iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 12 **Issue:** IV **Month of publication:** April 2024

DOI: https://doi.org/10.22214/ijraset.2024.59663

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue IV Apr 2024- Available at www.ijraset.com

Desktop Stream

Mrs. S. P. Kakade[1], Sanket Sinabhau Mane[2], Sayama Dilawar Mujawar[3], Vaishnavi Ashok Patil[4], Pranjali Pradip Patil[5]
Department of Computer Science and Engineering Dr. Daulatrao Aher College of Engineering, Karad, India

Abstract: ***The "Desktop Stream" approach is all about breaking barriers between Android devices and desktop computers. It lets you easily access your desktop from your Android, wherever you are. By combining Python for the desktop side and Java for Android, it brings mobility, accessibility, and multi-device collaboration to a whole new level. It uses screen sharing, allowing multiple Android devices to peek at your desktop at the same time. Thanks to a middleware server powered by Firebase and Ngrok, you can work together in real-time, share stuff, and get things done. This approach is perfect for today's world, where remote work and teamwork are crucial. It's built to be user-friendly, secure, and scalable. It turns your desktop into a super-portable resource that you can access from anywhere. "Desktop Stream" is not just a cool idea; it's a practical solution that makes remote desktop access a breeze. In this report, we'll dive into how it works, how we made it happen, and why it's a game-changer for remote collaboration.***
Keywords: Remote Desktop Access, Android Application, Firebase, Ngrok, Download desktop files

I. INTRODUCTION

"Desktop Stream" is a ground-breaking approach that redefines how you interact with your desktop computer using Android devices. It's all about making your desktop accessible from anywhere, seamlessly bridging the gap between your desktop and Android. By combining Python for the desktop and Java for Android, this approach offers a new level of mobility and collaboration. One of the most exciting features is screen sharing, which allows multiple Android devices to access your desktop simultaneously. And how does it all come together? We make use of a middleware server, powered by Firebase and Ngrok, to create a real-time connection. This means you can share data, collaborate with others, and be more productive, no matter where you are. In this report, we'll delve into the methodology, implementation, and results of "Desktop Stream," showcasing its potential as a ground-breaking solution for remote desktop access and cross-device collaboration. It's about turning your desktop into a truly mobile resource, accessible at your convenience. And yes, behind the scenes, we're using sockets to make it all work seamlessly.

II. LITERATURE SURVEY

To develop "Desktop Stream," we conducted a literature survey to understand the existing solutions and technologies in the field of remote desktop access and screen sharing. We reviewed several papers and articles related to remote access software and mobile screen sharing. Some of the notable references include:

A. Surendar Chandra, Jacob T. Biehl, John Boreczky, Scott Carter, Lawrence A. Rowe FX Palo Alto Laboratory Inc.,"Understanding Screen Contents for Building a High Performance, Real Time Screen Sharing System"

Loyal sharing of screen contents is a significant cooperation highlight. Earlier frameworks were intended to work over compelled networks. They performed ineffectively even without such bottlenecks. To fabricate an elite exhibition screen sharing framework, we observationally dissected screen contents for various situations. We showed that screen refreshes were irregular with significant stretches of inertia. At the point when dynamic, screens were refreshed at far higher rates than was upheld by before frameworks. The crisscross was articulated for intuitive situations. In any event, during dynamic screen refreshes, the quantity of up-dated pixels were every now and again little. We demonstrated the way that significant data can be lost assuming individual updates were blended. At the point when the accessible framework assets couldn't uphold high catch rates, we showed manners by which updates can be successfully imploded. Wes howed that Zlib misfortune less pressure performed ineffectively for screen refreshes. By examining the screen pixels, we fostered a reasonable change that significantly further developed pressure rates. Our framework caught 240 updates each second while just utilizing 4.6 Mbps for intuitive situations. In any case, while playing films in full screen mode, our methodology couldn't accomplish higher catch rates than earlier frameworks; the computer processor stays the bottleneck. A framework that in-corporates our findings is sent inside the lab.

©IJRASET: All Rights are Reserved

B. *Mounish Chennupati, SriRam Malladi, Prof. Swarnalatha P, VIT, Vellore, Tamil Nadu, India. "Survey on Monitoring PCs using Android"*

The ventures created for this application can be utilized to screen or control PC utilizing an Android gadget (generally a cell phone). For the most part we work our PC's utilizing gadgets like Mouse, Console and so on, so when one of these gadgets switches off there will be challenges in working our PC. These tasks help us in such circumstances by giving us an additional gadget to work. Besides we can work PC keeping it at a couple of distance from us which ceases us from utilizing an additional remote gadget. The PC gets message from the telephone in regards to the activities to be performed. Our study is utilized to examine the different advances and renditions of Android that were utilized to control or screen PC'.

C. *Tejas Patil1, Atharva Pawar2, Sahil Yadav3, Aju Palleri ,Pillai College of Engineering, New Panvel "Research and Analysis of Smart Mirror"*

The venture shows the plan of a brilliant gadget Savvy Mirror. The savvy reflect here is essentially for home climate. These savvy mirrors are not broadly utilized because of cost or high prerequisites of equipment. The proposed brilliant mirror will be worked by Raspberry Pi and will be associated by true through web. The brilliant mirror will comprise Raspberry-Pi, Drove screen, speakers, amplifier with two-way mirror and acrylic glass. With the assistance of voice acknowledgment Programming interface the mirror will speak with the client through voice orders and answers them likewise. The mirror will feature a few fundamental conveniences like time, nearby news, climate, and so forth. The mirror will likewise carry out some development role, for example, Home Mechanization utilizing Smart Mirror. This mirror with computerized reasoning will give an unprecedented encounter to the client.

III. SYSTEM ARCHITECHTURE

In this system, the user has to install "Desktop Stream" mobile application & register with the unique username & password. Also for sharing the desktop screen, execute the code on desktop & then the desktop application will displayed on the screen. Enter the registered username & password to get the unique key. Copy the unique key. Now to mirror the desktop screen on the mobile, login to Desktop Stream mobile application with the registered username & password. Paste the copied unique key in the mobile application. The desktop will get mirrored on the mobile if the key matches to the key generated on the desktop application. Also user can download the files present on desktop from the existing Desktop Stream mobile application.

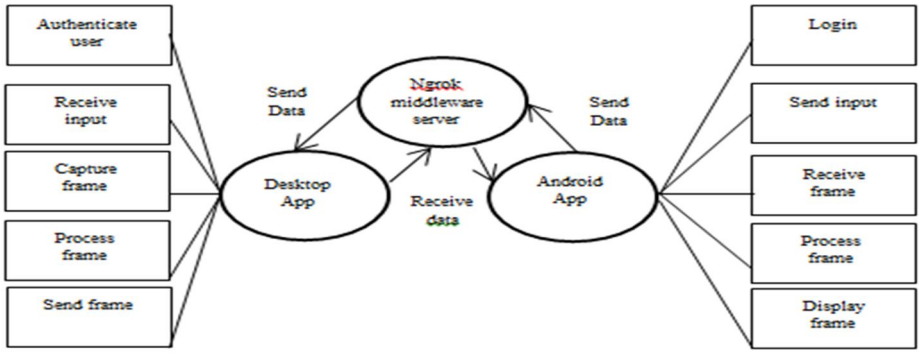


Fig.1: System Architecture

1) The software requirements for the "Desktop Stream" project include:

a) Desktop Component:

- Python: The desktop component must run on a computer with Python installed to manage screen sharing and communication.
- Firebase SDK: Integration of Firebase for real-time data exchange.
- Ngrok: Use of Ngrok for tunneling to make the desktop accessible from outside networks.

b) Android Application:

- Java: The Android application is developed using Java.
- Android Studio: Android Studio is required for Android application development.
- API: Android 8.0 or above requires for this project having minimum sdk version is 33
- Firebase SDK for Android: Integration of Firebase for real-time communication in the Android app.

c) Middleware Server:

- Firebase: Firebase serves as the authentication for real-time authentication for the desktop as well as Android devices.
- Ngrok server: Ngrok serves as the middleware for real-time data transfer between the desktop and Android devices.

2) The hardware requirements for the "Desktop Stream" project include:

a) Desktop Component:

- Desktop Computer: A computer with a minimum of 512MB RAM and running Python.
- Internet Connection: A stable internet connection is required for data sharing and access.

b) Android Devices:

- Android Phones/Tablets: Android devices with a touch screen interface.
- Internet Connection: Android devices must have internet connectivity for accessing the desktop remotely.

3) The functional requirements for the "Desktop Stream" project are as follows:

a) Screen Sharing Mechanism:

- The system must be able to capture the desktop screen.
- The captured desktop screen should be shared with multiple Android devices.
- The screen sharing must occur in real time to ensure seamless access.

b) Python and Java Integration:

- Integration of Python for the desktop component.
- Development of a user-friendly Android application using Java.

c) Middleware Server:

- Integration of Firebase and Ngrok to create a middleware server.
- The middleware server must facilitate real-time communication between the desktop and Android devices.

d) Multiple Concurrent Access:

- The system should allow multiple Android devices to access the desktop simultaneously.
- Each Android device should be able to view and interact with the desktop screen independently.

e) Cross-Device Collaboration:

- Users should be able to efficiently perform tasks on the desktop from their Android devices.
- The system should enable collaborative work, including data sharing and interaction.

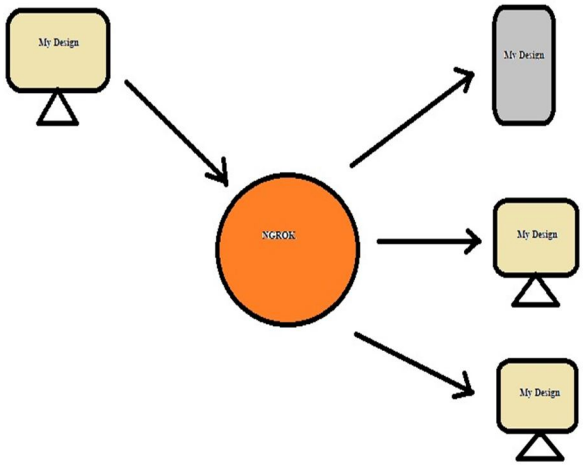


Fig.2: Ngrok

Ngrok -

Ngrok is a highly secure platform for remote access and is safe to use. Ngrok provides tunneling, as mentioned above, which enables users to access local-hosted servers from outside the machine. ngrok is able to bypass NAT Mapping and firewall restrictions by creating a long-lived TCP tunnel from a randomly generated subdomain on ngrok.com to the local machine.

After specifying the port that your web server listens on, the ngrok client program initiates a secure connection to the ngrok server and then anyone can make requests to your local server with the unique ngrok tunnel address.

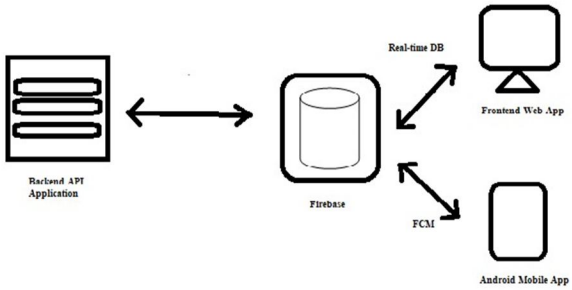


Fig.3: Firebase

Firebase –

Firebase gives mobile developers access to a complete range of fully managed mobile-centric services including analytics, authentication and Realtime Database. Cloud Functions rounds out the offering by providing a way to extend and connect the behavior of Firebase features through the addition of server-side code.

Security and data validation are available through the Firebase Realtime Database Security Rules, expression-based rules that are executed when data is read or written.

IV. IMPLEMENTATION DETAILS

Modules and Their Functionalities

A. Desktop Modules (in Python):

1) Socket Module:

Functionality: Responsible for establishing and managing socket connections, enabling real-time communication between the desktop and Android devices. It plays a vital role in transmitting data, including screen captures and user interactions.

2) SHA256 Module (or Library):

Functionality: Implements the SHA-256 hashing algorithm for data integrity and security. It ensures that data transferred between devices remains unchanged and confidential.

3) Firebase Authentication:

Functionality: Utilizes Firebase for user authentication and authorization. Firebase Authentication adds a layer of security to the system, allowing only authorized users to access the desktop remotely.

4) Threading Module:

Functionality: Facilitates multithreading in Python, enabling concurrent execution of various tasks within the project. It enhances system performance by managing multiple processes efficiently.

5) Ngrok Module:

Functionality: Integrates with Ngrok, a tool for secure tunneling of local network services. Ngrok ensures that the middleware server can communicate securely with Android devices over the internet.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

6) PyAutoGUI Module:

Functionality: Provides automation capabilities for controlling the desktop environment. It aids in capturing and sharing the desktop screen with Android

B. Android Modules (in Java):

1) Socket Module:

Functionality: Similar to the Python socket module, the Android socket module manages socket connections for real-time data transfer between Android devices and the desktop.

2) Threading Module:

Functionality: Enables multithreading in the Android application, allowing concurrent execution of tasks such as receiving screen captures and user interactions.

3) Bitmap Module:

Functionality: Handles the encoding and decoding of bitmap images. This is crucial for receiving and displaying screen captures from the desktop on Android devices.

4) SHA256 Module (or Library):

Functionality: Implements the SHA-256 hashing algorithm on the Android side for data integrity and security, ensuring that received data remains unaltered during transmission.

C. Ngrok:

Ngrok plays a crucial role in the "Desktop Stream" project as a secure tunneling solution for local network services. Its primary function is to create a secure tunnel between a public endpoint and a local network service, allowing external devices to access the local service securely over the internet. Here is a concise overview of Ngrok's role in the project:

1) Secure Tunneling**:** Ngrok establishes a secure and encrypted tunnel from a public endpoint to a local network service. This ensures the confidentiality and integrity of data transferred between the desktop and Android devices.
2) Dynamic DNS**:** Ngrok provides dynamic DNS services, assigning a unique and accessible domain name to the local service. This domain name remains consistent, even if the desktop's IP address changes, making it easy for Android devices to connect over the internet.
3) Middleware Server**:** In the "Desktop Stream" project, Ngrok hosts a middleware server responsible for routing data between the desktop component (Python) and the Android application, facilitating real-time screen sharing.
4) Internet Accessibility: Ngrok enables the local network service (middleware server) to be accessible over the internet. This feature is essential for remote access, as Android devices can connect to the server from anywhere.
5) Security and Authentication: Ngrok adds an additional layer of security by requiring authentication, ensuring only authorized Android devices can connect to the desktop.
6) Ease of Use: Ngrok offers a user-friendly command-line interface (CLI) for configuration and management, simplifying the process of creating and managing secure tunnels

V. RESULT & DECISION

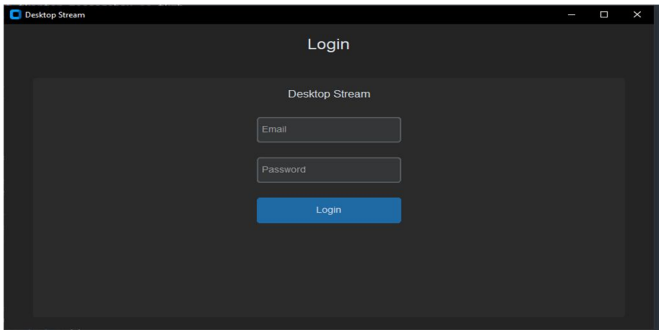


Fig.4: Desktop Stream Login Page

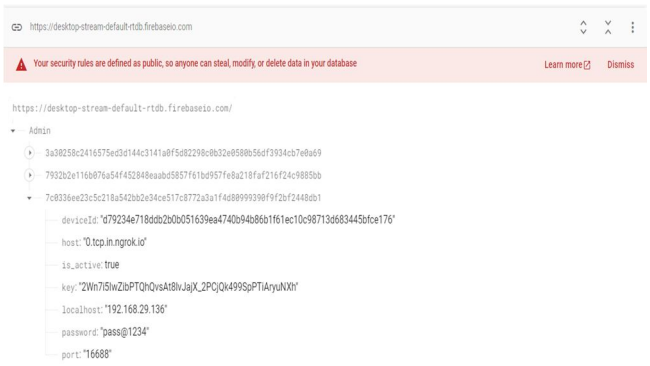


Fig.5: Firebase Console

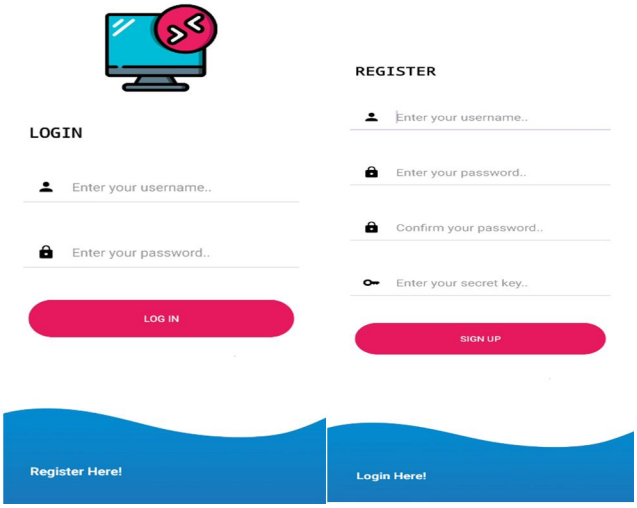


Fig.6: Android User Login & Registration

VI. FUTURE SCOPE

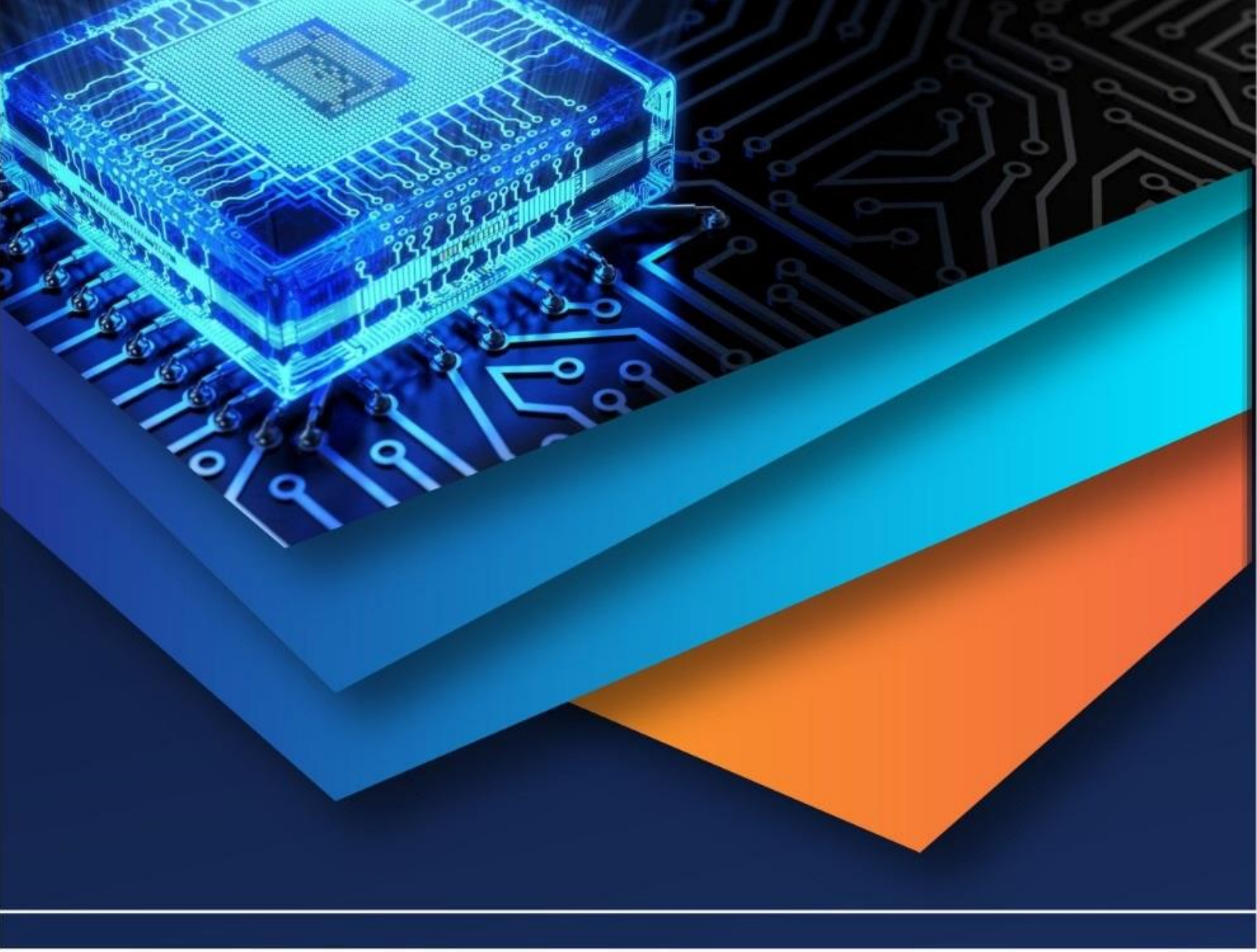
For the Future work, the Desktop Stream can also provide the fastest, efficient implementation on the Mac operating system.

VII. CONCLUSION

The "Desktop Stream" is to enable cross-device collaboration, allowing multiple Android devices to concurrently access the desktop. This capability greatly enhances productivity and flexibility, catering to the evolving requirements of users in today's dynamic digital landscape. It offers a dependable and effective platform for remote desktop access, setting new benchmarks for cross-device collaboration.

REFERENCES

[1] "Screen Sharer": Application for Android Mobiles -Yadav Amar, Phulari Aniket, Yelpale Sachin, Thorat
[2] Mirroring of Source and Sink Devices in Android Screen Casting - Sahil S. Sayyad, Yogesh P. Mourya, Hariom B. Saw, Prof. Uma Goradiya
[3] Survey on Monitoring PCs using Android - praneeth marreddy
[4] ApowerMirror review 2023: a solution to mirror screen to TV – Sabrina
[5] TeamViewer Screen Share: Review and Guide in 2023 – Charudatta
[6] Understanding Miracast as a Wireless Display Technology - Sumit Negi
[7] Understanding screen contents for building a high performance, real time screen sharing system - Surendar Chandra, Jacob T. Biehl, John Boreczky, Scott Carter
[8] Research and Analysis of Smart Mirror - Tejas Patil, Atharva Pawar , Sahil Yadav , Aju Palleri



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)