дослідження способів роботи з медіаданими»

Виконала:
студентка групи ІМ-23
Косенко Кароліна

Перевірів:
Орленко С. П.

Київ 2025

Мета роботи: дослідити яким чином платформа Андроїд надає можливість обробляти аудіо-файли та відео-файли та отримати практичні навички щодо використання інструментів відтворення медіа-даних.

Лістинг:

MainActivity.kt

```
package com.example.android_4

import android.app.Activity
import android.app.DownloadManager
import android.content.Context
import android.content.Intent
import android.media.MediaPlayer
import android.net.Uri
import android.os.Bundle
import android.os.Environment
import android.widget.Button
import android.widget.EditText
import android.widget.Toast
import android.widget.VideoView
import androidx.activity.result.contract.ActivityResultContracts
import androidx.appcompat.app.AppCompatActivity

class MainActivity : AppCompatActivity() {

    private lateinit var mediaPlayer: MediaPlayer
    private lateinit var videoView: VideoView
    private var selectedMediaUri: Uri? = null
```

```
private var isAudioSelected = false

override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_main)

    videoView = findViewById(R.id.videoView)

    val videoUri =
Uri.parse("android.resource://${packageName}/${R.raw
.sample_video}")
    videoView.setVideoURI(videoUri)

    mediaPlayer = MediaPlayer.create(this,
R.raw.sample_audio)

    val playAudio: Button =
findViewById(R.id.playAudio)
    val pauseAudio: Button =
findViewById(R.id.pauseAudio)
    val stopAudio: Button =
findViewById(R.id.stopAudio)
    val selectMedia: Button =
findViewById(R.id.selectMedia)

    val playVideo: Button =
findViewById(R.id.playVideo)
    val pauseVideo: Button =
```

```
findViewById(R.id.pauseVideo)

    val stopVideo: Button =
findViewById(R.id.stopVideo)

    val downloadMedia: Button =
findViewById(R.id.downloadMedia)

    val urlInput: EditText =
findViewById(R.id.urlInput)


    playAudio.setOnClickListener {
        if (isAudioSelected && selectedMediaUri
!= null) {
            mediaPlayer =
MediaPlayer.create(this, selectedMediaUri)
        }
        if (!mediaPlayer.isPlaying) {
            mediaPlayer.start()
        }
    }

    pauseAudio.setOnClickListener {
        mediaPlayer.pause()
    }

    stopAudio.setOnClickListener {
        mediaPlayer.stop()
        mediaPlayer.prepareAsync()
    }
```

```
        playVideo.setOnClickListener {
            if (!videoView.isPlaying) {
                videoView.start()
            }
        }

        pauseVideo.setOnClickListener {
            videoView.pause()
        }

        stopVideo.setOnClickListener {
            videoView.stopPlayback()
        }

        videoView.setVideoURI(Uri.parse("android.resource://"
            + "${packageName}/${R.raw.sample_video}"))
        videoView.seekTo(0)
    }

    selectMedia.setOnClickListener {
        val intent =
Intent(Intent.ACTION_OPEN_DOCUMENT).apply {
            addCategory(Intent.CATEGORY_OPENABLE)
            type = "*/*"
            putExtra(Intent.EXTRA_MIME_TYPES,
arrayOf("audio/*", "video/*"))
        }
        selectMediaLauncher.launch(intent)
    }
}
```

```
downloadMedia.setOnClickListener {  
    val url = urlInput.text.toString()  
  
    if (url.isNotEmpty()) {  
        val uri = Uri.parse(url)  
        val request =  
DownloadManager.Request(uri)  
            .setTitle("Downloading Media")  
            .setDescription("Downloading  
media file...")  
  
            .setDestinationInExternalPublicDir(Environment.DIRECTORY_DOWNLOADS, uri.lastPathSegment)  
  
            .setNotificationVisibility(DownloadManager.Request.VISIBILITY_VISIBLE_NOTIFY_COMPLETED)  
  
            .setAllowedNetworkTypes(DownloadManager.Request.NETWORK_WIFI or DownloadManager.Request.NETWORK_MOBILE)  
  
        val downloadManager =  
getSystemService(Context.DOWNLOAD_SERVICE) as  
DownloadManager  
        downloadManager.enqueue(request)  
  
        Toast.makeText(this, "Download  
started...", Toast.LENGTH_SHORT).show()  
    } else {  
        Toast.makeText(this, "Please enter a
```

```

valid URL", Toast.LENGTH_SHORT).show()
    }
}

private val selectMediaLauncher =
registerForActivityResult(ActivityResultContracts.StartActivityForResult()) { result ->
    if (result.resultCode == Activity.RESULT_OK) {
        result.data?.data?.let { uri ->
            selectedMediaUri = uri

            val contentResolver = contentResolver
            val type =
contentResolver.getType(uri)

            if (type?.startsWith("audio") ==
true) {
                isAudioSelected = true
                mediaPlayer.reset()
                mediaPlayer.setDataSource(this,
uri)

                mediaPlayer.prepare()
            } else if (type?.startsWith("video")
== true) {
                isAudioSelected = false
                videoView.setVideoURI(uri)
            }
        }
    }
}

```

```

    }

}

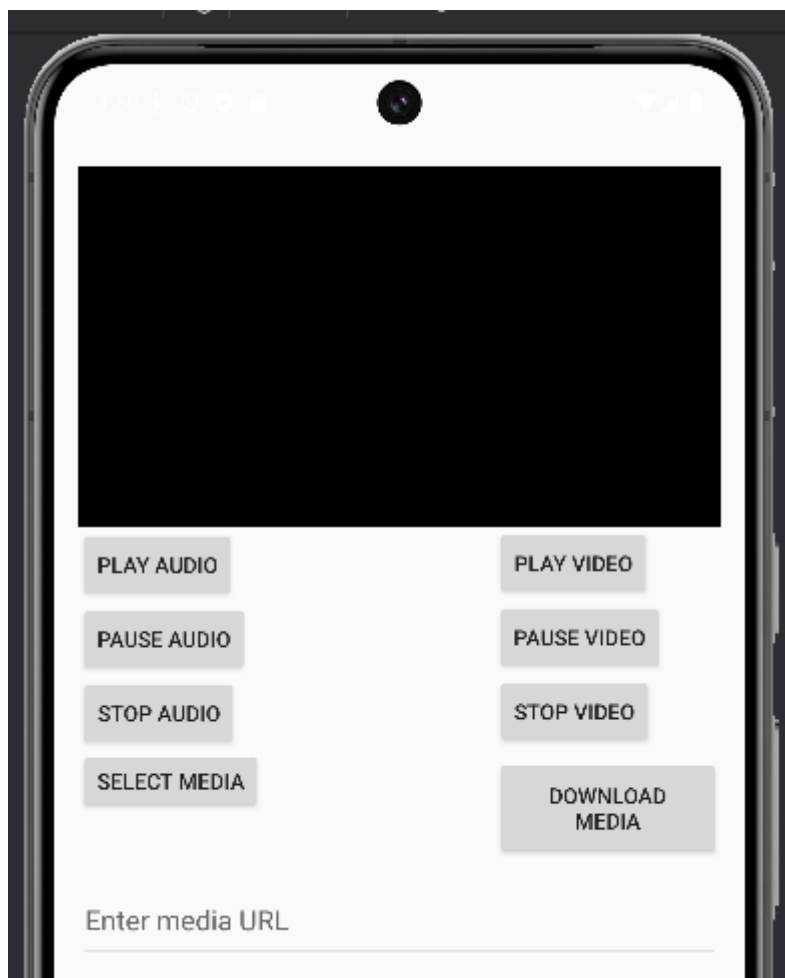
override fun onDestroy() {
    super.onDestroy()
    mediaPlayer.release()
    videoView.stopPlayback()
}
}

```

Скріншоти виконання програми:

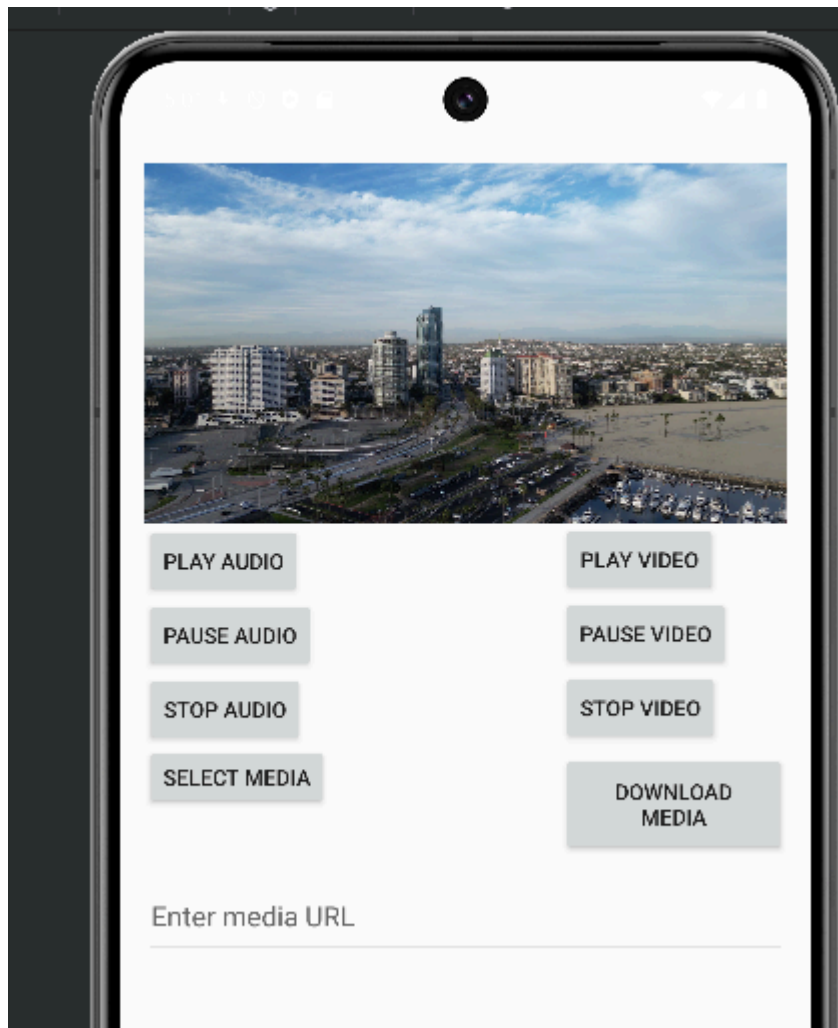
Перевірка базових функцій:

Так виглядає початковий екран без натискання будь-яких кнопок



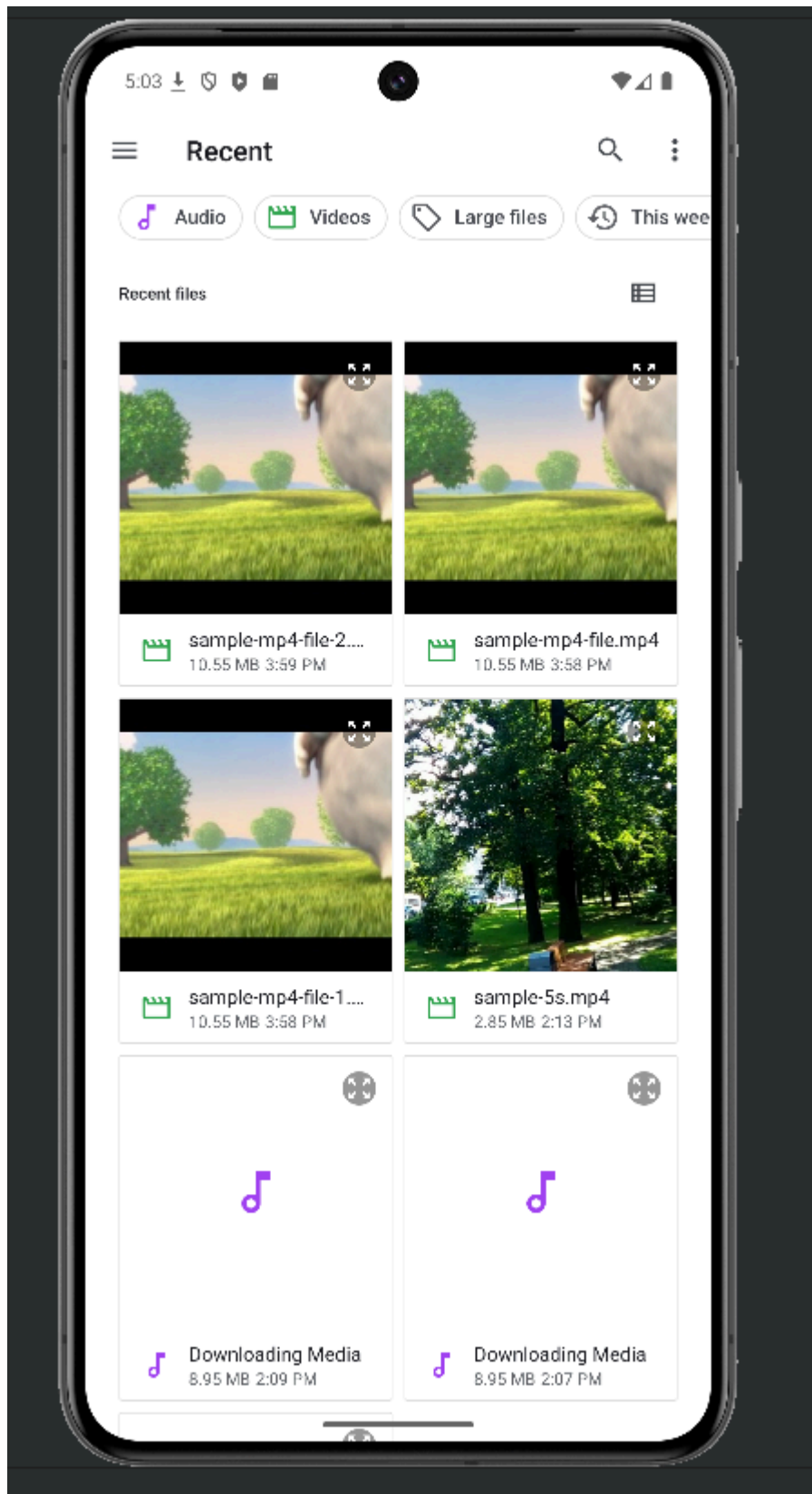
При натисканні кнопки “Play audio” починає грати музика, яка взята із simple_audio.mp3, пауза відповідно призупиняє пісню, а стоп зовсім вимикає, але на play знову можна поновити за бажанням.

При натисканні кнопки “Play video” з’являється замість чорного екрану, яке як і пісню можна призупиняти чи зупинити.



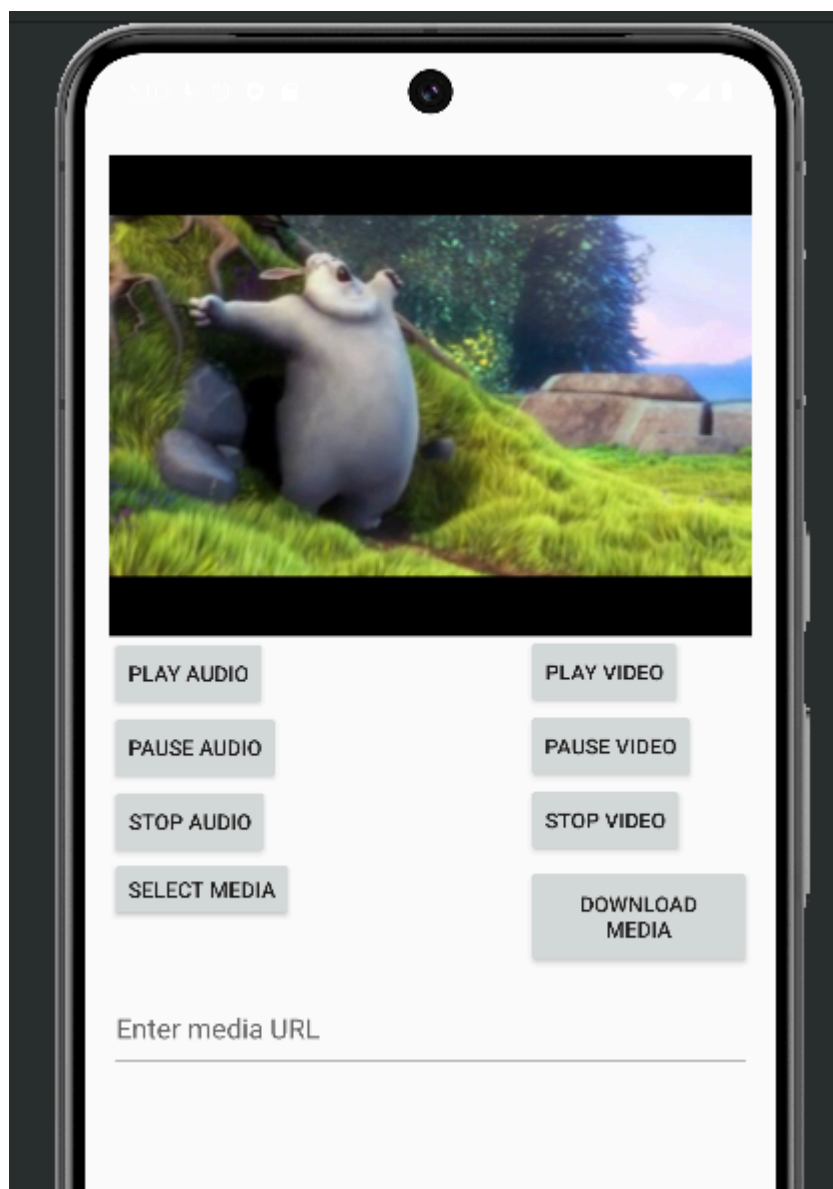
Вибір файлу із будь якого сховища:

У мене вже є скачані файли, тому сховище показує, що ми можемо обрати

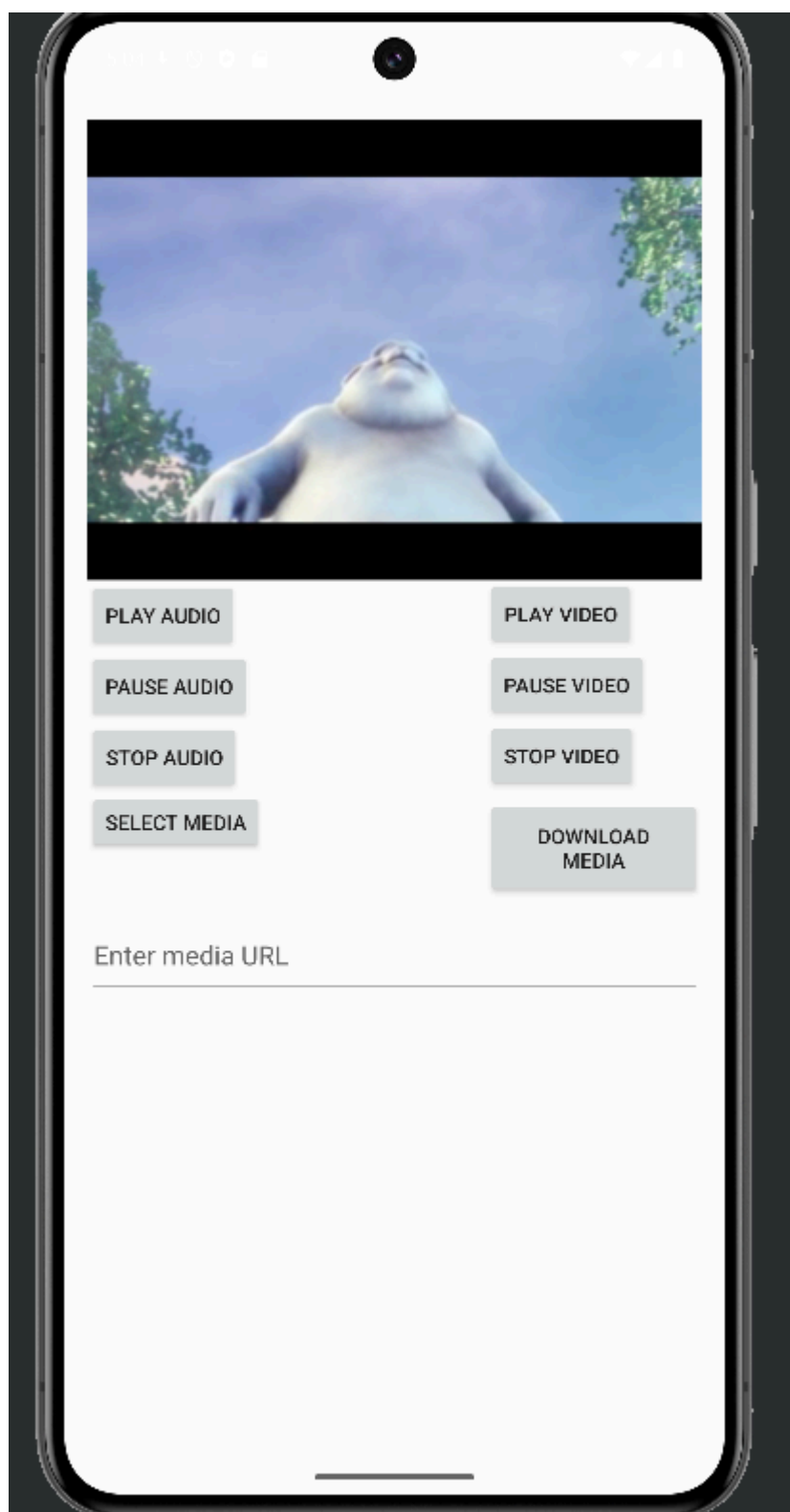


Для прикладу оберемо саме перше відео з галявиною і кроликом

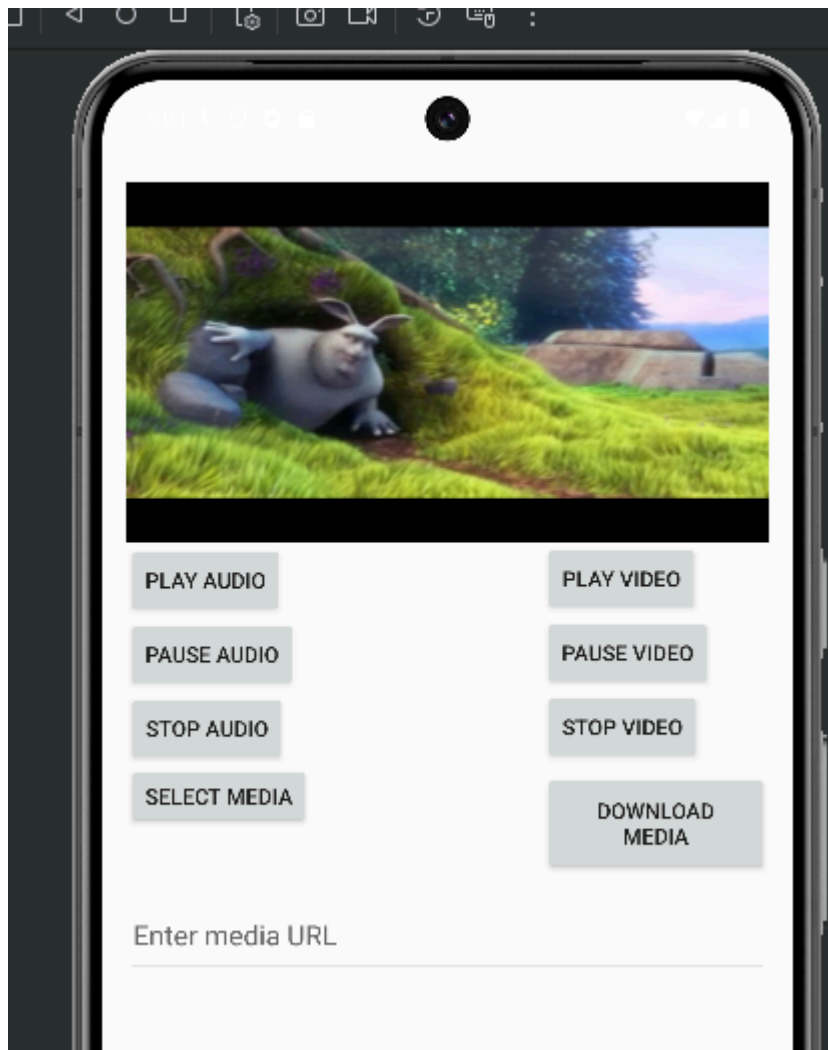
Після того як обрали, натискаємо кнопку “Play video”



Натискаємо “Pause video”:



та натискаемо “Stop video“



Після того як ми натиснемо stop картинка обраного відео стиснеться трішки, але коли ми натиснемо відразу після цього play, запуститься відео з локального сховища. А якщо ви хочете ще раз переглянути відео про кролика, то потрібно заново обирати його через кнопку “Select media”. Аудіофайли працюють так само як і відео.

Завантаження з інтернету:

Потрібно вставити URL-адресу для того щоб скачати щось .mp3 чи .mp4 формату, потім натиснути кнопку “Download media” знизу з’явиться повідомлення що файл почав скачуватись, у емуляторі приходив звук коли файл скачувався і з’являвся у сховищі, яке доступне за кнопкою “Select media”.

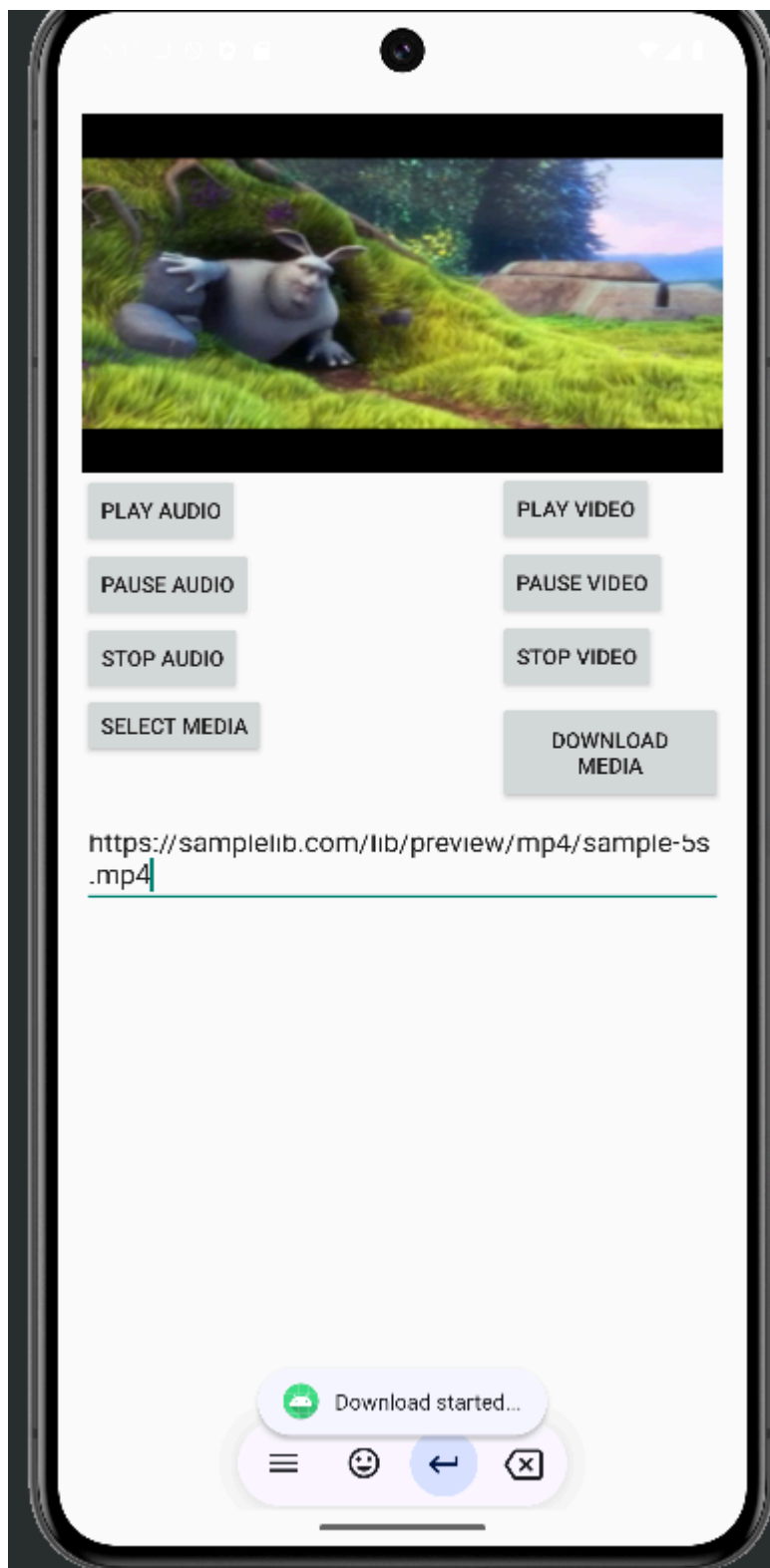
Ось посилання які я використовувала для перевірки програми:

<https://www.soundhelix.com/examples/mp3/SoundHelix-Song-1.mp3>

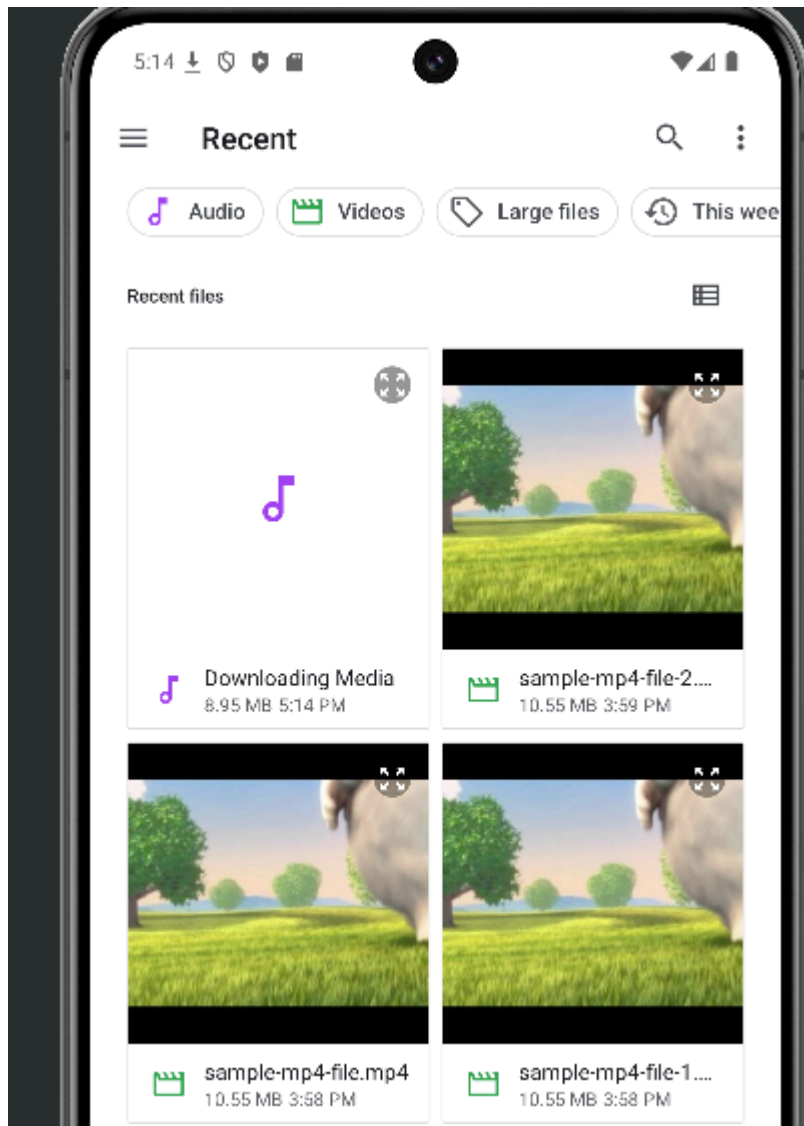
<https://samplelib.com/lib/preview/mp4/sample-5s.mp4>

<https://www.learningcontainer.com/wp-content/uploads/2020/05/sample-mp4-file.mp4>

Скачаємо ще раз будь який з цих файлів:



Після завершення скачування файл з'явився самого верху, перший:



Висновок:

У даній лабораторній роботі було розглянуто, які є можливості для обробки аудіо- та відео-файлів, а також використання інструментів для відтворення медіа-даних. У своєму коді я використовувала кілька інструментів для роботи з медіа, серед яких основними є MediaPlayer та VideoView, які використовуються для програвання аудіо та відео контенту, відповідно. Однак ці інструменти мають обмеження щодо форматів медіа та відсутність розширених можливостей управління відтворенням. Я використала їх, тому що MediaPlayer використовується для відтворення

аудіо та відео та підтримує багато стандартних форматів, а `VideoView` для простого відтворення відео, що дозволяє швидко додати медіаплеєр у додаток. Також читала, що корисно використовувати `ExoPlayer`, але я не стала, тому що використовувала вбудовані інструменти `Android`, які легкі для розуміння новачкам, хоча на мою думку `ExoPlayer` є більш потужним та гнучким рішенням для обробки медіа-контенту, яке дозволяє працювати з різними форматами медіа та має багато додаткових можливостей, його слід використовувати у більш складних проектах, де потрібно обробляти великі медіафайли з хорошою якістю.