iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 9 **Issue:** VI **Month of publication:** June 2021

DOI: https://doi.org/10.22214/ijraset.2021.34950

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: 7.429
Volume 9 Issue VI Jun 2021- Available at www.ijraset.com

Cross-Platform File Explore with NW.JS

K. Tejaswini[1], L. Sireesha[2], B. Sirisha[3], K. Karthik[4], D. Lokesh Sai Kumar[5]
[1, 2, 3, 4]*UG Students,* [5]*Assistant Professor, Department of C.S.E, Prasad V. Potluri Siddhartha Institute of Technology, A.P. India*

Abstract: ***Our typical desktop file explorer applications are written in high-level programming languages and frameworks. Those apps are depending on the operating system. It can only operate on one operating system at a time. Windows&, for example, has My Computer, Windows 10 has This PC, Ubuntu has Files, and so on. Any web browser will not be able to operate this old file explorer desktop programme.***
This file explorer is a desktop application written in NW.js that runs on any platform. The application is created using Web-based programming languages. HTML, CSS, and JavaScript are examples of basic and straightforward languages. This programme is OS agnostic.
As a result, it may operate on a variety of operating systems, including Windows, Linux, and MacOS. It cuts down on time spent on development and training. The cross-platform desktop programme has basic features such as file opening, copying, and pasting, directory navigation, showing the current directory path, and windowing functions such as maximise, minimise, and dismiss the window.

I. INTRODUCTION

High-level programming languages and frameworks are typically used to create desktop apps. It can only operate on one operating system at a time. This explorer is a cross-platform desktop application built using NW.js, a framework for building apps with basic and straightforward languages like HTML, CSS, and JavaScript. This programme is OS agnostic. As a result, it may operate on a variety of operating systems, including Windows, Linux, and MacOS [1][2]. It cuts down on time spent on development and training. Our programmes provide features such as file opening, copying, and pasting, directory navigation; showing the current directory path and windowing functions such as maximise, minimise, and close [3][4][5].
The NW.js framework was used to create File Explorer, a cross-platform desktop application. This application's main goal is to deliver native applications [6]. The key benefit of this programme is that it is operating system agnostic, meaning it can operate on other operating systems [7][8].

II. PROBLEM STATEMENT

File Explorer allows directing through the file system and executes elementary actions on the files

III. EXISTING SYSTEM

The current standing desktop solicitations are conventionally referred as established using high level programming languages and outlines. These solicitations are operating system dependent which means that it can only run on a single operating system [9]. These solicitations cannot run on a web browser [10].

IV. PROPOSED SYSTEM

The suggested system consists of a desktop programme called a file explorer. Because this programme is native, it's referred to be a cross-platform desktop application. The cross-platform desktop application is independent of the operating system. As a result, it may operate on a variety of operating systems, including Windows, Linux, MacOS, Fedora, and others. Files, folders, and windows are used to run the main functions of this application.

V. SCOPE OF THE SYSTEM

The scope of the project is to create a File Explorer allows navigating through the file system and performs basic operations on the files. This can run on other operating systems [11]. This file system can access only local disk (C:) directory. We can perform open, close, copy, paste, delete operations.

VI. DESIGN

Preliminary design involves running our programme in the Node.js utility and creating a static template for our application. Elaborate design is concerned with fine-tuning the architectural representation, which leads to detailed data structures and algorithmic software representations.

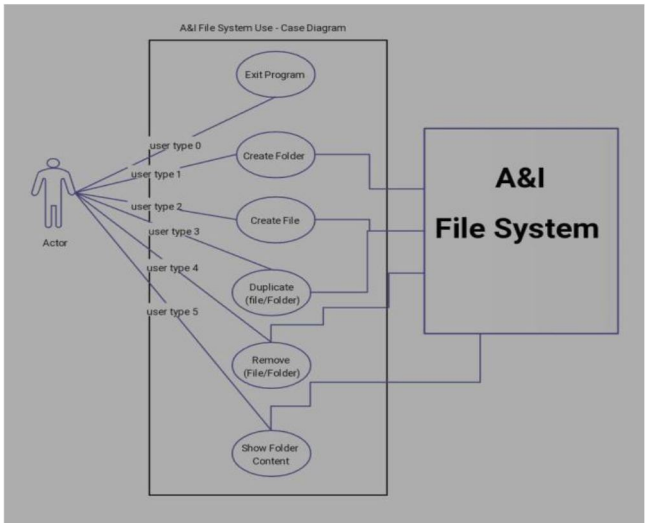
©IJRASET: All Rights are Reserved

A. Fundamental Design Concepts

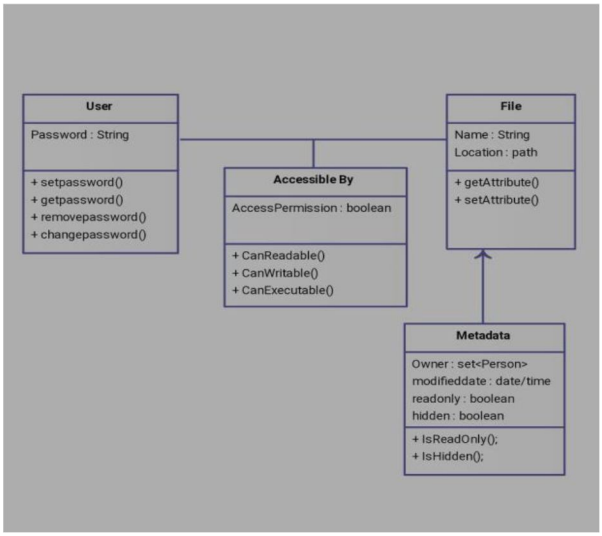
1) *Data Abstraction:* This is the process of describing important features without incorporating background information or explanations.
2) *Elaboration is the Process of Refinement:* A hierarchy is created by step-by-step dissecting a macro-statement of function until programming language statements are reached. Abstraction and refinement are two notions that work together.
3) *Modularity:* Software architecture is separated into modules, which are individual components.
4) *Software Architecture:* This term refers to the general structure of software as well as the methods in which that structure maintains a system's conceptual integrity.
5) *Control Hierarchy:* A programme structure that denotes a control hierarchy and describes the organisation of a programme component.
6) *Data Structure:* A data structure is a representation of the logical relationship between distinct data pieces.
7) *Software Procedure:* It focuses on each module's processing separately.
8) *Information Hiding:* Modules should be described and constructed in such a way that information contained inside them is unavailable to other modules that don't require it.

VII. UML DIAGRAMS

A. Use-Case Diagram

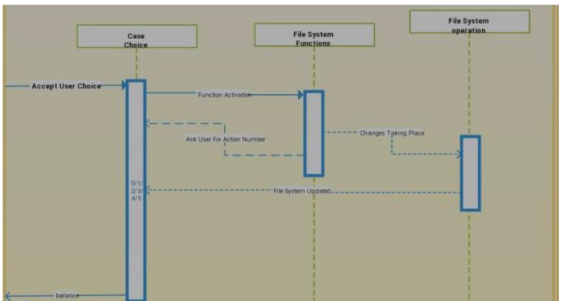


B. Class Diagram

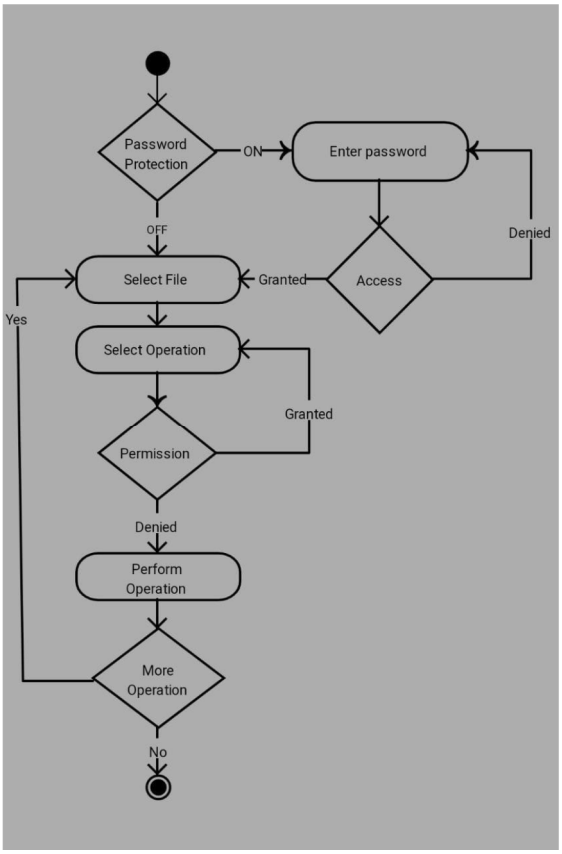


©IJRASET: All Rights are Reserved

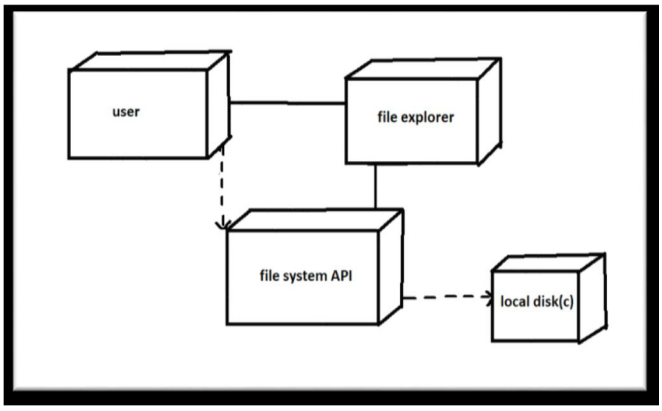
C. Sequence Diagram



D. Activity Diagram

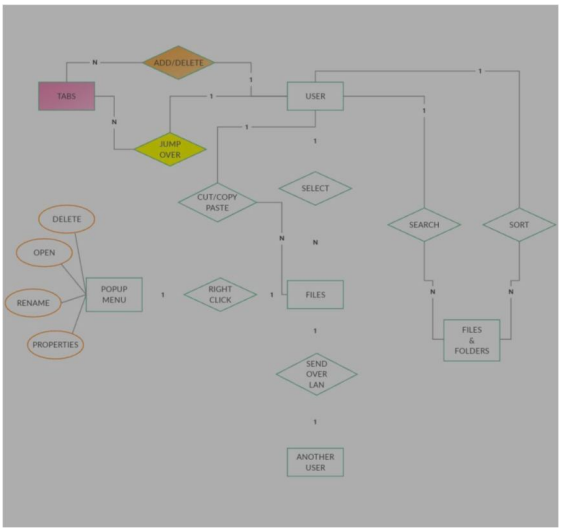


E. Deployment VIEW



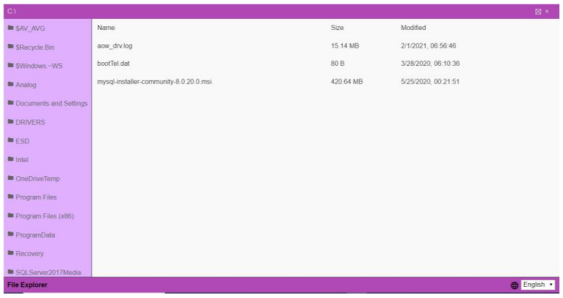
©IJRASET: All Rights are Reserved

F. Entity-Relationship Diagram

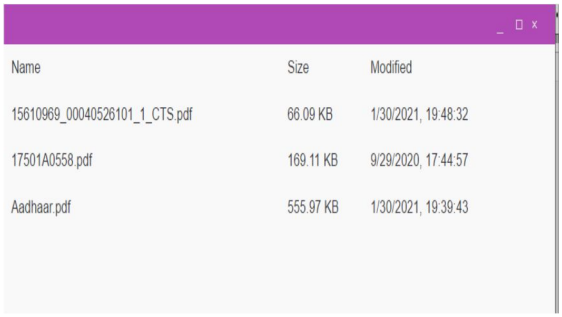


VIII. RESULTS

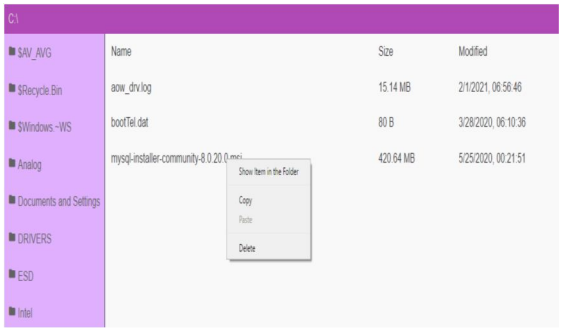
A. File Explorer Application



B. Window Events

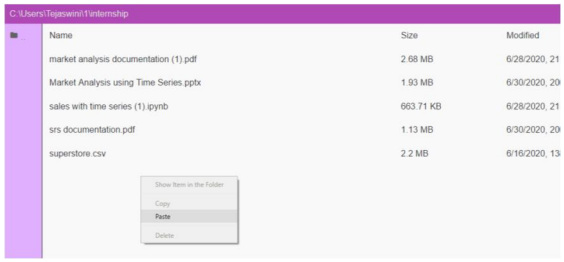


C. Context Menu



D. Copy Button

E. Paste button



IX. CONCLUSION

After a thorough examination of the component's positive aspects and limits, it is reasonable to conclude that the product is a highly efficient GUI-based component. This application is up and running and meets all of the user's needs. This component is simple to instal in any of the supported operating systems.

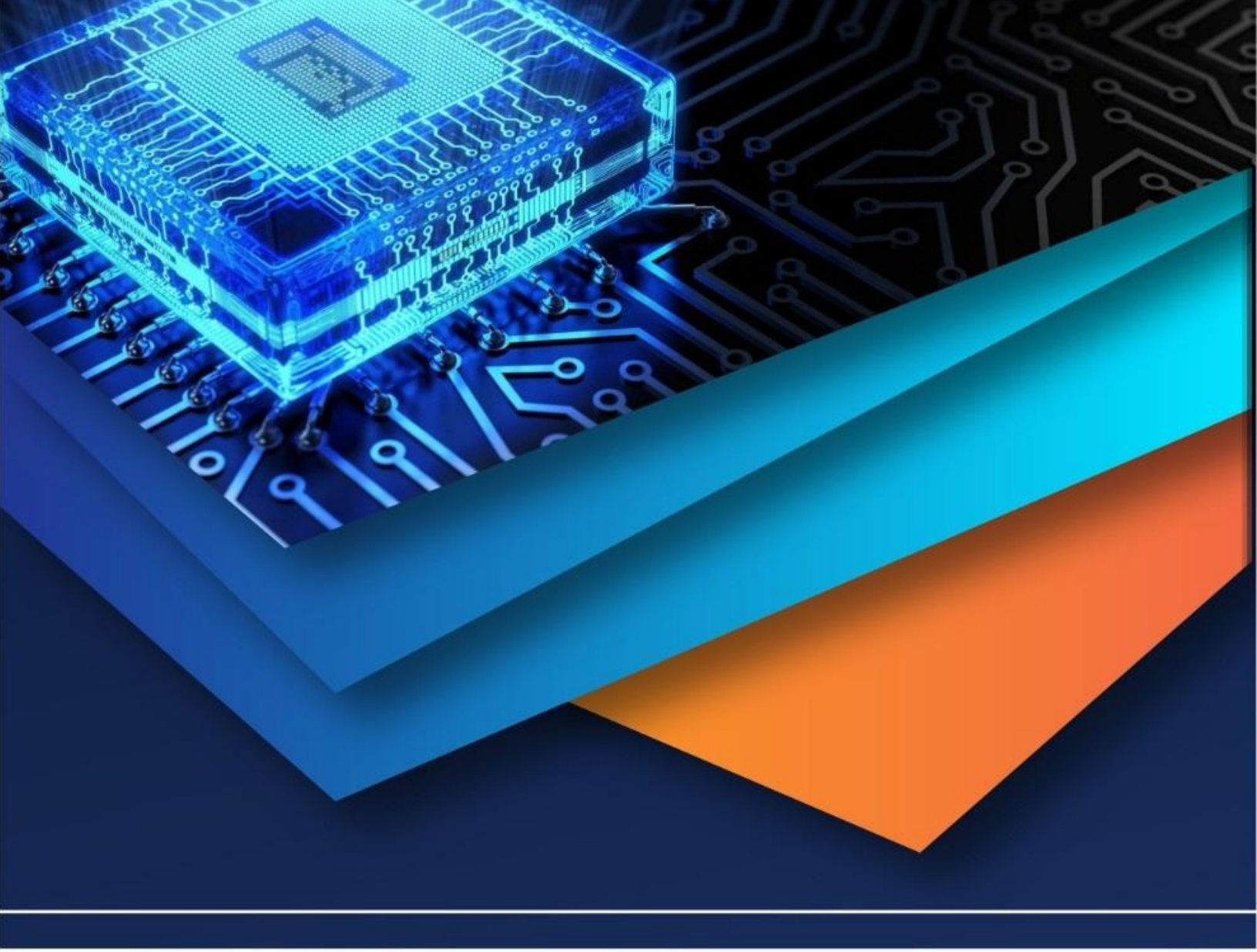
X. FUTURE SCOPE

This module may be improved by connecting the programme to the server so that the user may use the web browser to view the file explorer (such as Google chrome). So that users may access their file system from anywhere, and data from the operating system manager may be accessible quickly in the event of a system crash. We may add advanced file actions like creating a file, renaming it, printing it, seeing its properties, and so on.

REFERENCES

[1] V.Narmada, BN Swamy and DLS Kumar. An enhanced security algorithm for distributed databases in privacy preserving data bases in International Journal of Advanced Engineering Sciences and Technologies (IJAEST).
[2] Phani Praveen, S., and K. Thirupathi Rao. "Client-Awareness Resource Allotment and Job Scheduling in Heterogeneous Cloud by Using Social Group Optimization." International Journal of Natural Computing Research (IJNCR) 7.1 (2018): 15-31.
[3] Praveen, S. Phani, and K. Thirupathi Rao. "An Effective Multi-faceted Cost Model for Auto-scaling of Servers in Cloud." Smart Intelligent Computing and Applications. Springer, Singapore, 2019. 591-601.
[4] Praveen, S. Phani, and K. Thirupathi Rao. "An Optimized Rendering Solution for Ranking Heterogeneous VM Instances." Intelligent Engineering Informatics. Springer, Singapore, 2018. 159-167.
[5] Sindhura, S., Praveen, S. P., Syedbi, S., Pratap, V. K., & Krishna, T. B. M. (2021). An Effective Secure Storage of Data in Cloud Using ISSE Encryption Technique. Annals of the Romanian Society for Cell Biology, 5321-5329.
[6] Praveen, S. Phani, et al. "A new approach for optimizing resource provisioning in cloud computing using OCRP algorithm." Softw. Eng. Technol 5.12 (2013): 381.
[7] Swapna, D., and S. Phani Praveen. "An Exploration of Distributed Access Control Mechanism Using BlockChain." Smart Intelligent Computing and Applications. Springer, Singapore, 2020. 13-20.
[8] Praveen, S. Phani, and U. Tulasi. "A Study on Qos Challenges in Cloud Computing." IJCC 2.1 (2013).
[9] Praveen, S. Phani, U. Tulasi, and K. Ajay Krishna Teja. "A cost efficient resource provisioning approach using virtual machine placement." Int. J. Comput. Sci. Inf. Technol. 5.2 (2014): 2365-2368.
[10] Praveen, S. Phani, and T. Bala Murali Krishna. "An Overview of Emerging Technologies & Security Issues in Mobile Ad Hoc Networks." IJCC 1.1 (2012).
[11] Praveen, S. Phani, K. Thirupathi Rao, and B. Janakiramaiah. "Effective allocation of resources and task scheduling in cloud environment using social group optimization." Arabian Journal for Science and Engineering 43.8 (2018): 4265-4272.

©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)