**FILE LIBRARY WITH GROUP SECRET KEY GENERATION**

**B.Nithish** **Ms. R. Sri Lakshmi**

M.Sc CFIS, Center of Excellence in Digital Forensics,

Department of Computer Science and Engineering, Perungudi, Chennai-600089

Dr. M.G.R Educational and Research Institute,

Maduravoyal, Chennai-600 095.

**Ms. Sarika Jain Dr. S. Geetha**

Center of Excellence in Digital Forensics, Head of the Department

Perungudi, Chennai-600089 Department of Computer Science and engineering,

Dr. M.G.R Educational and Research Institute,

Maduravoyal, Chennai-600 095.

*Abstract*— In this application, we are developing cryptography systems that include applying a technique called a computation to plain text to transform it into something that will appear to anyone without the ability to decode it as jibberish. Here, the user must register their details and log in before they can read the list of archives saved in this application; the content of the record won't be visible to the user because it will be encoded when the record is moved there. After the entry of the overseer, the client can move their interested records and download them. The key and the archive are both completely converted to incomprehensible code during the transfer. The peculiarities of the uploader and downloader are kept in mind by the management. The client can then use the decrypted key to obtain the archive of the person who interested them at that time.

# **I.Introduction**

The user must select their login information in order to view the document's outline once it has been stored in this application; however, the client cannot view the document's content because it is encoded when it is moved into this programme. The customer can download with the part of administrator and relocate their enchanted record. The key and the record are both completely converted to non-clear code during the transfer. The leader must constantly be on the lookout for downloader and uploader nuances. The client can then use the decrypted key to download the report of the intriguer.

# **II.Literature survey**

**Paper1**

# **TITLE 1:Asynchronous Distribution and Coded Caching for Time-Varying File Popularities**

**AUTHOR:** [Mohamed Amir](https://ieeexplore.ieee.org/author/38275847400).

**ABSTRACT:**

Data communication has recently experienced exponential growth and currently dominates wireless communication; therefore, proactive caching was developed to reduce peak traffic rates by storing content, in advance, at different nodes in the network. We consider proactive caching for a broadcast wireless network with one central hub such as a satellite (ST) and K associated mobile units (MUs) such as mobile mini-ground stations or end users..We outline a fresh plan that reduces the sum rate of files delivered, and we demonstrate how the file delivery messages can be used to proactively and continuously update the MU finite caches. We demonstrate how this approach lowers the network's downloaded traffic. The suggested method makes use of index coding to jointly encode the delivery of various requested files along with cache updates to other MUs to track changes in the popularity of the files. It is suggested to perform an offline optimisation of the delivery sum rate of the scheme, which necessitates knowledge of the popularity of the files over the entire transmission duration.

**Paper2**

# **TITLE 2:** A Secure Communication System Using Lightweight Group Key Generation in Self-Organizing Networks

**AUTHOR:** [Sirui Peng](https://ieeexplore.ieee.org/author/37087407198).

# **ABSTRACT:**

# Self-organizing networks lack a central node, hence conventional key distribution techniques based on open infrastructure are ineffective. The security vulnerabilities associated with standardised pre-shared keys are foreseeable. Due to its portability, security, and decentralisation, physical-layer secret key generation has emerged as a method to take into consideration. The expansion of the pairwise key into the group key is difficult because the majority of the prior research has concentrated on two devices. Information leakage would result from some data being sent on the unencrypted channel because channel reciprocity only applies to two devices. In self-organizing networks, a secure communication system is designed in this paper. To produce the paired keys, it uses an adaptive quantizer and suggests DORCE, or Difference Of quantization Results at one device.The distinction between pairwise keys is how the authenticated users share the group key. Because users' joining and departing will not have a significant impact on the network topology, the technique is implemented in a mesh topology,

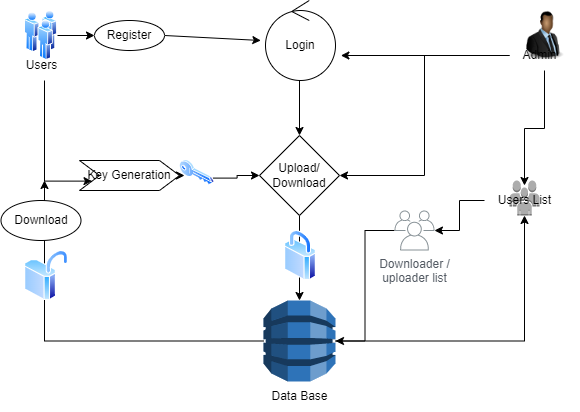
# **Existing system**

A two-layer, scalable blockchain platform for distributed multi-cloud storage in the IIoT. The approach is extremely scalable because it is built around an asynchronous consensus group.

# **proposing system**

The customer can get ready to move their nuances and download from it with the aid of an uncrashed key after being approved by the overseer by using this programme.

# **ARCHITECTURE DIAGRAM**



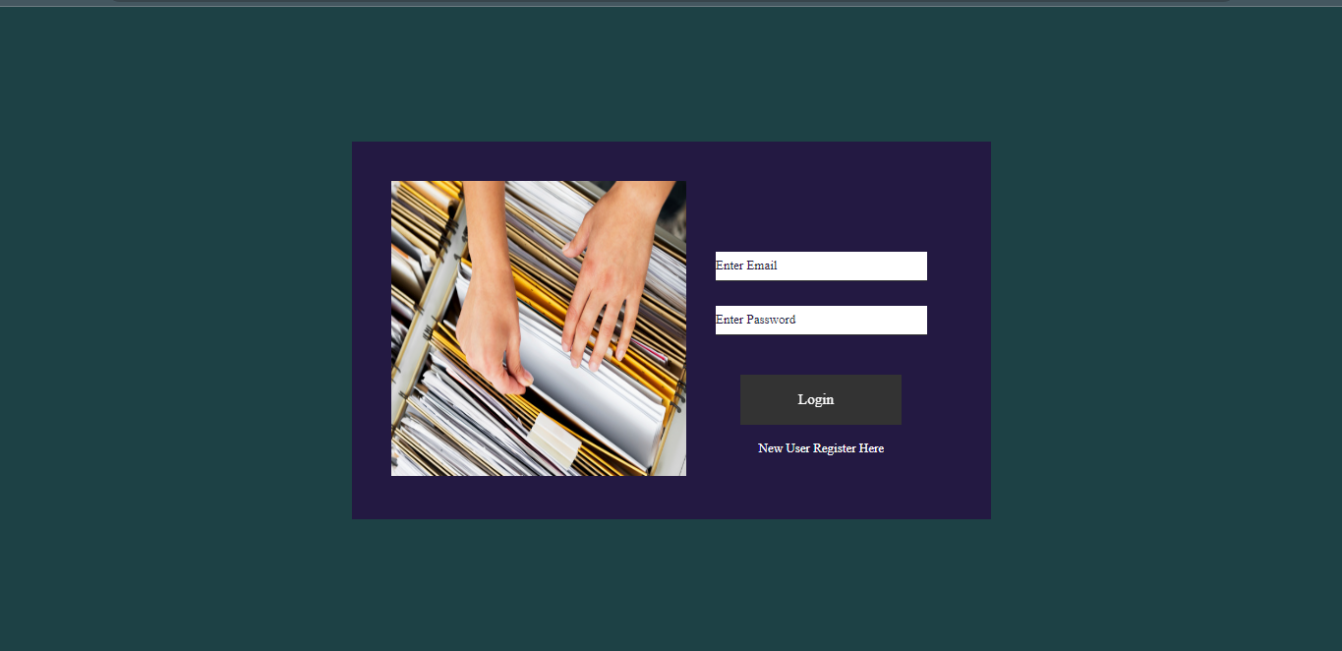
**ER DIAGRAM**

# **C:\Users\SPIRO24\Downloads\Untitled Diagram-Page-7.drawio.png**

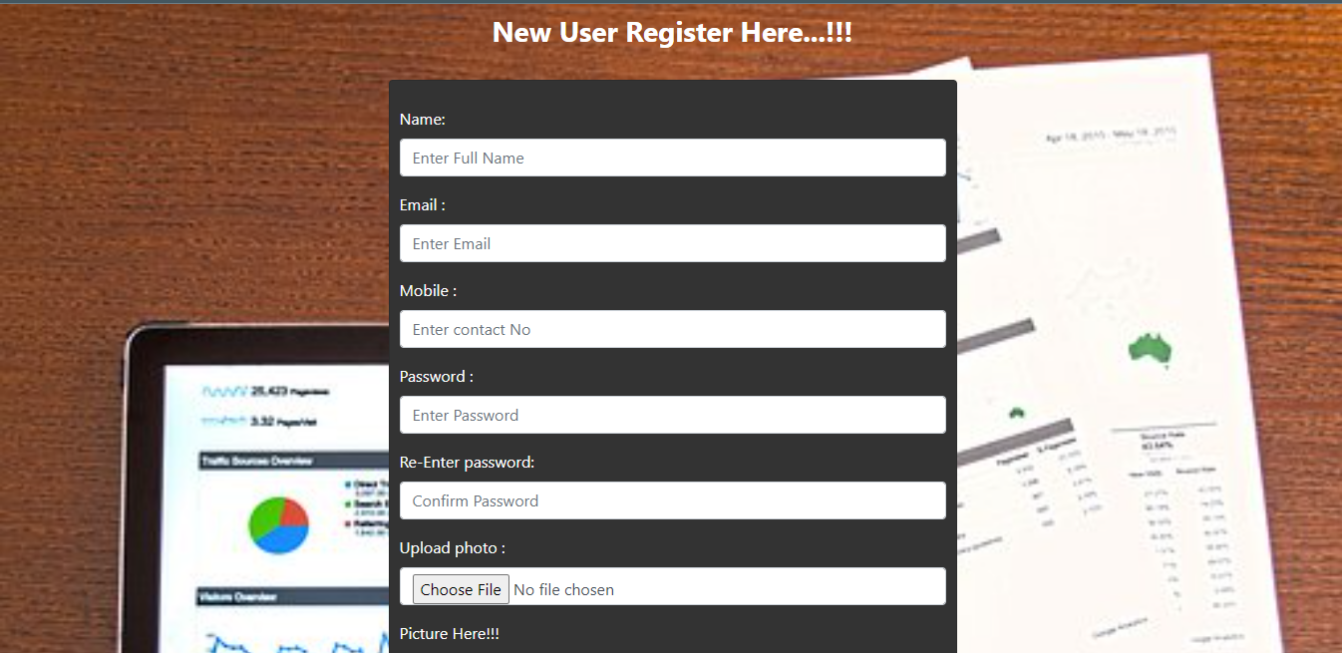
# **SCREENSHOTS**



***1. HOMPAGE***



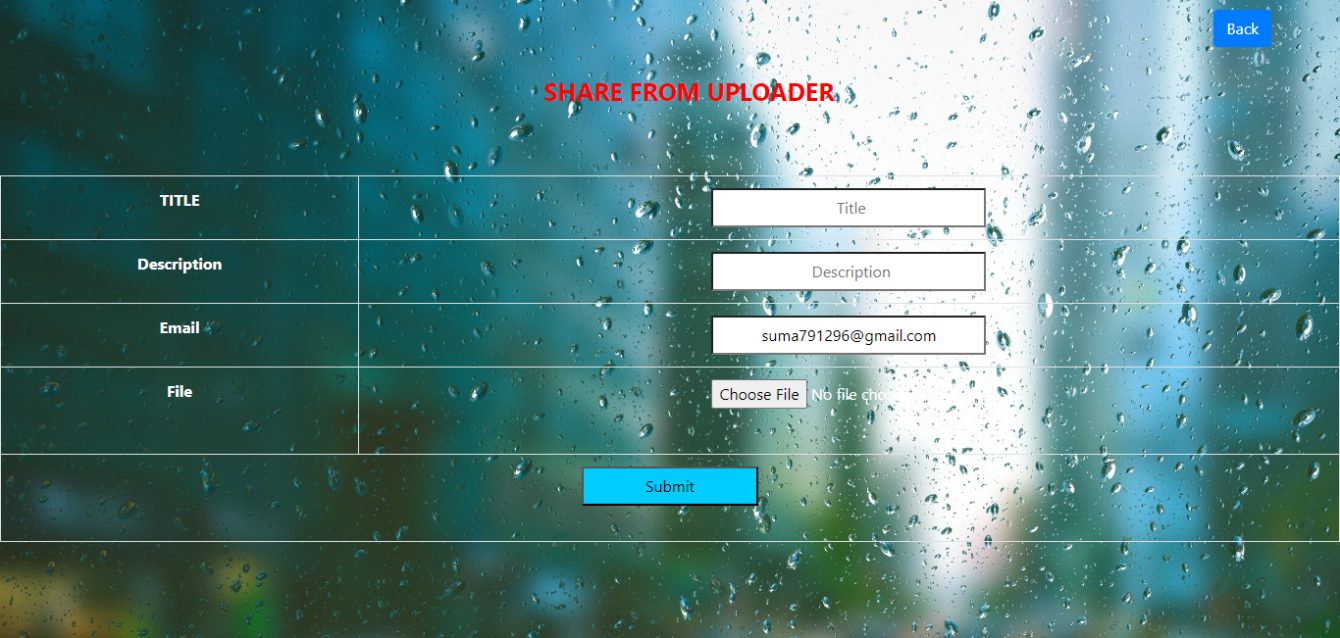
1. ***USER LOGIN PAGE***



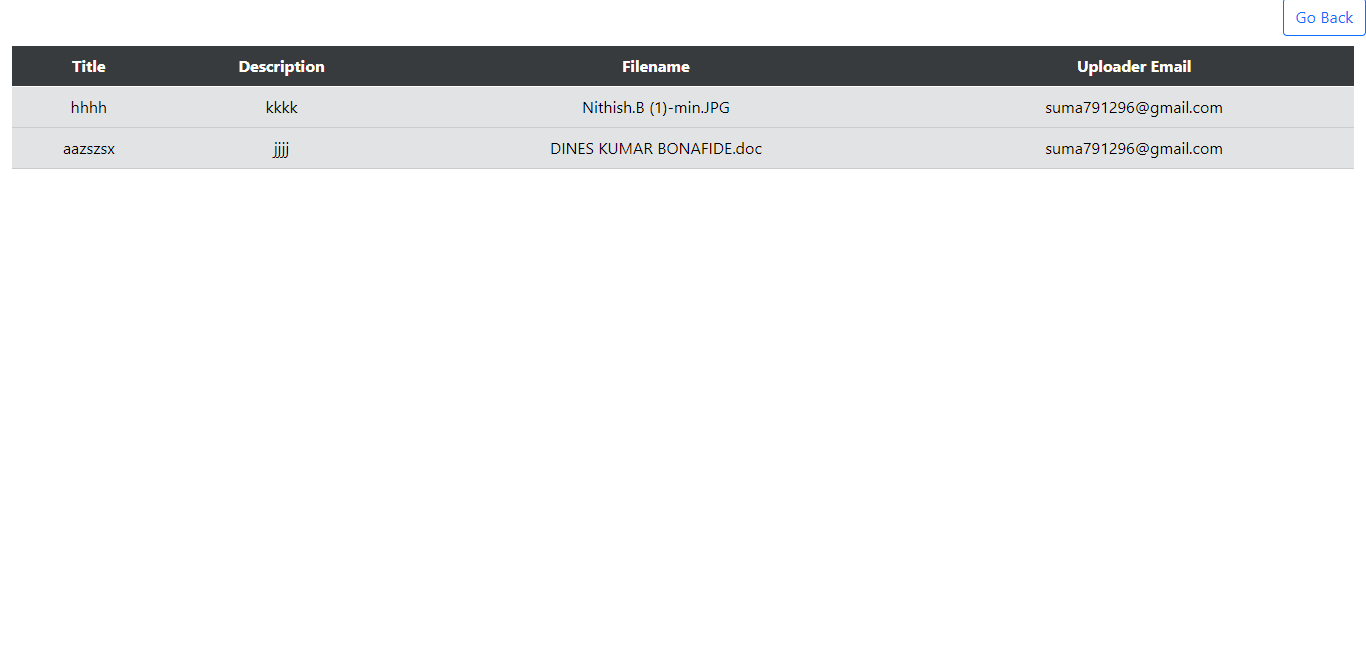
1. ***NEW USER REGISTER PAGE***



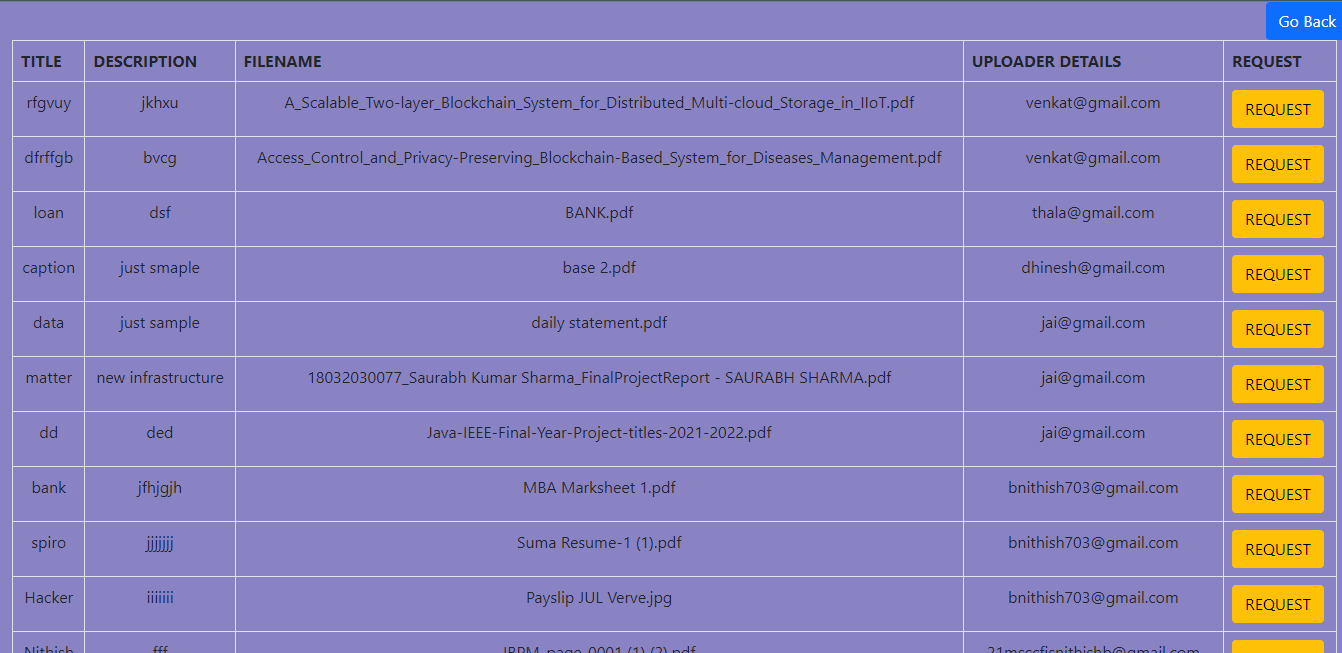
1. ***USER HOME PAGE***



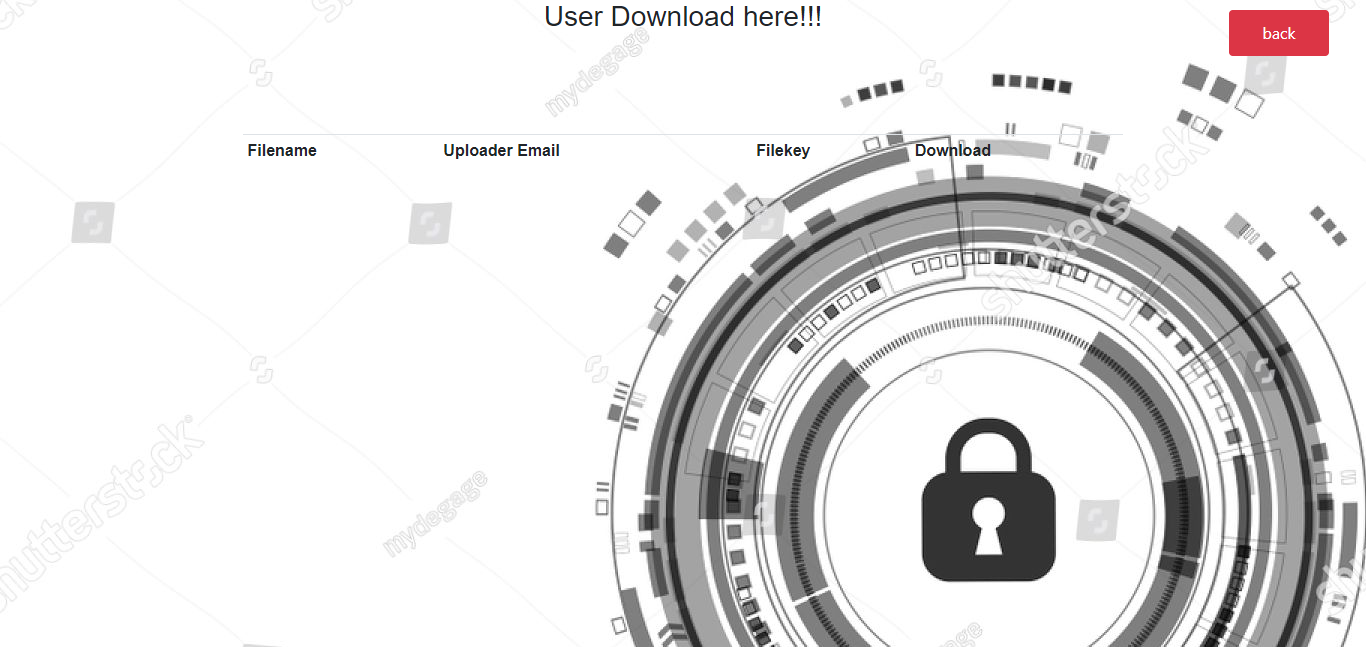
1. ***FILE UPLOAD PAGE***



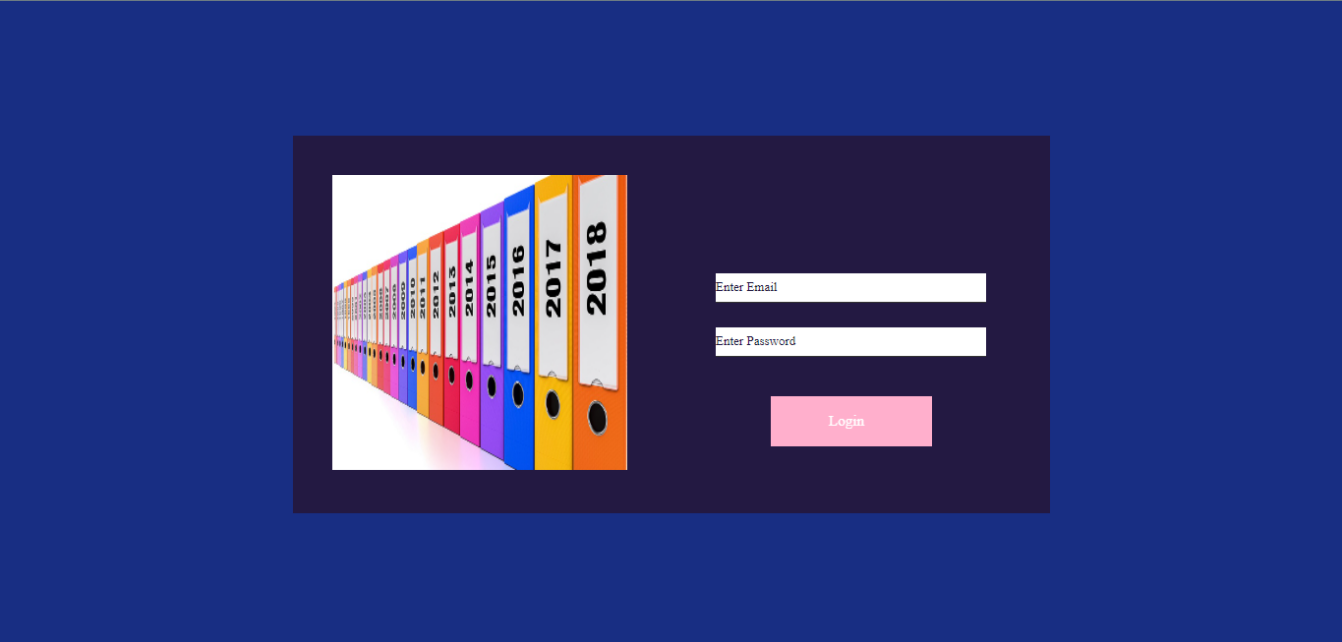
1. ***UPLOADED FILE VIEW PAGE***



1. ***FILE REQUEST PAGE***



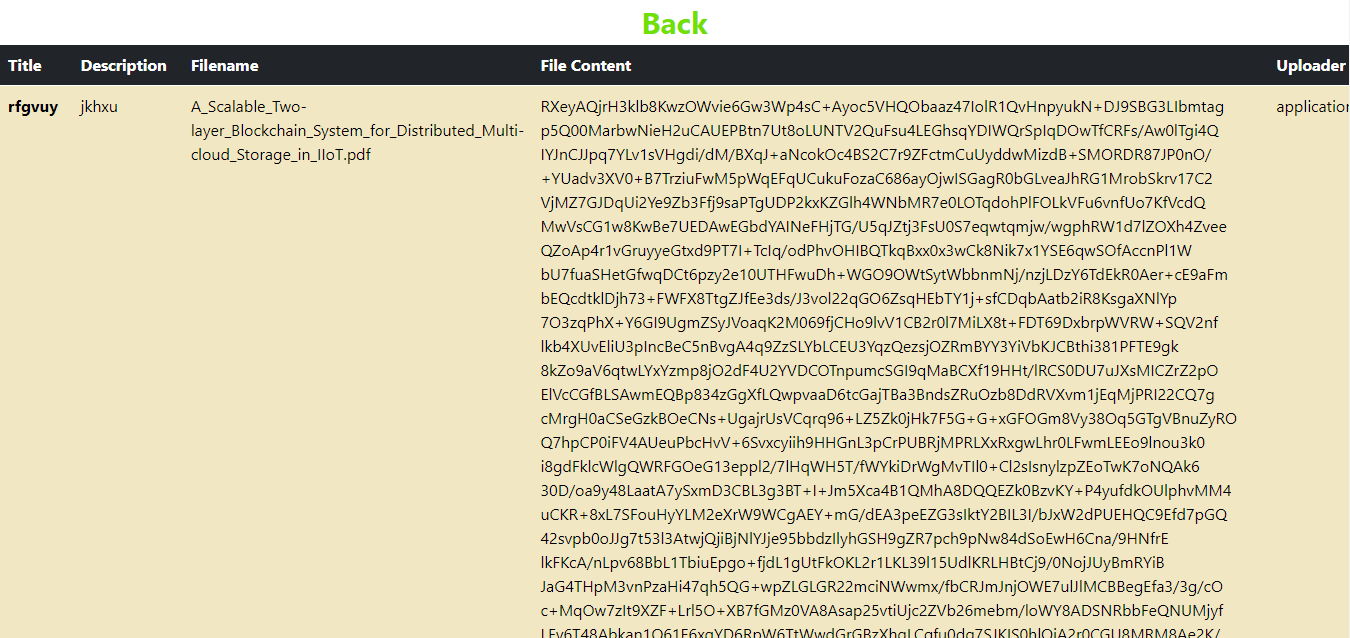
1. ***FILE DOWNLOAD PAGE***



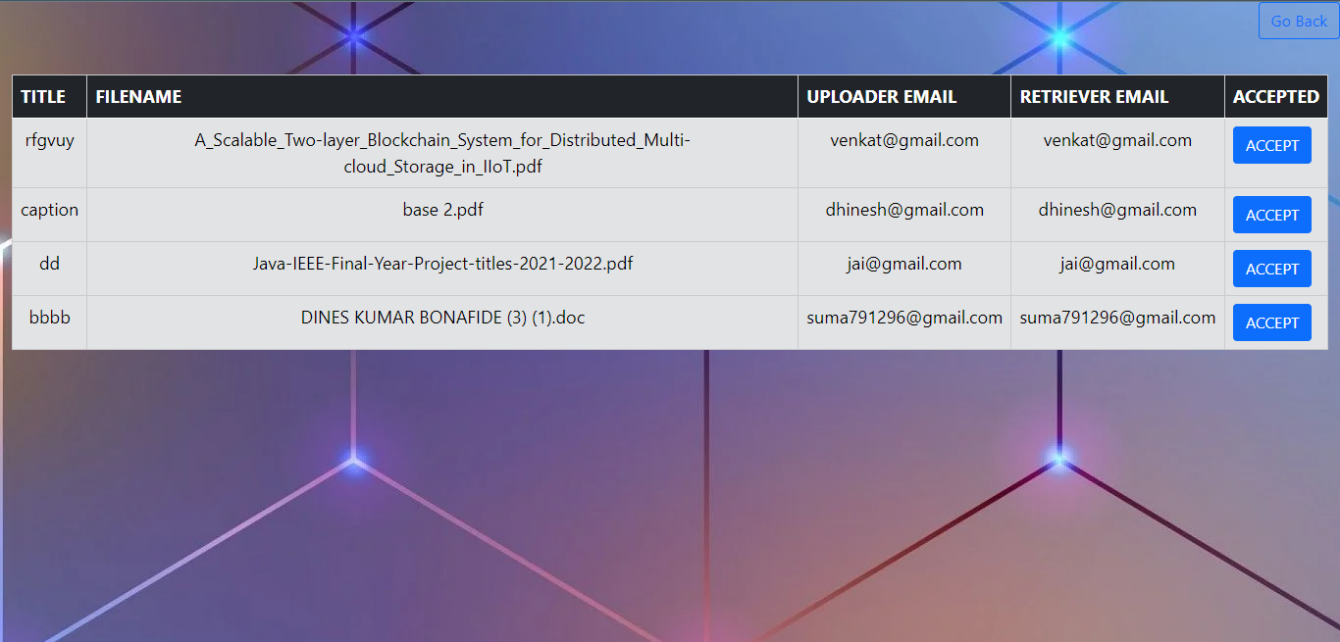
***9. MANAGEMENT LOGIN PAGE***



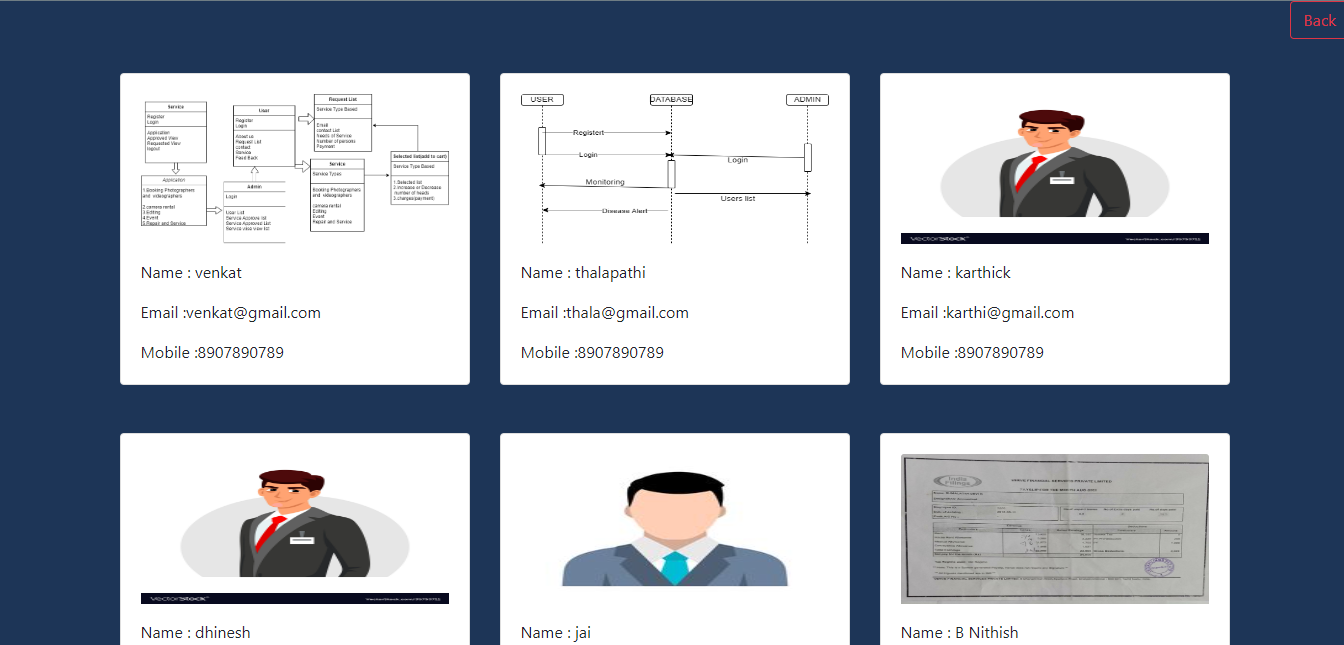
***10. MANAGEMENT HOME PAGE***



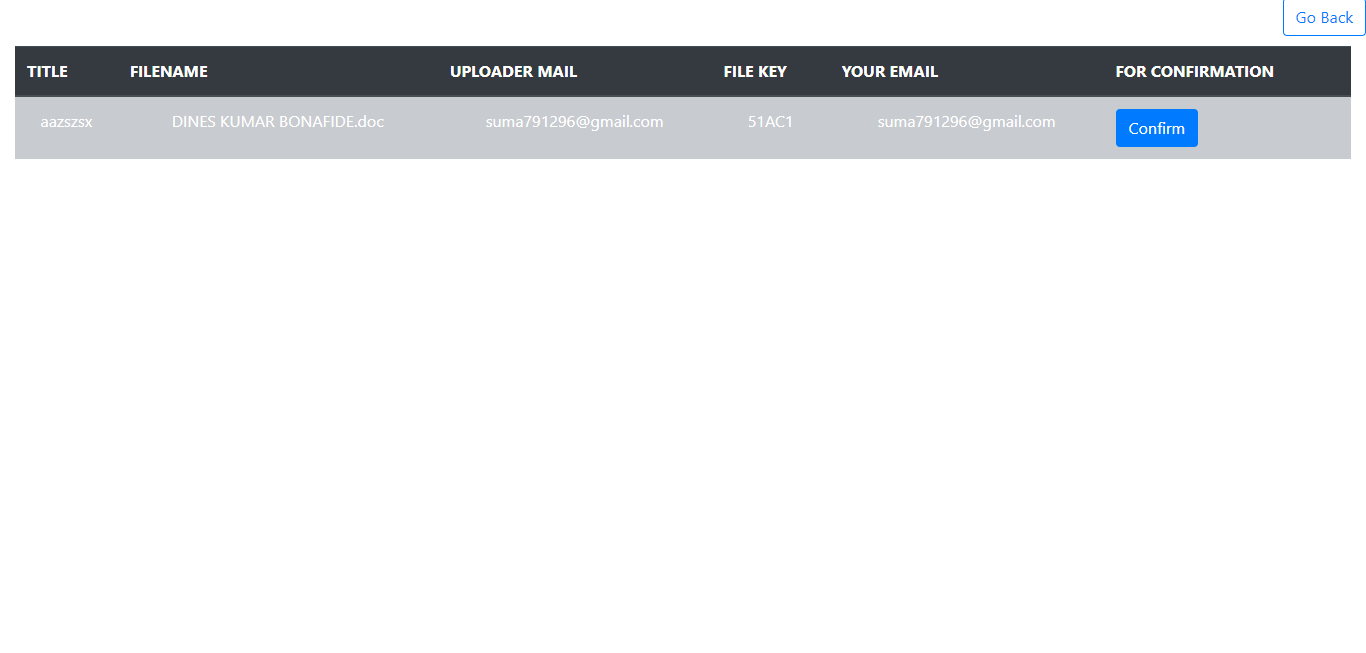
***11. FILE LIST VIEW PAGE***



***12. USER REQUEST PAGE***



1. ***USER LIST VIEW PAGE***



1. ***CONFORMATION PAGE***

# **Conclusion:**

The administrator maintains awareness of the downloader's and transferred's subtleties. With the help of the deciphered key, the client can download the list of those who dazzled. The client can download and relocate their intriguing records with the help of the administrator. The report and key are both completely converted to illogical code during the move.

**REFERENCES:**

**1.** Z. J. Czech, G. Havas and Β. S. Majewski, "Perfect Hashing", *Theoretical Computer Science*, no. 182, pp. 1-143, 1997.

[CrossRef](https://doi.org/10.1016/S0304-3975(96)00146-6" \t "_blank)[Google Scholar](https://scholar.google.com/scholar?as_q=Perfect+Hashing&as_occt=title&hl=en&as_sdt=0%2C31" \t "_blank)

**2.** Z. J. Czech, G. Havas and B. S. Majewski, "An optimal algorithm for generating minimal perfect hash functions", *Information Processing Letters*, vol. 43, no. 5, pp. 257-264, October 1992.

[CrossRef](https://doi.org/10.1016/0020-0190(92)90220-P" \t "_blank)[Google Scholar](https://scholar.google.com/scholar?as_q=An+optimal+algorithm+for+generating+minimal+perfect+hash+functions&as_occt=title&hl=en&as_sdt=0%2C31" \t "_blank)

**3.** R. Gupta, S. Bhaskar and S. Smolka, "On Randomization in Sequential and Distributed Algorithms", *ACM comput. Surveys*, vol. 26, no. 1, pp. 7-86, March 1994.

[CrossRef](https://doi.org/10.1145/174666.174667" \t "_blank)[Google Scholar](https://scholar.google.com/scholar?as_q=On+Randomization+in+Sequential+and+Distributed+Algorithms&as_occt=title&hl=en&as_sdt=0%2C31" \t "_blank)

**4.** J. Ebert, "A versatile data structure for edge-oriented graph algorithms", *Communications of the ACM*, vol. 30, no. 6, pp. 513-519, June 1987.

**5.** C. Cifuentes and K. J. Gough, "Decompilation of binary programs", *Software – Practice and Experience*, vol. 25, no. 7, pp. 811-829, 1995.

[CrossRef](https://doi.org/10.1002/spe.4380250706" \t "_blank)[Google Scholar](https://scholar.google.com/scholar?as_q=Decompilation+of+binary+programs&as_occt=title&hl=en&as_sdt=0%2C31" \t "_blank)