iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 10 **Issue:** XI **Month of publication:** November 2022

DOI: https://doi.org/10.22214/ijraset.2022.47461

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 10 Issue XI Nov 2022- Available at www.ijraset.com

File Pioneer

Addanki Abhaya Vikas[1], Ammanamanchi Ravisankar Bharadwaj[2], Jonnadula Narasimharao[3]
[1]*B.Tech Student, Department of Computer Science and Engineering, CMR Technical Campus, Medchal, Hyderabad, Telangana, India*
[2]*B.Tech Student, Department of Computer Science and Engineering, CMR Technical Campus , Medchal, Hyderabad, Telangana, India*
[3]*Associate Professor, Department of Computer Science and Engineering, CMR Technical Campus, Medchal, Hyderabad Telangana, India*

Abstract: Automation drives us in a safe and faultless way where we can reach the roots of accuracy. There are some fields where there is a demand to be automated. Automation plays a significant role in Wireless Fidelity. Data Transmission also plays a vital role in the Automation of files. Transferring data from one device to another throws them into a problematic situation where we require HDMI Cables, Ports, Sockets, etc. This makes the transfer of data[1] in the wired systems. And this process is also time-consuming and works manually. And the transferred data could not be in the desired format. If the gap between the device is not in range then the transfer remains paused. This bounds the scope of transfer for data. And the vast amount of data required for data transactions. This makes the extermination of the data.
Keywords: Security, Formatting, NodeJS, ExpressJS, npm(node package manager), SCP (Secure Copy Protocol)

I. INTRODUCTION

Technology makes its path in a mechanized direction. where it is independent of its nature, every sector in this modern era is driving into an automated path where it makes tasks easy. It isn't easy to transfer files automatically without connecting them to any peripheral device. It is mandatory to automate this field such as the files being transferred from one device to another without involving subordinates. And in case, the requirements are in one format and we receive files in another. Then we should look after other websites or any applications to change the format of the data in the file. This throws the user into a stubborn way where he injects some malicious functions without prior knowledge which can kill his system through some bitter websites or applications.
Securing the user's data also plays a crucial role in transferring the file in a network any miscellaneous act can make data loss which drives the user into diplomatic situations where he may lose all his sensitive data

II. LITERATURE SURVEY

Saima Iqbal et al. have experimented with BLOWFISH to transfer files securely using randomly generated keys. Blowfish symmetric algorithms are used for the encryption process. A unique transaction id is also generated to fetch the random key that will be used for decryption at the receiver end. A validation key is also proposed which is used to validate that file is decrypted by the intended user only. As proposed, combining the random key, validation key, and transaction id will provide more robustness to secure the transfer of files using Blowfish symmetric key encryption.
Saurabh Malgaonkar et al.have experimented with distributed files sharing management: A file-sharing application using distributed computing concepts which uses to transverse a file from one node to another node quickly
Bhubaneshwar Prasad Sharma et al have experimented with NodeJS Quick Learning which provides us a path for creating a space and architecture to work on a file system.

III. PROPOSED SYSTEM

We propose a file transfer system, a server-based[1] application that uses the wifi as a medium, In which the server handles both the upload and download of the files. A user can upload multiple files and download files in the desired format. A user can access the server until he is in the range of the wifi range. The data can be used all over the network Until the network is strong and long-range server access can be done in no time. It follows the SCP Protocol in which the file sharing is between the hosts if they are connected or members[4] of a network. Then also the file can be done securely[2][3], so the user can download the file safely

© IJRASET: All Rights are Reserved |

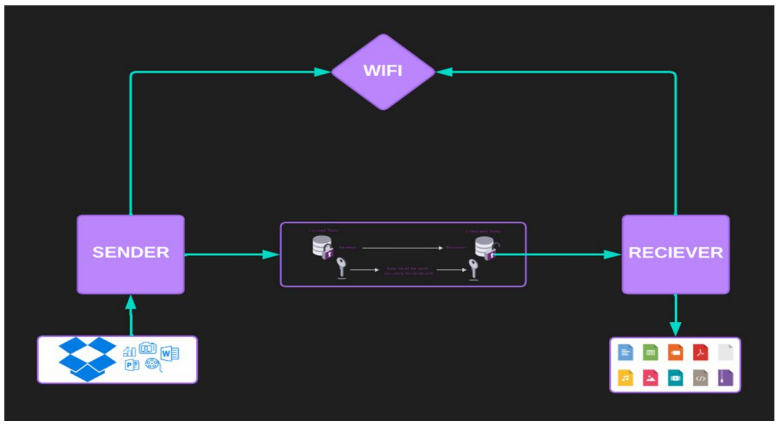


Fig.1: Diagram of Architecture

A. *Required Component*

The following are the required components:

1) *NodeJs:* NodeJs is the open source and javascript runtime environment that provides the functionality to run an application inside an operating system. We used it as the backend for accessing the System's local storage.
2) *ExpressJs:* It is a Javascript framework that provides the server functionalities for running the application. Expressjs provides the routing feature. and It acts as middleware. Nodejs uses Expressjs to adapt the functionality of the server.
3) *EJS(Embedded JavaScript):* EJS provides the javascript functionality using tags. It uses the EJS extension but also we can use the HTML inside of the page and when we want a function we can directly dump the javascript program into the file and also it is fastly executable.
4) *NPM(Node Package Manager):* NPM is a package manager, and it is completely open source, in which we can download the packages with functionalities so that we can adapt to different tasks in a single program. In some cases, it provides execution, and security, and helps to access online APIs.
5) *Libreoffice-convert-win:* It is an npm package that uses the installed LibreOffice software to format the office formats to the pdf format, Nodejs have access and this npm package connects it with the NodeJs functionality to get the desired output
6) *Fluent-FFmpeg:* It is an npm package that uses the installed ffmpeg.exe file after setting its path to the system variables, it is completely used to format the mp4 or video to audio.
7) *JIMP:* JIMP is a NodeJs feature used to format the image to jpeg or jpg, we o need to download it, it is a pre-installed package in node.js

B. *Connections*

Expressjs is the backbone of building the server and NodeJs combines all the packages and their methods to add more features to the server.EJS serves as the front end for the application. and the execution is done on the computer. The server uses local storage when the user sends the data stored on the local disk. And when the computer is connected to the wifi and all the users are connected to the same medium then it can access the server to get their desired output.

C. *Working Procedure Of The Model*

The following are the steps involved in starting the server:

1) *Set up Wifi:* Turn on the wifi in the devices in which the server gets to run and connect the server to a network that includes all the remaining devices.
2) *User Interface:* After the wifi setup, the user needs to type their IP address with an add-on of port(3000) to the default browser which directs to the index page, after login into the server we can directly access its features.
3) *Upload and download:* The browser opens the upload if the user is the sender, then the user uploads multiple files that get stored in the local disk and messages as success with a secure key which is later given to the receiver. If the browser opens the download page then the user is the receiver. After typing the secure key in the given input field it provides the files that the user gives. In that, it provides the format as well, so the receiver can download what he wants.

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

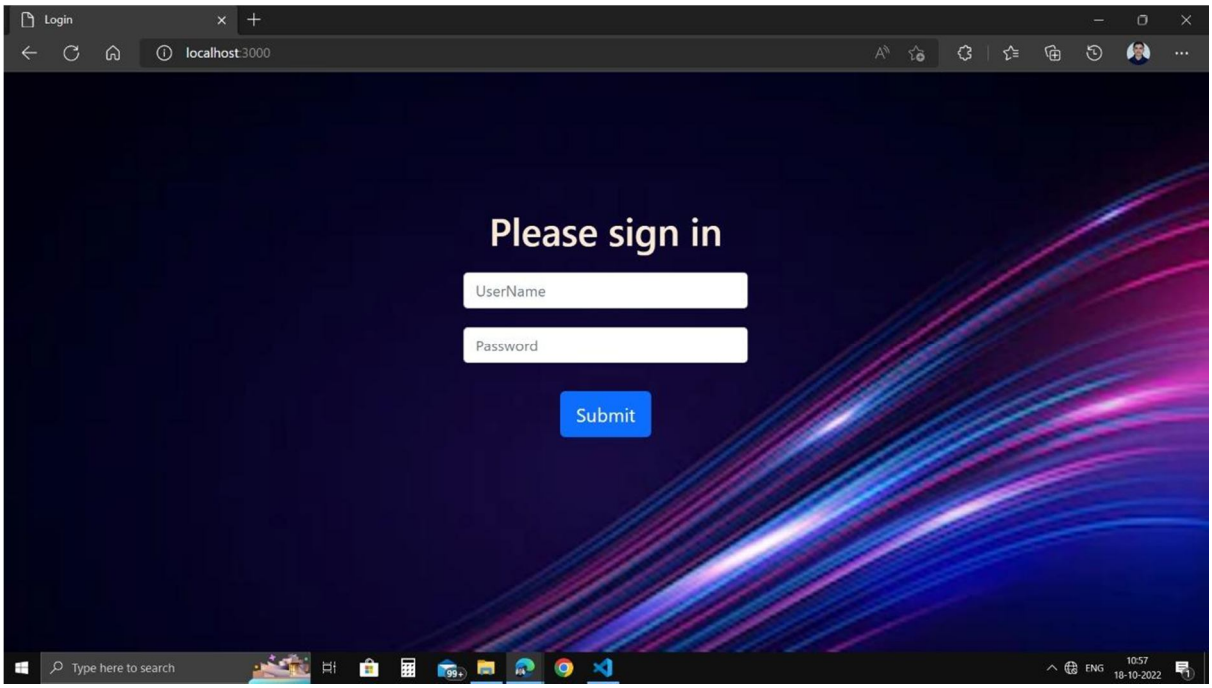
IV. RESULTS

The images of the developed model are shown below:

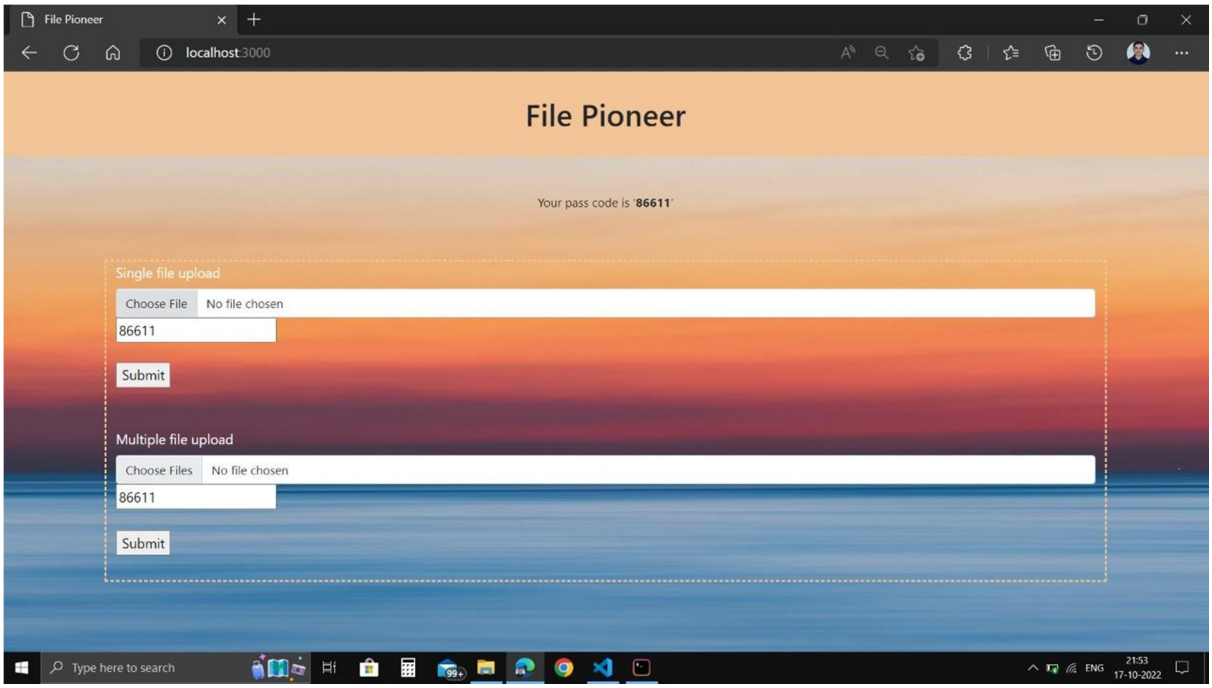
A. Screenshots

1) User Interface

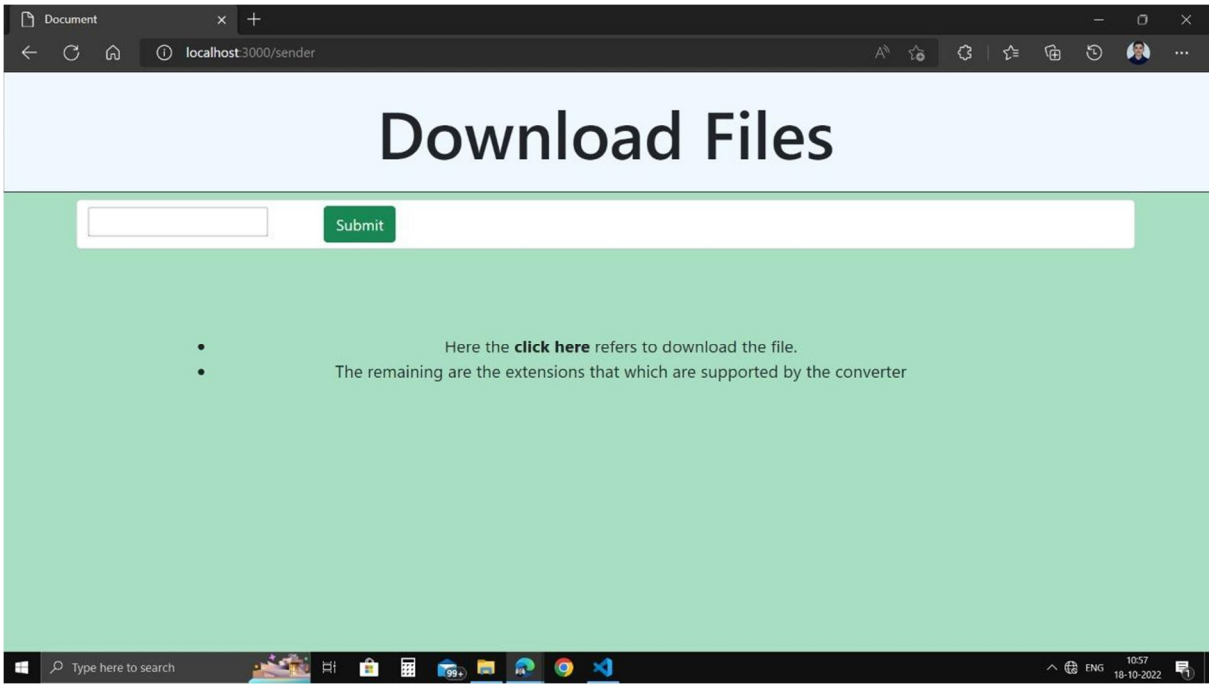
Here the first page is the login page, which specifies whether the user is the sender or the receiver.



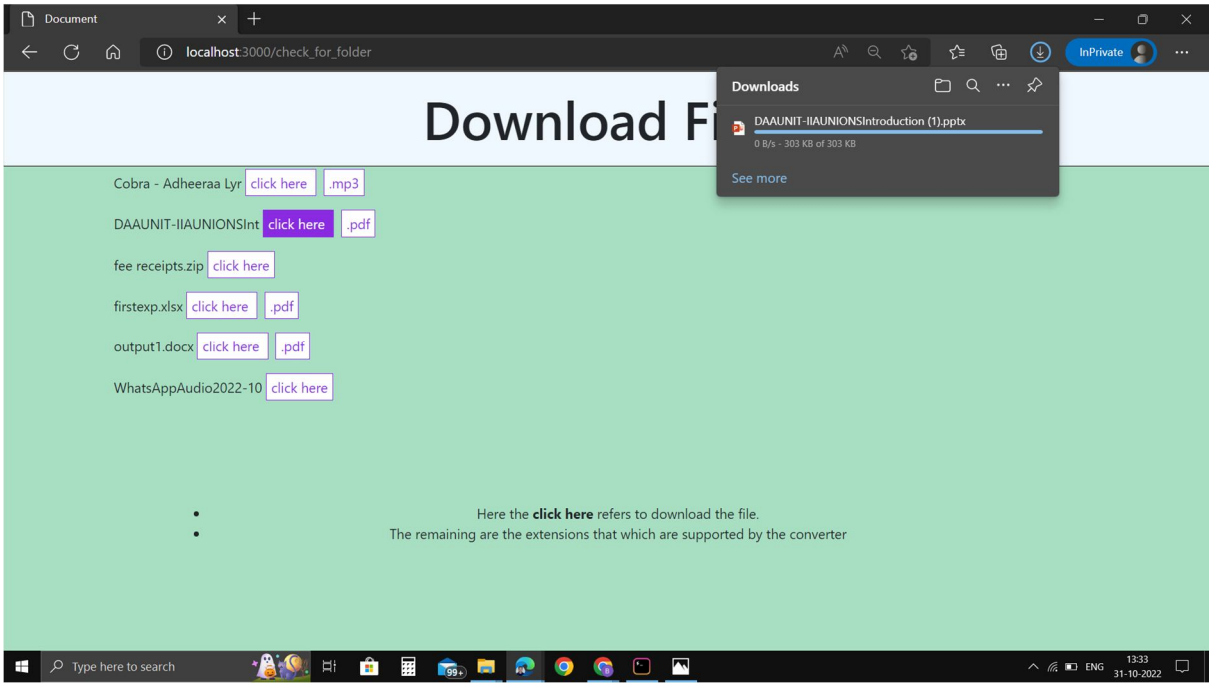
It is the sender page where the user sends the files, two fields specify a single file upload and a multiple file upload. and a secure key is also specified which is used to retrieve the files that are uploaded.



This page is for the receiver where the input field is filled with the secure key[2]. the files that are related to the secure key[2] get visible.



Clicking the "click here" button leads to downloading the files. Click on the file format button to download the file in that format.



V. CONCLUSION

This project focuses on transferring of File and formatting it into the desired conversion of the file, without bounding the range and consuming data, it requires minimum time compared to competitive applications. This project can transfer files of any format using passcode validation which provides users more security and in a comfortable manner

VI. FUTURE SCOPE

In the future, we may include an external server with a database that helps to do this process in a fast manner. Currently, it is running on the computer server which leads to an increase in the processor's tasks, at an external server we can give a high priority to a particular job. We can also include it in the satellites as well as in the robots to share the information in a parallel manner.

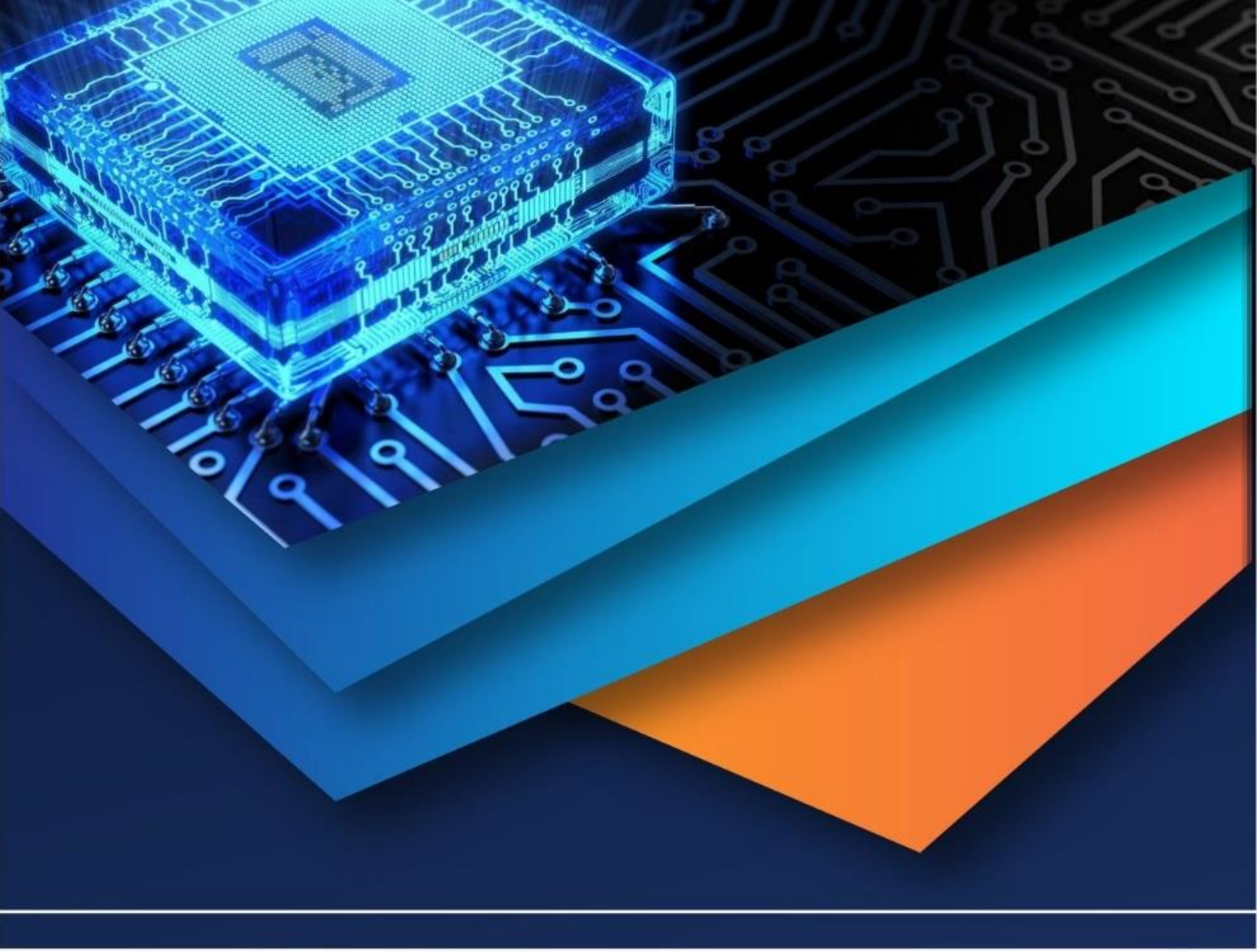
VII. ACKNOWLEDGMENT

We thank CMR Technical Campus for supporting this paper titled "**FILE PIONEER**", which provided good facilities and support to accomplish our work. Sincerely thank our Chairman, Director, Deans, Head Of the Department, Department Of Computer Science and Engineering, Guide and Teaching and Non- Teaching faculty members for giving valuable suggestions and guidance in every aspect of our work.

REFERENCES

[1] Firas Abdullah Thweny AI-Saedi and Zainab AI Taweel "Design and Implementation of File Sharing Server" International Journal of Emerging Trends & Technology in computer science 2015
[2] Seifedine Kadry "A New Security Paradigm of File Sharing" International Journal of Computer Science Issues 2010
[3] Aniello Castiglione, Luigi Catuogno, Aniello Del Sorbo,-Ugo Fiore, Francesco Palmieri "A secure file sharing service for distributed computing environments" The journal of supercomputing 2014
[4] Saurabh Malgaonkar, Sakshi Surve, Tejas Hirave "Distributed files sharing management: A file sharing application using distributed computing concepts" Computational Intelligence & Computing Research (ICCIC), 2012 IEEE International Conference.

© IJRASET: All Rights are Reserved |



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)