Published by :
http://www.ijert.org

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 12 Issue 07, July-2023

FILE EXPORTER
Portage of file using Express JS

Madduri Sai Madhan Reddy[1], Koppula Bhanu Prasad Reddy[2] , Muthineni Anusha[3] ,G. Manisha[4], Reddyvari Venkateswara Reddy[5]
[1,2,3] B. Tech Student ,Department of CSE-Cybersecurity ,CMRCET Hyderabad ,Telangana
[4] Assistant Professor, Department of CSE- Cybersecurity, CMRCET, Hyderabad, Telangana
[5]Associate Professor, Department of CSE-Cybersecurity, CMRCET, Hyderabad, Telangana

Abstract:- Nowadays every field is being automated in every region. Automation plays a major role in Wireless Fidelity. Data Transmission also plays a vital role in the Automation of Files. Transferring data from one device to another device throws them into a problematic situation where we require HDMI Cables, Ports, Sockets, etc. This makes the transfer of data 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 transaction device is not in range, then the transfer remains paused. This bounds the range of transfer for data. And the huge amount of data required for data transactions makes theuser lose his data. Encryption is not inherently offered or enforced by every provider. The ability to resume a transfer is totally lost if the connection is lost. When it is important for a file to transfer from one device to another or another user then we should approach a source that which downloaded into the source i.e., a database and then later it reaches the user and then he will get downloaded and when there is no network in that locality i.e., internet then it would be a problematic situation. In that situation if we have a solution that won't consume any internet and also helps in sharing the files in any format and also help in maintaining the constant flow of upload and download or view in any format then it would savetime and as well as the internet, this will happen in the case of the File Pioneer, This project helps in transfer of the file in the network which use the Wi-Fi asthe bridge and that the servers web applications can be accessible in the format thus all the connected members can receive the file and connection is required. The server's application is made of express.js, and SCP (Secure Copy Protocol) which share the files from host to user or others, in this it is used in a similar way. In this, the transfer of the file is done server-based, and the server handles the files either to upload or download functionality in the network. It is like the server holds the file and server carried it out to all the systems or hosts that are connected within the network. It is secure according to the SCP protocol. Servers can be spoofed to send data to a random port on an unintended computer side. The data that is transferred is completely secure, At the implementation level more levels of encryption can be added so that all files can get more security as long as the server is working, At the initial level the process is having basic security, Thus the files can be downloaded safely.

Keywords— data,Wi-Fi enabled,SCP protocol.

1. INTRODUCTION

This project is titled "File Exporter" Traditional methods for exporting any or data from one user to another user or transversal of files for multiple users will be automated using JavaScript in the beyond surface area connected to a network in a safe way with the help of a secure copy protocol and using a key between the users could be shared securely in the region and also receive the file in the desired format.

This project has been developed to transfer files or documents without using orcharging any data. And receiving different files in the desired format and if was user wishes to receive what format to have. Itis fruitful for multi-usage and transferring from multiple users.It is a boon for automation techniques since end-user availability can be automated. A secure file transfer system is achieved by using a multilevel secure key that gets auto-generated.

The main feature of our project is to transfer files from one system to another system without using any data. We can also transfer our files to multiple users at a similar period who are connected to the same network. Our project provides a high level of security and gives priority to users' data using a secure key, and works under FCFS first come first service technique which helps end users. One of the best features of this project is formatting so, the user can receive documents in the desired features of this project is formatting so, the user can receive documents in the desired format.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 thisfield 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.

2. LITERATURE REVIEW

Bhuvaneswar Prasad et al. have researched on NodeJS which provides a platform for file transversal NodeJS is an open-source server environment that allows the running of JavaScript on the server side with Object Oriented Programming concepts.It has different important features.

Firas Abdullah Thweny 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 which is used to fetch the random key which 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 use only. The combination of the random key, validation key, and transaction id as proposed will provide more robustness to secure the transfer of files using Blowfish symmetric key encryption.

3. IMPLEMENTATION.

The proposed approach leverages a web server to streamline the file exporting process, eliminating the need for third-party plugins or additional software installations. This integration with the web server facilitates convenient and secure exports within a controlled environment.

Key features of this web server-based file exporter include support for various file formats such as PDF,CSV, XLSX, and more. It ensures the conversion and formatting of web-based content into these file formats while preserving data integrity and maintaining the original structure. Customizable export options allow users to specify preferences, including page orientation, file compression, and metadata inclusion.

Security is of utmost importance, and the web server-based file exporter implements robust measures to protect sensitive data during the export process. Encryption protocols secure files in transit, while authentication mechanisms ensure authorized access to the export functionality. Integration with existing user management systems enables seamless integration into web applications and facilitates proper access controls.

To ensure compatibility and easy integration, the solution adheres to industry standards and employs established web technologies such as HTTP, HTTPS, and RESTful APIs. This approach allows for seamless integration with a wide range of web servers and existing web application architectures.

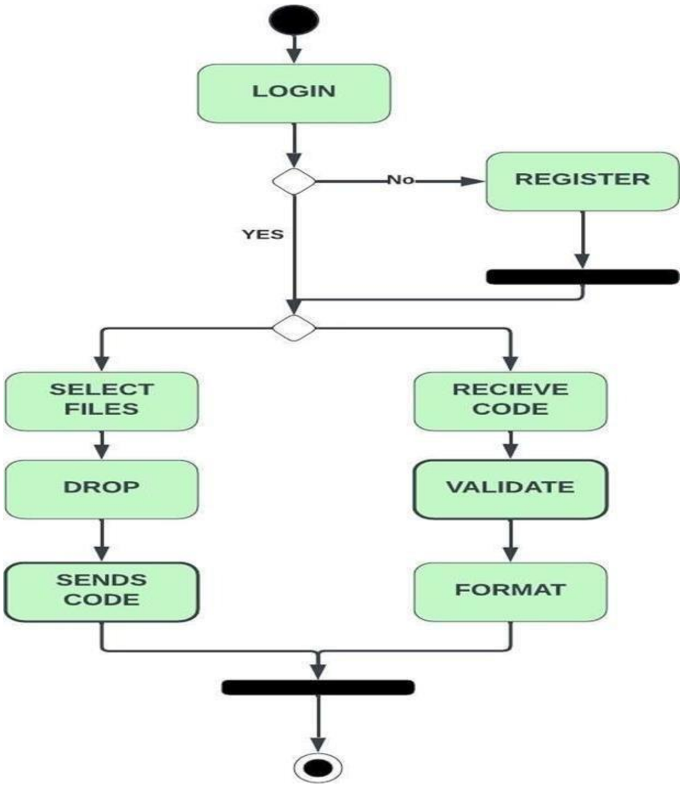

Figure: Steps of Implementation.

3.1 Windows Form Application

ADO.NET: stands for (ActiveX Data Object), is special set of .net framework allows the user to work with different type of databases such as Access, SQL server and Oracle. It provides two ways to work with data in a database: connected mode and disconnected mode.
In order to convert Excel file to XML file, we will use the ADO.NET in disconnected mode.
The following Actions:

1. Creates a connection to the Excel file using OLEDBConnection object.
2. Opens a connection with the property settings specified.
3. Selecting the data from the Excel file using DataAdapter object.
4. TableMapping.
5. Fills this DataTable with the imported data.
6. Set the property “Table” with the first DataTable contained in the DataSet “DtSet”.
7. Exports this Dataset to an XML file.
8. Reads the Data Columns and adds them the Array List “collist”.
9. Closes the connection to the data source.

3.2 Encrypt Node(s) in XML file:

Since we need to encrypt only a portion of the file, XML Encryption is the best choice if the application requires a combination of secure and insecure communication (which means that some of the data will be securely exchanged and the rest will be exchanged as is the Select Nodes To Encrypt() method will perform the following actions:

1. Gets the “EmployeeID” of the nodes to be encrypted.

2. Loops through the selected items and creates an EncryptedXML object “exml”.

3. Selects the XML element(s) needed to be encrypted.

4. Encrypts the element(s) using the key generated “sharedkey” by the encryption algorithm object.

5. Creates an encrypted data object “ed” and specifies its properties. 6. Creates a cipherData element and sets its value to the encrypted XML element “EncryptedElem”.

7. Replaces the plaintext XML element “Elem” with the encrypted data object “ed”. 8. Saves the encrypted data to a file "EncryptedData.xml"

3.3 Encrypt Entire Column in XML file:

The SelectColumnToEncrypt() method will perfom the following actions: This method is similar to the SelectNodesToEncrypt() method but instead, it takes the entire column to be encrypted. Therefore, an “XMLNodeList” object will be created to hold all the elements inside the plaintext XML file: To select an ordered list “xmllist” of XML nodes, we should create an object of XmlNodeList() and use the function SelectNodes()which takes an argument as X-Path expression. Referring to [9], X-path is a syntax for defining parts of an XML document. Convert Encrypted XML file to Excel: The ConvertXMLtoExcel()will perform the following actions: 1. Creates a new Excel Application, workbook “exbook” containing one Datasheet “exsheet”.

2. Creates a new dynamic DataSet which has a DataTable that itself contains DataRows and DataColumns.

3. Loads the encrypted XML file specified by the user.

4. Creates DataColumns, sets their Data types and adds them to the DataTable.

5. Creates XMLNodeList object “empList” that contains all “TableMapping” nodes specified by the user.

6. The DataRows of the XMLElements in “empList” are filled with data available in “EncryptedData.xml” file

7. Sets the value of the property “DataTableENC” to the first table of the dataSet “ds”.

8. Fills the Datasheet’s cells “exsheet” with the DataTable’s headers and rows.

9. Saves the new workbook Excel file and closes it, and quits the Excel Application.

3.4 Decrypt the XML file:

The DecryptXML()method will perform the following actions: 1. Loads the encrypted XML document specified by the user.

2. Creates an XMLNodeList ’regList’ holding all encrypted nodes in the encrypted XML document

3. Loops through this list and searches inside all “” tags if they contain the current description of the logged on user.

4. When this description is found , it retrieves the encrypted XML Element(s), Creates an encrypted data object ”ed2” and loads the encrypted element “c.firstChild” into the encrypted data object.

5. Creates an encrypted XML object “exml2”.

6. Decrypts the encrypted element”ed2” using the shared key

7. Replaces the encrypted element “c.firstChild” with the plain-text XML element “decryptedElem”.

8. Saves the decrypted data to an external file (optional).

4. REQUIRED COMPONENT

The following are the required components:
Node JS: Node JS 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.

1) Express JS: It is a Java script framework that provides the server functionalities for running theapplication. Express JS provides the routing feature. and It acts as middleware. Nodejs uses Express JS to adapt the functionality of the server.

2) EJS(Embedded JavaScript): EJS provides the java script functionality using tags. It usesthe EJS extension but also we can use the HTML inside of the page and when we want a function we candirectly dump the java script program into the fileand also it is fastly executable.

3) NPM(Node Package Manager): NPM is a package manager, and it is completely open

source, in which we can download the packageswith 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.

4) Libre office-convert-win: It is an NPMpackage that uses the installed LibreOfficesoftware to format the office formats to the pdf format, Nodejs have access and this NPM package connects it with the Node JSfunctionality to getthe desired output

5) 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 usedto format the mp4 or video to audio.

6) JIMP: JIMP is a Node JS 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

4.1 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.

4.2 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 thelocal 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

5. CONCLUSION

In conclusion, the web server-based file exporter provides a comprehensive solution for seamless and secure file exporting within web applications, addressing the increasing demand for efficient data exchange and collaboration. Its integration with the web server enables users to export files directly from web-based platforms, facilitating streamlined workflows and enhancing user experience

6. FUTURE SCOPE

In the future, we may include an external server with a database that helps us to do this process quickly. 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.

REFERENCES

[1] Firas Abdullah Thweny AI-Saedi and Zainab AI Taweel “Design and Implementation of File Sharing Server” International Journal of Emerging TrendsTechnology in computer science 2015

[2] Seifedine Kadry “A New Security Paradigm of File Sharing” InternationalJournal of Computer Science Issues 2010

[3] Bhuveneshwar Prasad Sharma “NodeJS Quick Learning” International Journal of Computer ScienceIssues 2020

[4] 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

[5] [1] 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.