IJNRD.ORG ISSN : 2456-4184
International Research Journal IJNRD Research Through Innovation
INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG
An International Open Access, Peer-reviewed, Refereed Journal

QUICK MEDIA GRABBER: EFFORTLESS VIDEO AND AUDIO DOWNLOAD APP

Prof. Parvati Kadli[1], Abhishek Hiremath[2], Dasari Akhil Shankar Rao[3], Dhiraj M[4], Shivakumar[5]
[1]Professor, Department of Computer Science and Engineering, Proudhadevaraya Institute of Technology, Hosapete, Karnataka, India
[2-5]Students, Department of Computer Science and Engineering, Proudhadevaraya Institute of Technology, Hosapete, Karnataka, India.

Abstract : **The research aims to investigate the frequency and challenges associated with downloading audio and video content from various social media platforms. The primary focus is to identify the platforms commonly used for content downloading, any encountered difficulties during the process. Quick Media Grabber is an innovative Android application crafted with user convenience in mind. Developed using Android Studio and featuring a blend of Java and Kotlin programming languages, this app provides a seamless and efficient solution for users to download videos and audio content from a variety of social media platforms. The sleek and intuitive interface of Quick Media Grabber simplifies the content acquisition process, ensuring a hassle-free experience.**

Keywords: Audio and video downloads, social media platforms, Quick Media Grabber, Android app, User convenience.

I. INTRODUCTION

A. Definition

In the fast-paced world of social media, accessing and preserving your favorite audio and video content can be a cumbersome process. Users often find themselves copy-pasting URLs, navigating through various applications, and sometimes even compromising on quality. But there's a better way. Our project has developed a revolutionary solution, that eliminates these obstacles and simplifies the entire process. With just a single tap, you can effortlessly download the content you love from various social media applications. We understand the frustration that arises from complex download procedures and the privacy concerns related to third-party apps. Our solution addresses these issues while ensuring the highest quality, watermark-free downloads. In this introduction, we'll delve deeper into how our innovative technology surpasses existing methods, enhances the user experience, and changes the way you interact with social media content. Designed for efficiency and resourcefulness, Quick Media Grabber allows users to effortlessly navigate the app and initiate downloads with just a few taps. Its versatility extends to compatibility with a wide spectrum of social media platforms, offering users the flexibility to access and save their favorite multimedia content without complications.

B. Problem Statement

Accessing online videos or audio for offline viewing is often hindered on certain platforms, preventing users from downloading content or saving it in their device's gallery. Users encounter intricate barriers, such as the need to navigate through a cumbersome process involving captcha-solving, anonymous page redirects, and persistent advertisements to obtain the desired video or audio files. Moreover, the inability to buffer content smoothly leads to interruptions during online video streaming, causing users to miss specific segments due to playback issues, thereby hindering the seamless viewing experience. These impediments significantly challenge users' efforts to access and enjoy online multimedia content effortlessly and without interruption.

C. Background of the problem

The audio-video downloader project encountered several significant problems during development and deployment. One major issue was the lack of important features, such as batch downloading and subtitle support, which were essential for enhancing user experience and functionality. Despite being advertised, these features either were not available or did not work properly, leading to frustration among users and detracting from the downloader's overall value. Additionally, there were limitations in the formats supported for download, further restricting the downloader's utility and versatility.

Another critical problem was the inability of users to download videos at all. This issue significantly undermined the downloader's primary function and rendered it ineffective for its intended purpose. Moreover, users reported instances where downloads would get stuck or corrupted, resulting in incomplete or unusable files. These technical glitches and failures eroded user trust and confidence in the downloader's reliability and performance.

Furthermore, users experienced frustration due to limited download speed and the downloader's capability to only download low-quality videos. These limitations hampered the user experience and made the downloader less competitive compared to alternative solutions available in the market. Concerns about viruses or malware also arose, raising doubts about the security and safety of using the downloader.

In addition to functionality issues, users expressed concerns about the downloader's ethics and legality. There were reports of the downloader downloading videos from illegal sources, raising ethical and legal questions about its use. Moreover, some users felt that the premium features were too expensive, while the free version was limited or unusable, leading to dissatisfaction among users.

Further exacerbating the problems were hidden charges or subscriptions that users were unaware of until after using the downloader. Non-existent or unhelpful customer support compounded users' frustrations, with long wait times for responses and support staff lacking knowledge or understanding of the problems reported. These issues contributed to a poor user experience and negatively impacted the downloader's reputation and viability in the market.

D. Objectives

- To download files through single tap without lengthy process.
- To develop a user-friendly interface that allows users to easily navigate the application.
- To ensure compatibility with a diverse range of audio and video applications.
- To implement a download queue system that allows downloading process.
- To support various video and audio formats.
- To improve download speed using multithreading.

II. EXPERIMENTAL SETUP

SOFTWARE SETUP

I. Android Studio (Android Studio Iguana | 2023.2.1, Stable):

Android Studio is the official Integrated Development Environment (IDE) for Android app development. Based on the powerful code editor and developer tools from IntelliJ IDEA, Android Studio offers even more features that enhance your productivity when building Android apps, such as:

- A flexible Gradle-based build system
- A fast and feature-rich emulator
- A unified environment where you can develop for all Android devices
- Live Edit to update composable in emulators and physical devices in real time
- Code templates and GitHub integration to help you build common app features and import sample code
- Extensive testing tools and frameworks
- Lint tools to catch performance, usability, version compatibility, and other problems
- Built-in support for Google Cloud Platform, making it easy to integrate Google Cloud Messaging and App Engine.

Each project in Android Studio contains one or more modules with source code files and resource files.

The types of modules include:

- Android app modules
- Library modules
- Google App Engine modules

By default, Android Studio displays your project files in the Android project view, as shown in figure 1. This view is organized by modules to provide quick access to your project's key source files. All the build files are visible at the top level, under **Gradle Scripts**.

Each app module contains the following folders:

- **manifests**: Contains the `AndroidManifest.xml` file.
- **java**: Contains the Kotlin and Java source code files, including JUnit test code.
- **res**: Contains all non-code resources such as UI strings and bitmap images.

The Android project structure on disk differs from this flattened representation. To see the actual file structure of the project, select **Project** instead of **Android** from the **Project** menu.

Gradle build system

Android Studio uses Gradle as the foundation of the build system, with more Android-specific capabilities provided by the Android Gradle plugin. This build system runs as an integrated tool from the Android Studio menu and independently from the command line. You can use the features of the build system to do the following:

- Customize, configure, and extend the build process.
- Create multiple APKs for your app with different features, using the same project and modules.
- Reuse code and resources across source sets.

By employing the flexibility of Gradle, you can achieve all of this without modifying your app's core source files.

Gradle if you use Groovy. They are plain text files that use the Kotlin or Groovy syntax to configure the build with elements provided by the Android Gradle plugin. Each project has one top-level build file for the entire project and separate module-level build files for each module. When you import an existing project, Android Studio automatically generates the necessary build files.

Note: We might reference either the **`build.gradle.kts`** or **`build. Gradle`** file alone in the documentation, but they're conceptually interchangeable. For example, if you see **`build.gradle.kts`** but you use the Groovy DSL to configure your build, you can think of it as the **`build. Gradle`** file (and the other way around).

Build variants

International Research Journal
IJNRD
Research Through Innovation

The build system can help you create different versions of the same app from a single project. This is useful when you have both a free version and a paid version of your app or if you want to distribute multiple APKs for different device configurations on Google Play.

Multiple APK support

Multiple APK support lets you efficiently create multiple APKs based on screen density or ABI. For example, you can create separate APKs of an app for the `hdpi` and `mdpi` screen densities, while still considering them a single variant and letting them share test APK, `javac`, `dx`, and ProGuard settings.

Resource shrinking

Resource shrinking in Android Studio automatically removes unused resources from your packaged app and library dependencies. For example, if your app uses Google Play services to access Google Drive functionality, and you are not currently using Google Sign-In, then resource shrinking can remove the various drawable assets for the `SignInButton` buttons.

Manage dependencies

Dependencies for your project are specified by name in the module-level build script. Gradle finds dependencies and makes them available in your build. You can declare module dependencies, remote binary dependencies, and local binary dependencies in your `build.gradle.kts` file.

Android Studio configures projects to use the Maven Central Repository by default. This configuration is included in the top-level build file for the project.

For more information about configuring dependencies, read Add build dependencies.

Debug and profile tools

Android Studio helps you debug and improve the performance of your code, including inline debugging and performance analysis tools.

Inline debugging

Use inline debugging to enhance your code walkthroughs in the debugger view with inline verification of references, expressions, and variable values.

Inline debug information includes:

- Inline variable values
- Objects that reference a selected object
- Method returns values
- Lambda and operator expressions
- Tooltip values

To enable inline debugging, in the **Debug** window, click **Settings** and select **Show Variable Values in Editor**.

Performance profilers

Android Studio provides performance profilers so you can easily track your app's memory and CPU usage, find deallocated objects, locate memory leaks, optimize graphics performance, and analyze network requests.

To use performance profilers, with your app running on a device or emulator, open the Android Profiler by selecting **View** > **Tool Windows** > **Profiler**.

Heap dump

When profiling memory usage in Android Studio, you can simultaneously initiate garbage collection and dump the Java heap to a heap snapshot in an Android-specific `HPROF` binary format file. The HPROF viewer displays classes, instances of each class, and a reference tree to help you track memory usage and find memory leaks.

Memory Profiler

Use Memory Profiler to track memory allocation and watch where objects are being allocated when you perform certain actions. These allocations help you optimize your app's performance and memory use by adjusting the method calls related to those actions.

Data file access

The Android SDK tools, such as Systrace and Logcat, generate performance and debugging data for detailed app analysis.

To view the available generated data files:

1. Open the Captures tool window.
2. In the list of the generated files, double-click a file to view the data.
3. Right-click any HPROF files to convert them to the standard.
4. Investigate your RAM usage file format.

Code inspections

Whenever you compile your program, Android Studio automatically runs configured lint checks and other IDE inspections to help you easily identify and correct problems with the structural quality of your code.

The lint tool checks your Android project source files for potential bugs and optimization improvements for correctness, security, performance, usability, accessibility, and internationalization.

Annotations in Android Studio

Android Studio supports annotations for variables, parameters, and return values to help you catch bugs, such as null pointer exceptions and resource type conflicts.

The Android SDK Manager packages the Jetpack Annotations library in the Android Support Repository for use with Android Studio. Android Studio validates the configured annotations during code inspection.

II. Android OS(Android version 12 and above):

Android Studio is the official integrated development environment (IDE) for Android app development, provided by Google. It offers a comprehensive set of tools and features to facilitate the creation, testing, and deployment of Android applications. For projects targeting Android OS versions higher than 7.0 (Nougat), Android Studio provides various enhancements and optimizations to support the latest platform features and APIs.

- Enhanced Support for New Features: Android Studio includes support for the latest Android OS features introduced in versions higher than 7.0. This includes improvements in areas such as performance, security, user interface design, and system behavior.
- Compatibility with Latest APIs: Android Studio ensures compatibility with the latest APIs and libraries available for Android development. Developers can leverage these APIs to access new functionalities and capabilities offered by newer Android OS versions.
- Updated Emulator and Device Support: Android Studio offers an updated emulator with improved performance and compatibility with newer Android OS versions. It also provides support for testing on physical devices running the latest Android OS releases.
- Optimization for Performance and Stability: Android Studio incorporates optimizations to enhance the performance and stability of app development workflows. This includes improvements in build times, code analysis, debugging, and deployment processes.
- Compatibility with Support Libraries: Android Studio ensures compatibility with the latest versions of support libraries and compatibility packages, enabling developers to utilize modern app development practices and design patterns.
- Updated Design Tools and Resources: Android Studio includes updated design tools and resources to assist developers in creating visually appealing and user-friendly interfaces for their apps. This includes layout editors, theme editors, and support for Material Design guidelines.

III. Kotlin(Starting from IntelliJ IDEA 2023.3 and Android Studio Iguana (2023.2.1) Canary 15,):

It is an open-source, statically typed programming language supported and developed by JetBrains and open-source contributors.

Some handy links:

- Kotlin Site
- Getting Started Guide
- Try Kotlin
- Kotlin Standard Library
- Issue Tracker
- Kotlin YouTube Channel

Kotlin Multiplatform capabilities

Support for multiplatform programming is one of Kotlin’s key benefits. It reduces time spent writing and maintaining the same code for different platforms while retaining the flexibility and benefits of native programming.

- Kotlin Multiplatform Mobile for sharing code between Android and iOS
- Getting Started with Kotlin Multiplatform Mobile Guide
- Kotlin Multiplatform Benefits
- Share code on all platforms
- Share code on similar platforms

Editing Kotlin

- Kotlin IntelliJ IDEA Plugin (source code)
- Kotlin Eclipse Plugin
- Kotlin Sublime Text Package

Build environment requirements

This repository is using Gradle toolchains feature to select and auto-provision required JDKs from AdoptOpenJdk project.

Unfortunately, AdoptOpenJdk project does not provide required JDK 1.6 and 1.7 images, so you could either download them manually and provide path to installation via JDK_1_6 and JDK_1_7 environment variables or use following SDK managers:

- Asdf-vm
- Jabba
- SDKMAN!

Alternatively, it is still possible to only provide required JDKs via environment variables (see Gradle. Properties for supported variable names).

For local development, if you're not working on the standard library, it's OK to avoid installing JDK 1.6 and JDK 1.7. Add Kotlin. build. isObsoleteJdkOverrideEnabled=true to the local. properties file, so build will only use JDK 1.8+. Note, that in this case, build will have Gradle remote build cache misses for some tasks.

IV. XML

Extensible Markup Language (XML) is a simple, very flexible text format derived from SGML (ISO 8879). Originally designed to meet the challenges of large-scale electronic publishing, XML is also playing an increasingly important role in the exchange of a wide variety of data on the Web and elsewhere.

This page describes the work being done at W3C within the XML Activity, and how it is structured. Work at W3C takes place in Working Groups. The Working Groups within the XML Activity are listed below, together with links to their individual web pages.

The download formal technical specifications here, because we publish them. This is not a place to find tutorials, products, courses, books or other XML-related information. There are some links below that may help you find such resources.

The pages for each Working Group. Each document also contains email addresses you can use to send comments or questions, for example if you have been writing software to implement them and have found problems or errors.

Please do not send us email asking us to help you learn a language or specification; there are plenty of resources online, and the people editing and developing the specifications are very busy. We are interested in technical comments and errata.

Working Groups

There is more detail about each of these Working Groups in the Activity Statement and also on the individual Working Group public web pages.

Most Working Groups have both a public web page and another more private one that is only accessible to W3C Members. The private page has telephone numbers, schedules for meetings and conference calls, links to internal editing drafts, and other administrative information.

XSLT Working Group

The XSLT Working Group is responsible for XSL Transformations (XSLT) and a number of supporting specifications.

The Efficient XML Interchange Working Group

The Efficient XML Interchange Working Group is responsible for developing ways to exchange XML documents in ways that are as efficient as is practical without compromising the interoperability of XML itself. This Working Group is not about producing a closed, proprietary or obfuscated "binary XML"—The W3C is all about increasing interoperability! The EXI format is a compressed stream of parse events that can use an XML Schema to avoid having to transmit known information and to use native type representations. The receiver of an EXI stream doesn't have to reconstitute the original document, but can process the parse events directly as if parsing had happened, saving CPU, memory, time and bandwidth.

XML Query Working Group

The XML Query Working Group is working on the XML Query Language, a way to provide flexible query facilities and processing of forests of trees, typically exchanged using XML or JSON. This includes publication of XQuery and also XPath, in conjunction with the XSLT Working Group.

HARDWARE SETUP

1. **System/Laptop – Windows 7 and Above**
2. **Memory - 8GB RAM & 256GB SSD and above**
3. **Android Smart Phone**

III. DESIGN AND METHODOLOGY

DESIGN

- **Command Module:** The Command Module allowed you to isolate this functionality, making it easier to manage and execute these actions in response to a user's request or certain triggers without directly integrating the copying logic within the application's main structure. By doing this, you've created a modular and reusable way to manage the URL copying operation, enhancing the overall organization and maintainability of your application.
- **Configuration Module:** The Configuration Module in your project serves as a centralized system for managing application settings. It stores critical configurations, like API endpoints and database connections, separate from the core code. This design allows for easy access, modification, and validation of parameters without altering the main application structure. By isolating configurations, it ensures consistency, simplifies updates, and supports quick adaptation to diverse environments.
- **Aria2 Module:** aria2 is a lightweight multi-protocol & cross platform download utility operated in command-line. It supports HTTP/HTTPS, FTP, SFTP, BitTorrent and Meta link. aria2 can download a file from multiple sources/protocols and tries to utilize your maximum download bandwidth. It supports downloading a file from HTTP(S)/FTP/SFTP and BitTorrent at the same time, Using Meta link's chunk checksums, aria2 automatically validates chunks of data while downloading a file like BitTorrent.
- **Format Module**: The Format Module is responsible for formatting data within the app, ensuring that information is displayed in a structured and readable format. It standardizes the representation of data, optimizing its visual presentation for user comprehension.
- **UI Material Design 3 Module:** The UI Material Design 3 Module encompasses the visual aspects of your app, maintaining a consistent and attractive design. It adheres to Material Design guidelines, ensuring a standardized and appealing user interface. By offering consistent UI elements, it improves the overall aesthetic and user experience, creating an intuitive and visually appealing environment for users to navigate and interact with the app's features.
- **Multiple Language Module**: The Multiple Language Module enables your app to support various languages, catering to a diverse user base. By incorporating multiple language options, the app becomes more inclusive, allowing users to interact in their preferred language. This enhances the app's accessibility and usability, broadening its reach to a more global audience and improving user satisfaction by offering content in languages they understand.
- **Splash Screen:** The Splash Screen in your Android app acts as an initial interface upon launch, displaying a branded image or animation. It serves to engage users during the app's loading process, providing a visual cue that the app is launching.
- **Mutagen Module:** Mutagen is a Python module to handle audio metadata. It supports ASF, FLAC, MP4, Monkey's Audio, MP3, Musepack, Ogg Opus, Ogg FLAC, Ogg Speex, Ogg Theora, Ogg Vorbis, True Audio, WavPack, OptimFROG, and AIFF audio files. All versions of ID3v2 are supported, and all standard ID3v2.4 frames are parsed.
- **Downloads Directory Module:** The Downloads Directory Module manages and organizes downloaded content within the app. It provides a structured system for accessing and handling downloaded files, ensuring they are easily locatable and manageable by users. By organizing downloaded content, this module improves user accessibility to their saved files, allowing for a more streamlined and user-friendly experience when interacting with downloaded materials.

- **Media Downloader Module:** A "Download Module" typically refers to a component or feature within software, applications, or web systems designed to enable users to retrieve files or data from a server or a source onto their local device. Download videos and audio files from video platforms supported by yt-dlp (formerly YouTube-dl). This module provides the functionality for users to download various types of content, such as documents, images, videos, software packages, and more.

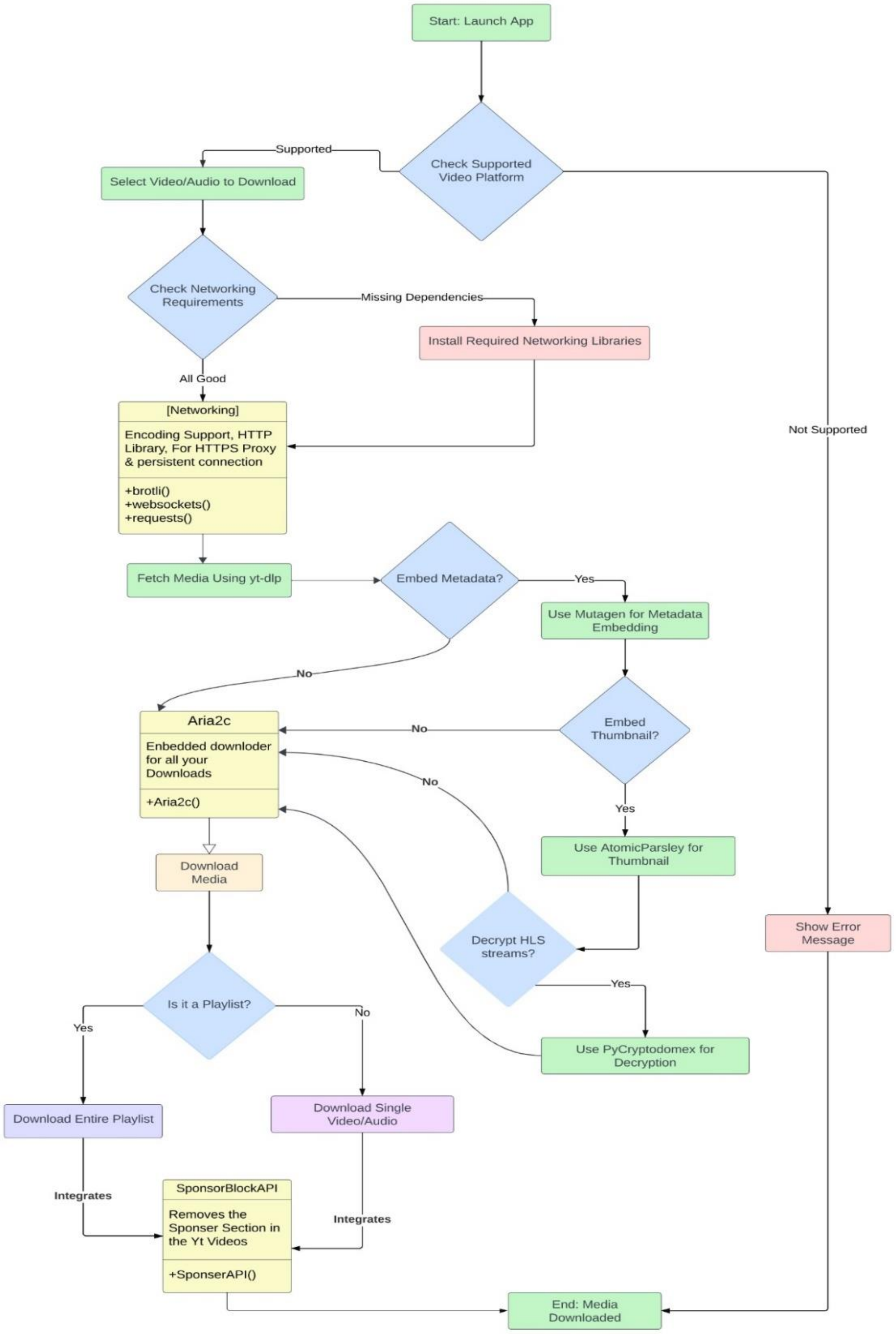

Start: Launch App
Check Supported Video Platform
Supported
Select Video/Audio to Download
Not Supported
Check Networking Requirements
Missing Dependencies
Install Required Networking Libraries
All Good
[Networking]
Encoding Support, HTTP Library, For HTTPS Proxy & persistent connection
+brotli()
+websockets()
+requests()
Fetch Media Using yt-dlp
Embed Metadata?
Yes
Use Mutagen for Metadata Embedding
No
Aria2c
Embedded downloader for all your Downloads
+Aria2c()
No
Embed Thumbnail?
No
Yes
Use AtomicParsley for Thumbnail
Download Media
Decrypt HLS streams?
Yes
Use PyCryptodomex for Decryption
Show Error Message
Is it a Playlist?
Yes
No
Download Entire Playlist
Download Single Video/Audio
Integrates
SponsorBlockAPI
Removes the Sponser Section in the Yt Videos
+SponserAPI()
Integrates
End: Media Downloaded

Figure 1: Architecture Diagram

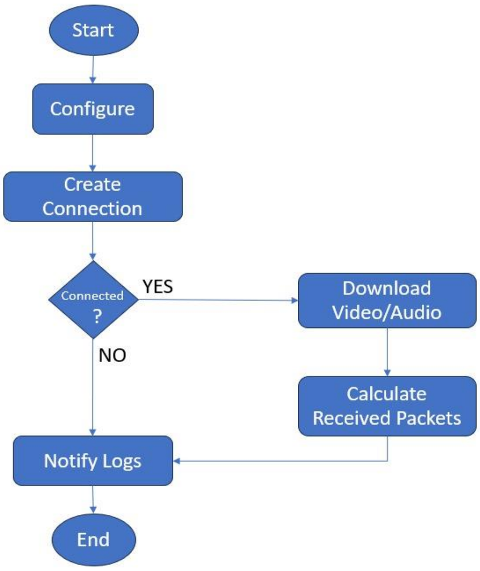


Figure 2: Working model

METHODOLOGY

The methodology followed a systematic approach integrating various key practices. Initially, the project underwent rigorous testing and iteration cycles to identify and rectify any bugs or issues. Comprehensive documentation was maintained throughout the development process to ensure clarity and transparency in project requirements and functionalities. Agile development methodologies were employed, allowing for flexibility and adaptability to changing project needs and priorities. Continuous integration and deployment practices facilitated seamless integration of new features and updates into the project environment.

Performance testing was conducted to assess the efficiency and responsiveness of the downloader, ensuring optimal performance under different conditions. Scalability considerations were taken into account to accommodate potential increases in user base and data volume over time.

Cross-platform compatibility testing ensured that the downloader functioned seamlessly across various devices and operating systems. Robust data backup and recovery strategies were implemented to safeguard against data loss or corruption. Post-launch support and maintenance services were provided to address any issues or concerns that arose after deployment.

User education and support initiatives were undertaken to empower users with the knowledge and resources to effectively utilize the downloader. Performance monitoring tools were employed to track and analyze system performance metrics, enabling proactive identification and resolution of performance-related issues. Adherence to compliance and regulatory standards ensured that the downloader met legal requirements and industry regulations.

Algorithm Steps:

Step 1. Input:
- User provides a URL of the video or audio they want to download.
- User selects the desired format and quality for the downloaded file.

Step 2. Validate URL:
- Check if the provided URL is valid and points to a supported video or audio source.

Step 3. Download Metadata:
- Fetch metadata (e.g., title, duration, size) of the video or audio file from the URL.

Step 4. Select Download Format:
- Based on user preferences, choose the appropriate format and quality for the downloaded file.

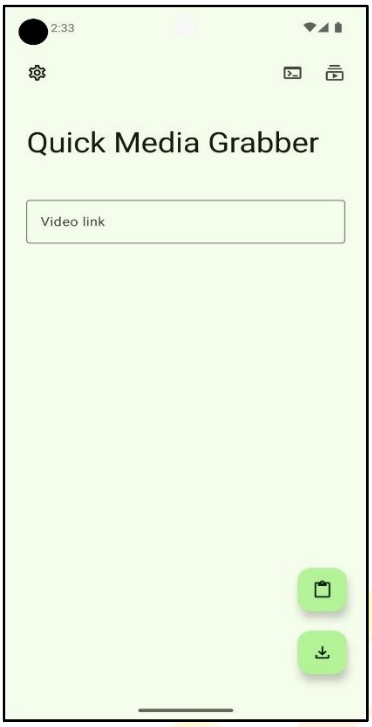
Step 5. Initiate Download:
- Start downloading the video or audio file from the URL to the local storage of the device.

Step 6. Track Download Progress:
- Monitor the download progress to provide real-time feedback to the user (e.g., progress bar).

Step 7. Handle Errors:
- Handle any download errors or interruptions (e.g., network issues) and provide appropriate error messages to the user.

Step 8. Save Downloaded File:
- Once the download is complete, save the downloaded video or audio file to the local storage of the device.

Step 9. Provide Completion Notification:
- Notify the user when the download is finished and provide options to open or share the downloaded file.

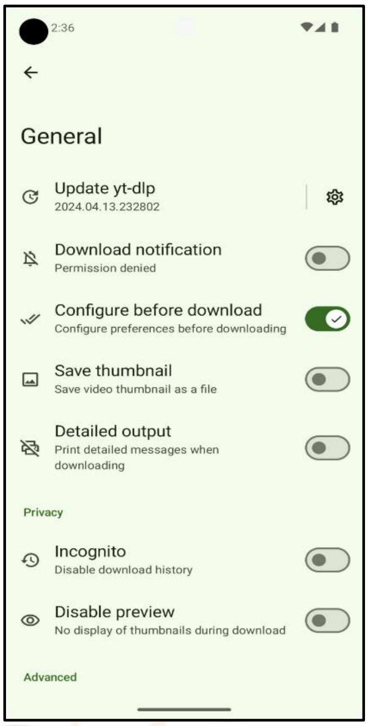


2:33

Quick Media Grabber

Video link

Figure 1: Home Page



2:36

General

Update yt-dlp
2024.04.13.232802

Download notification
Permission denied

Configure before download ✓
Configure preferences before downloading

Save thumbnail
Save video thumbnail as a file

Detailed output
Print detailed messages when downloading

Privacy

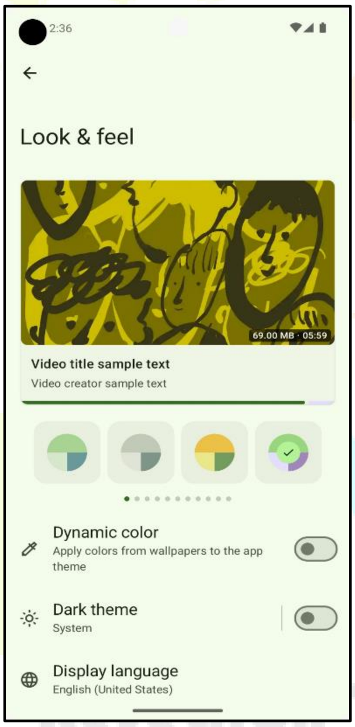
Incognito
Disable download history

Disable preview
No display of thumbnails during download

Advanced

Figure 2: General Setting Page

Figure 3: Theme Page



2:36

Look & feel

69.00 MB · 05:59

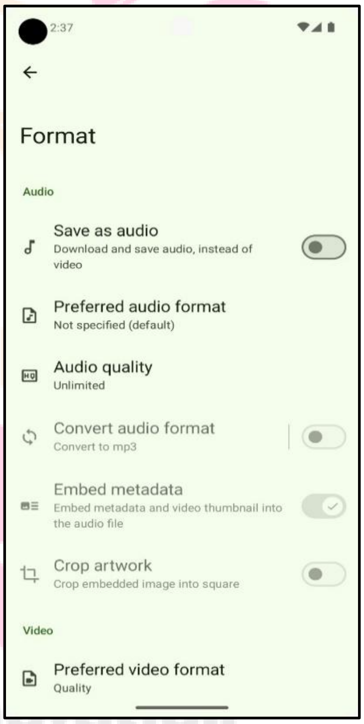
Video title sample text
Video creator sample text

Dynamic color
Apply colors from wallpapers to the app theme

Dark theme
System

Display language
English (United States)

Figure 4: Format Page



2:37

Format

Audio

Save as audio
Download and save audio, instead of video

Preferred audio format
Not specified (default)

Audio quality
Unlimited

Convert audio format
Convert to mp3

Embed metadata ✓
Embed metadata and video thumbnail into the audio file

Crop artwork
Crop embedded image into square

Video

Preferred video format
Quality

IV. CONCLUSION

Our app is designed to solve the problem of ease and convenience when it comes to downloading audio and video content from various social platforms. In today's digital age, social media has become a primary source of entertainment, information, and communication for millions of people worldwide. Users frequently encounter content on applications such as Instagram, YouTube, Facebook, and more that they wish to save and access later. However, the process of downloading this content can be cumbersome, involving multiple steps and sometimes the use of third-party applications. This is where your app steps in to streamline and simplify the process.

One of the main issues our app addresses is the time-consuming and complicated nature of downloading audio and video content from social platforms. In the absence of a dedicated solution, users often resort to a variety of workarounds, like screen recording, which may result in a loss of quality or watermarked content. Additionally, the use of third-party apps can raise privacy and security concerns, as users must grant these applications access to their accounts and personal data.

V. ACKNOWLEDGMENT

Without acknowledging the people who made the project possible, without whose unwavering support and guidance our efforts would be considered vain., the happiness that comes with its successful completion would be incomplete. As we completed our project on "**QUICK MEDIA GRABBER: EFFORTLESS VIDEO AND AUDIO DOWNLOAD APP**" we felt delighted to thank and show thanks to everyone who helped us along the way. We appreciate all of the help and inspiration we received along the way from our guide, **Prof. Parvati Kadli**, Professor and Head of 'CSE'. We are grateful to the management and principal of Proudhadevaraya Institute of Technology Hosapete **Dr. Rohitha U.M**, for providing us with the facilities and welcoming atmosphere that have allowed us to further our education. We'd also want to mention all of the teaching and non-teaching staff of the 'CSE'. We'd want to mention. our parents and friends for their gracious cooperation and support over the project's duration.

VI. REFERENCES

[1] https://github.com/yt-dlp/yt-dlp
[2] https://mutagen.readthedocs.io/en/latest/
[3] https://github.com/aria2/aria2/
[4]https://developer.mozilla.org/en-US/docs/Web/XML/XML_introduction
[5] https://developer.android.com/studio
[6] https://kotlinlang.org/docs/releases

International Research Journal
IJNRD
Research Through Innovation