



**L** LOVELY  
**P** ROFESSIONAL  
**U** NIVERSITY

*Transforming Education Transforming India*

# **Tittle-Cloud Storage Application**

**Subtitle: Secure File Upload, Encryption, and Management**

Group members- Alen P Shyju

Amar Kumar Nayak

Yash Raj

Mohamed Zoetyande

Date:11-11-2024

# Title: Cloud Storage Application

**Subtitle:** Secure File Upload, Encryption, and Management

**GitHub:** <https://github.com/AlenShyju/cloud--File-storage-app>

---

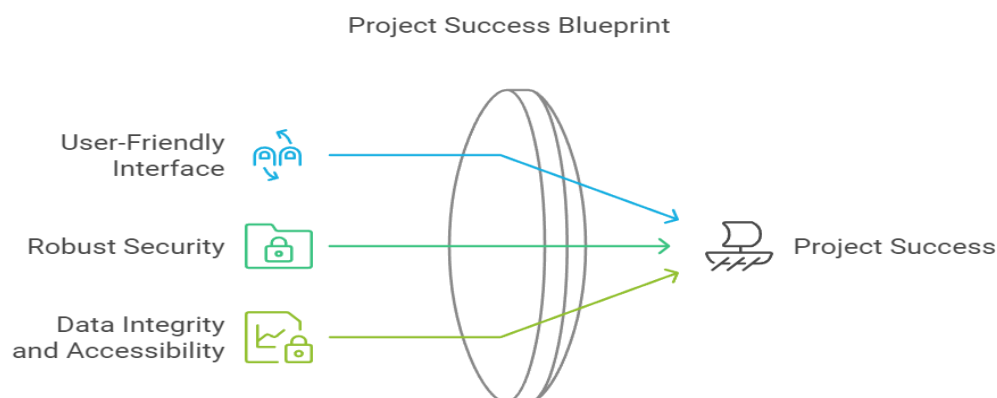
## Introduction to Cloud Storage

Cloud storage provides a modern solution for saving and accessing files via remote servers on the internet. Instead of storing files locally, users can store their data on cloud servers, ensuring access from any device, at any time, from any location. This enhances flexibility, data redundancy, and ease of sharing. The goal of this project is to create a secure cloud storage solution that allows users to upload, manage, and secure their files through encryption and an intuitive user interface.

## Project Objectives

The primary objectives of this project are:

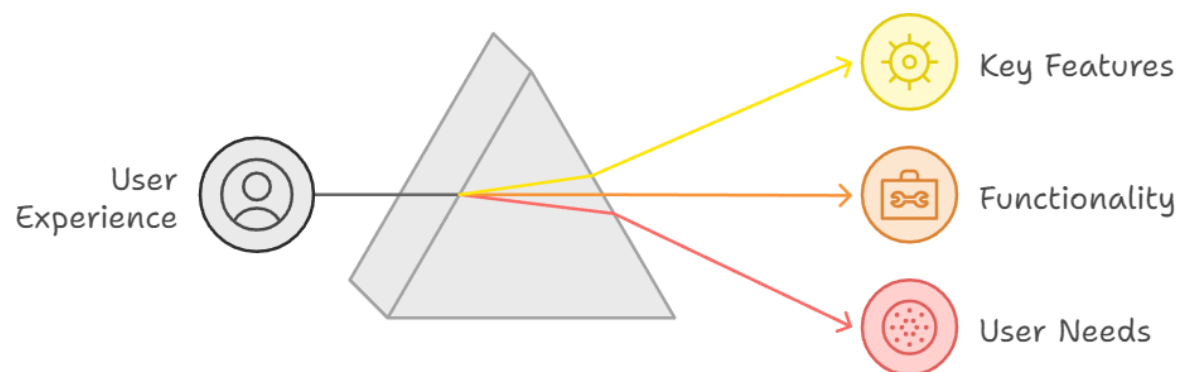
- **User-Friendly Interface:** Design a simple and intuitive user interface that allows seamless file uploads and management.
- **Robust Security:** Ensure the safety of user data by implementing encryption for files and using OTP-based authentication for secure access.
- **Data Integrity and Accessibility:** Allow users to manage and access their files easily, without compromising security or data integrity.



## Key Features Overview

The application focuses on these core features:

- **File Upload and Encryption:** Users can upload files that are immediately encrypted using secure key mechanisms to ensure data protection.
- **File Management:** Includes functionalities such as viewing, downloading, deleting, and sharing files from within the application interface.
- **Secure Access Control:** Files are only accessible to authorized users through decryption keys, ensuring that data remains private.
- **OTP Authentication:** An OTP is sent to the user's email during login, enhancing the security of user authentication.

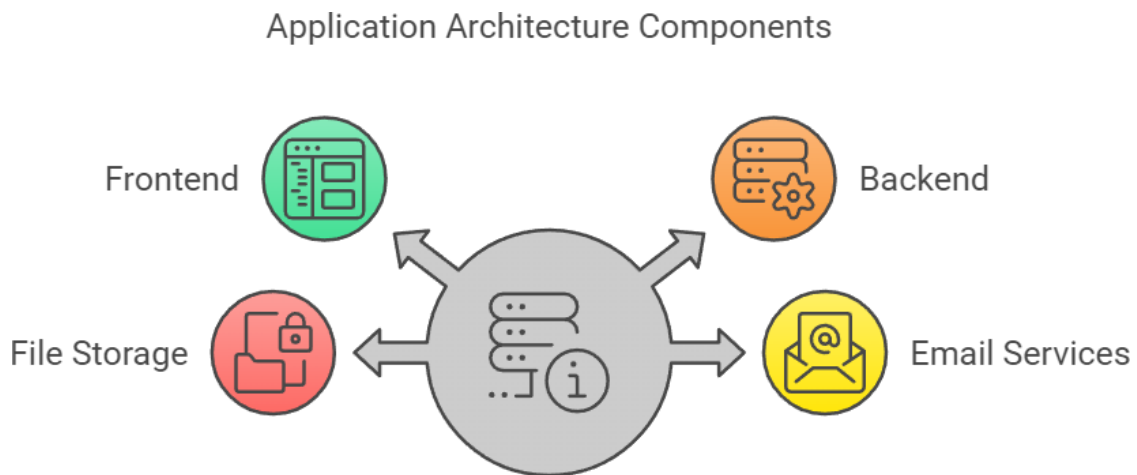


## Application Architecture

The application follows a client-server architecture, ensuring scalability and easy integration:

- **Frontend:** Built using HTML, CSS, and Bootstrap for a clean, responsive design.
- **Backend:** Powered by Flask (Python web framework) to manage sessions, file uploads, and user routing.

- **File Storage:** Files are stored in a local directory with encryption to maintain privacy.
- **Email Services:** The application integrates Flask-Mail for sending OTPs and user notifications.



## User Interface Design

The user interface is designed to prioritize simplicity and functionality:

- **File Upload Section:** Allows users to upload files easily with the option for encryption.
- **File List Display:** Shows all uploaded files with options to download, delete, or view them.
- **Modal Pop-ups:** Used for critical actions like deleting or downloading files, ensuring smooth and intuitive interaction.

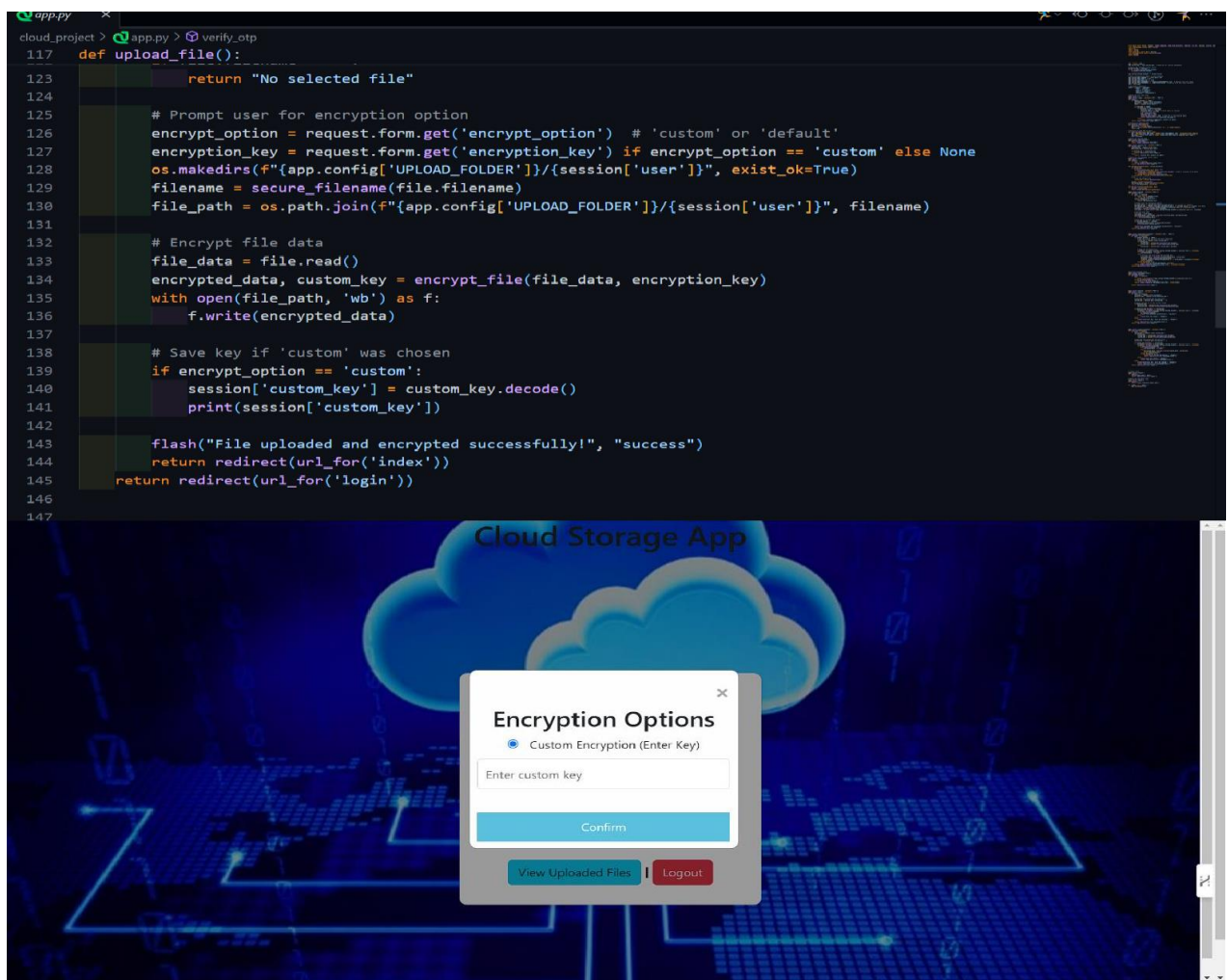
```
index.html x
cloud_project > templates > index.html
31 <body>
58 <div id="encryptionModal" class="modal">
77 </div>
78 <script>
79 // Function to open the modal
80 function openModal() {
81     document.getElementById("encryptionModal").style.display = "block";
82 }
83 // Function to close the modal
84 function closeModal() {
85     document.getElementById("encryptionModal").style.display = "none";
86 }
87 // Function to enable/disable custom key input
88 function toggleKeyInput(enable) {
89     const keyInput = document.getElementById("customKeyInput");
90     keyInput.style.display = enable ? 'block' : 'none';
91     if (!enable) {
92         keyInput.value = ''; // Clear the input if default option is selected
93     }
94 }
95 // Function to handle modal form submission
96 function submitEncryptionForm() {
97     // Get the encryption option and key
98     const encryptionOption = document.querySelector('input[name="encrypt_option"]:checked').value;
99     const customKey = document.getElementById("customKeyInput").value;
100     // Append the encryption option and key to the main form before submitting
101     const uploadForm = document.getElementById("uploadForm");
102     const optionInput = document.createElement("input"); // Yash Raj, 2 days ago • new update
103     optionInput.setAttribute("type", "hidden");
104     optionInput.setAttribute("name", "encrypt_option");
```



# File Encryption Mechanism

File encryption is a critical security feature. The application uses **Fernet symmetric encryption** from the cryptography library to ensure files are secured:

- Files are encrypted upon upload, either using a **default key** or a **custom key** provided by the user.
- Decryption requires the corresponding key to access the file during download or viewing.



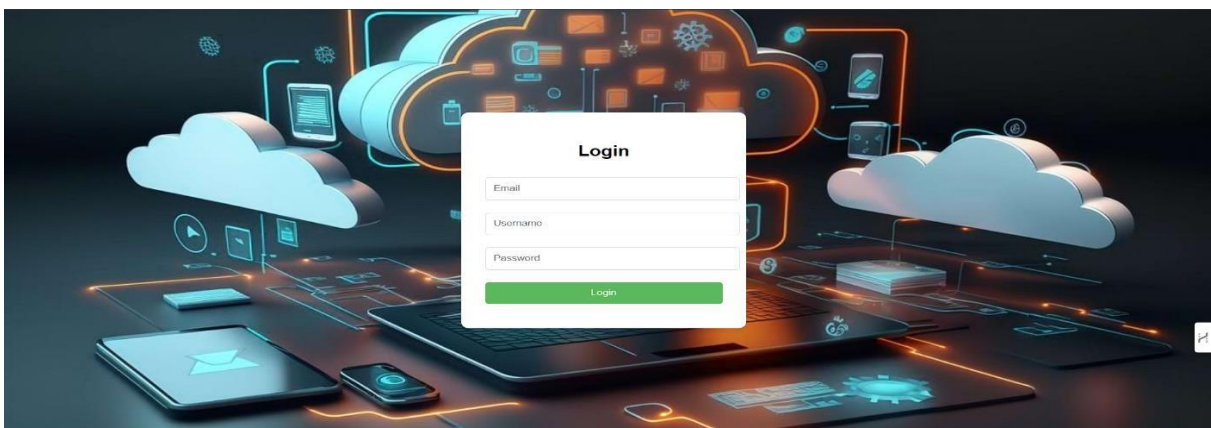


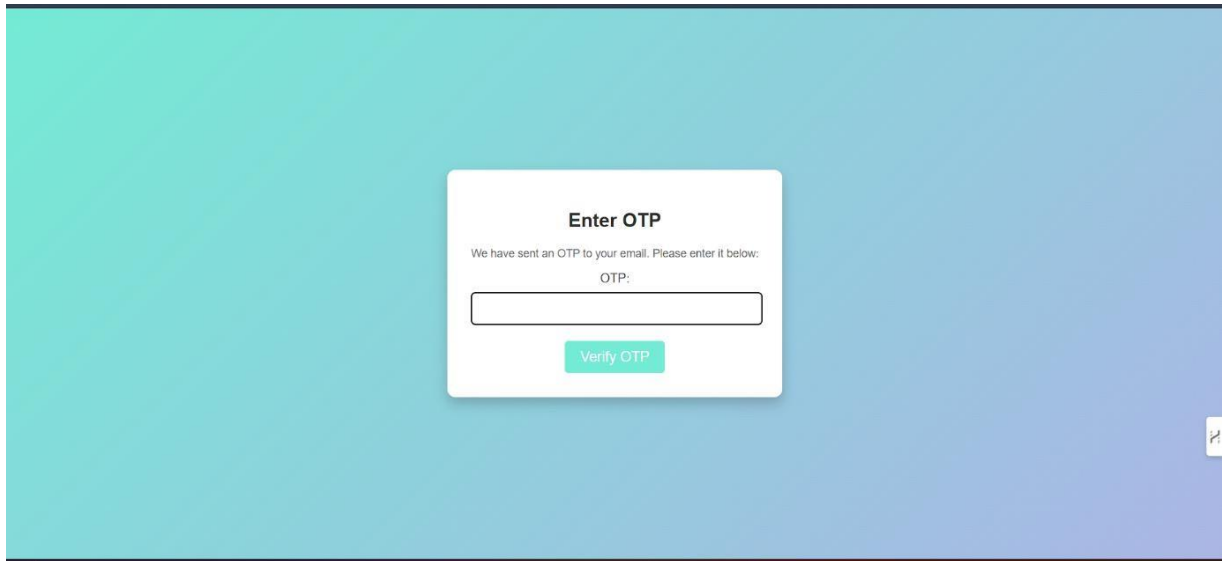
# Authentication and Security

The application adopts a two-step authentication process:

1. **Username and Password Verification:** User credentials are validated against stored credentials.
2. **OTP Verification:** After login, an OTP is sent to the user's registered email for verification, adding an extra layer of security to prevent unauthorized access.

```
cloud_project > app.py > ...
68 # Function to send OTP to email
69 def send_otp_email(user_email, otp):
70     msg = Message("Your OTP Code", sender="your_email@gmail.com", recipients=[user_email])
71     msg.body = f"Your OTP code is {otp}. Please use this code to complete your login."
72     mail.send(msg)
73
74 # Route for the OTP page
75 @app.route('/generate_otp')
76 def generate_otp_page():
77     return render_template('otp.html')
78
79 @app.route('/verify_otp', methods=['POST'])
80 def verify_otp():
81     entered_otp = request.form['otp']
82     generated_otp = session.get('otp')
83
84     if entered_otp == generated_otp:
85         return redirect(url_for('index'))
86     else:
87         return 'Invalid OTP, please try again.'
88
89 # Route for the homepage (after login)
90 @app.route('/index')
91 def index():
92     if 'user' in session:
93         return render_template('index.html')
94     return redirect(url_for('login'))
95
96 def encrypt_file(file_data, custom_key=None):
97     if custom_key:
```

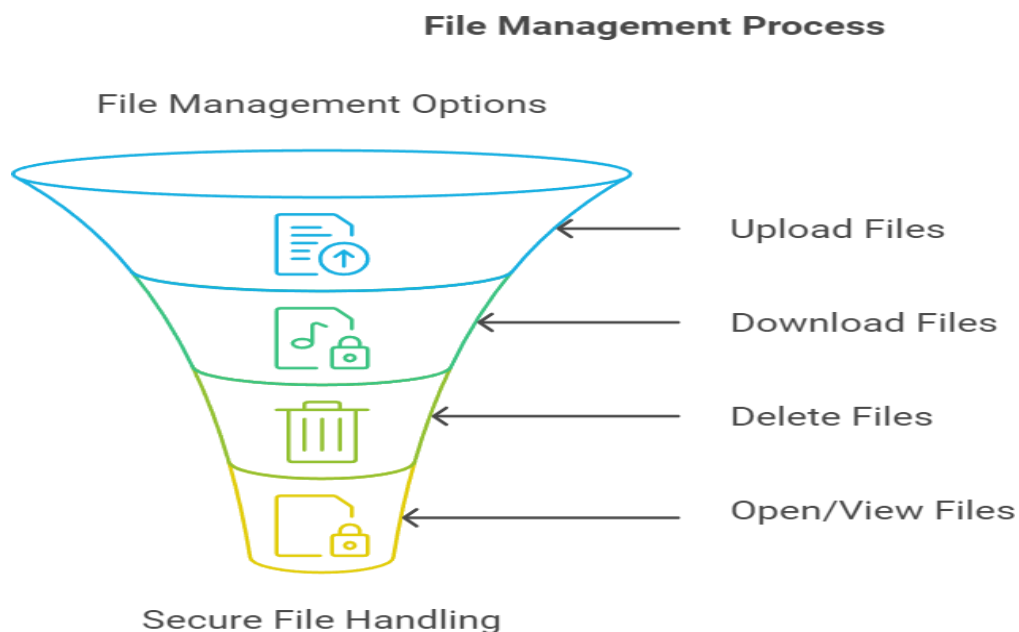




## File Management Features

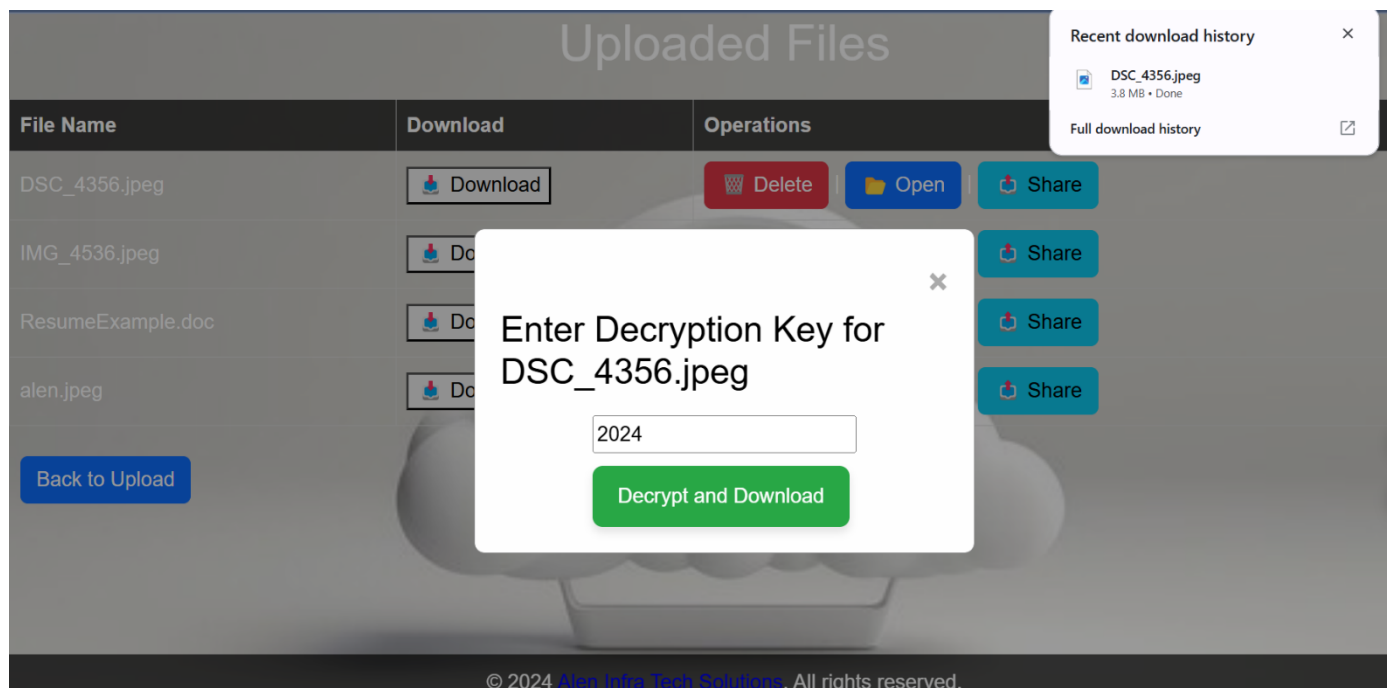
Users have complete control over their files:

- **Upload:** Files can be uploaded securely with optional encryption.
- **Download:** Files can be retrieved after entering the appropriate decryption key.
- **Delete:** Files can be permanently removed, ensuring data is securely wiped.
- **Open/View:** Encrypted files can be decrypted and viewed within the application interface.

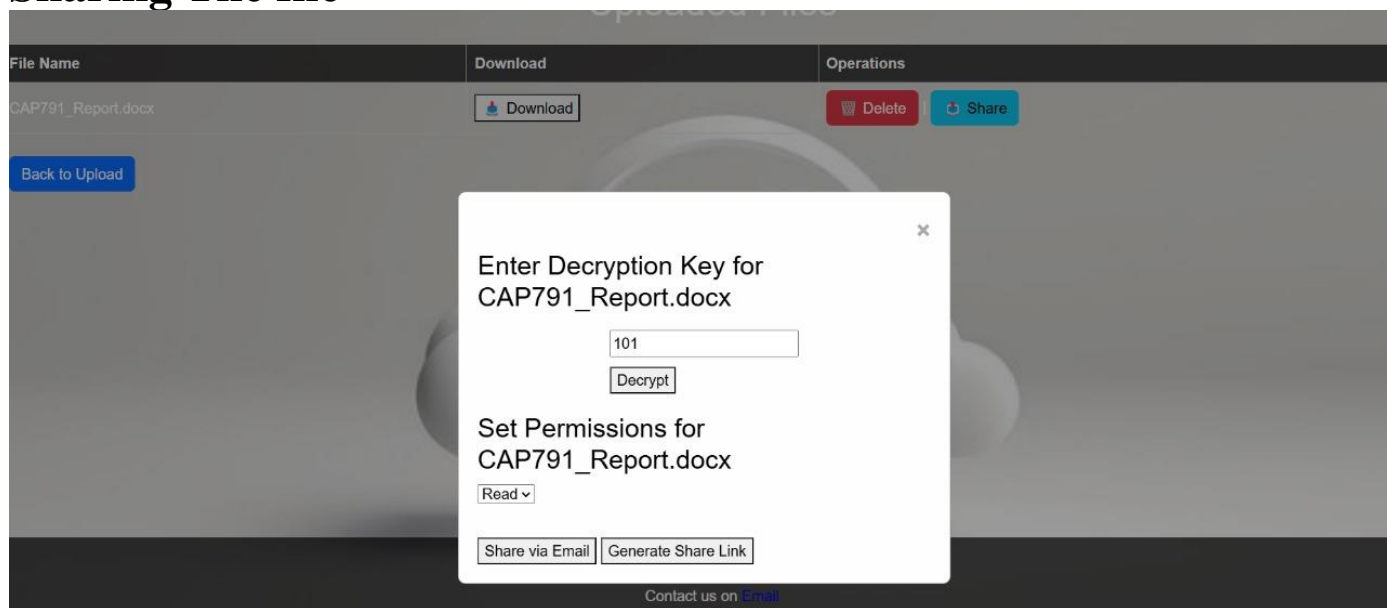




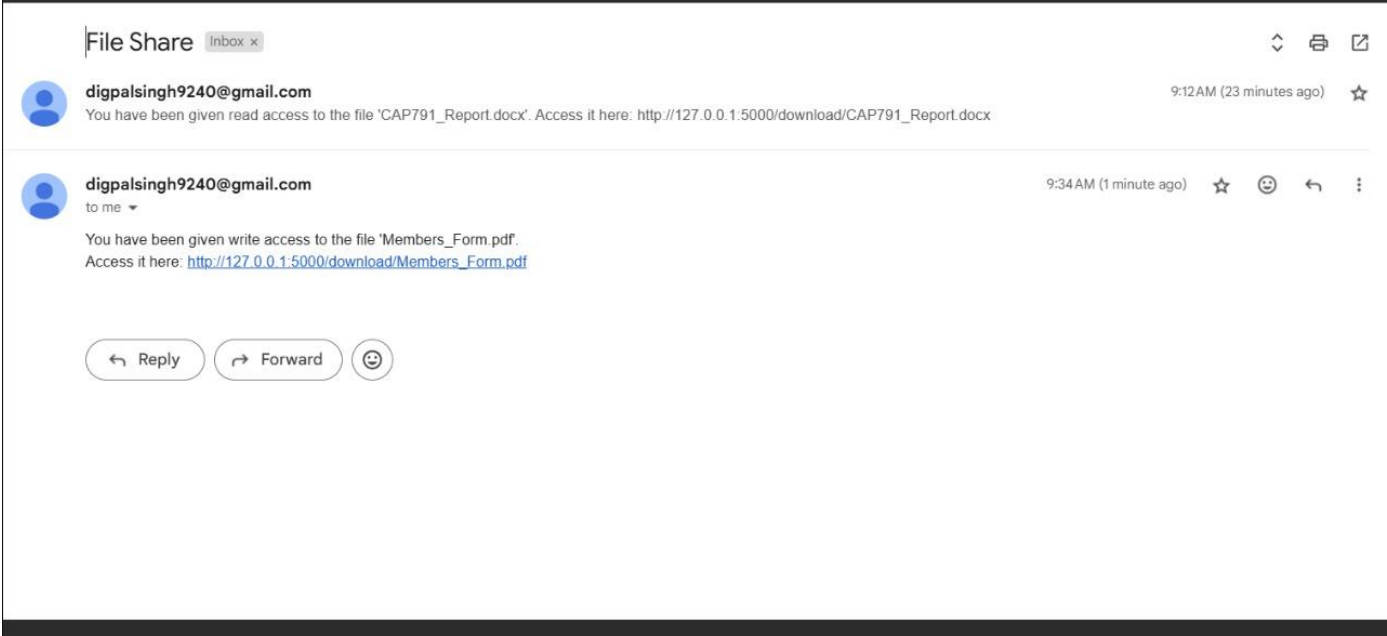




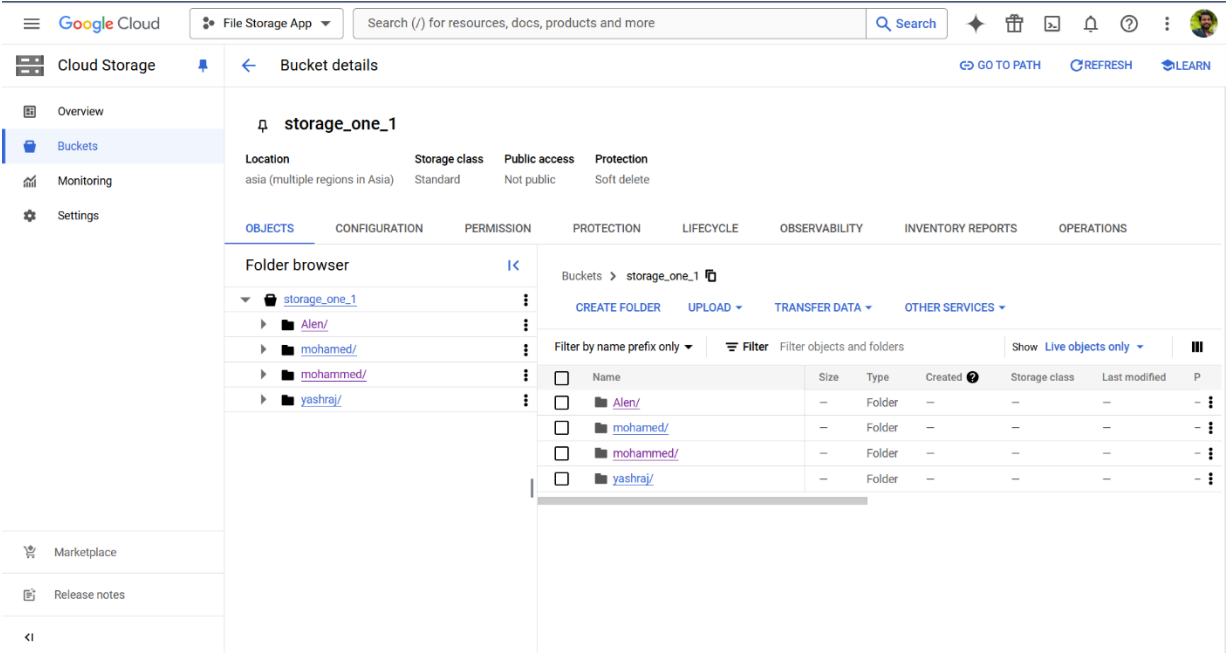
## Sharing The file

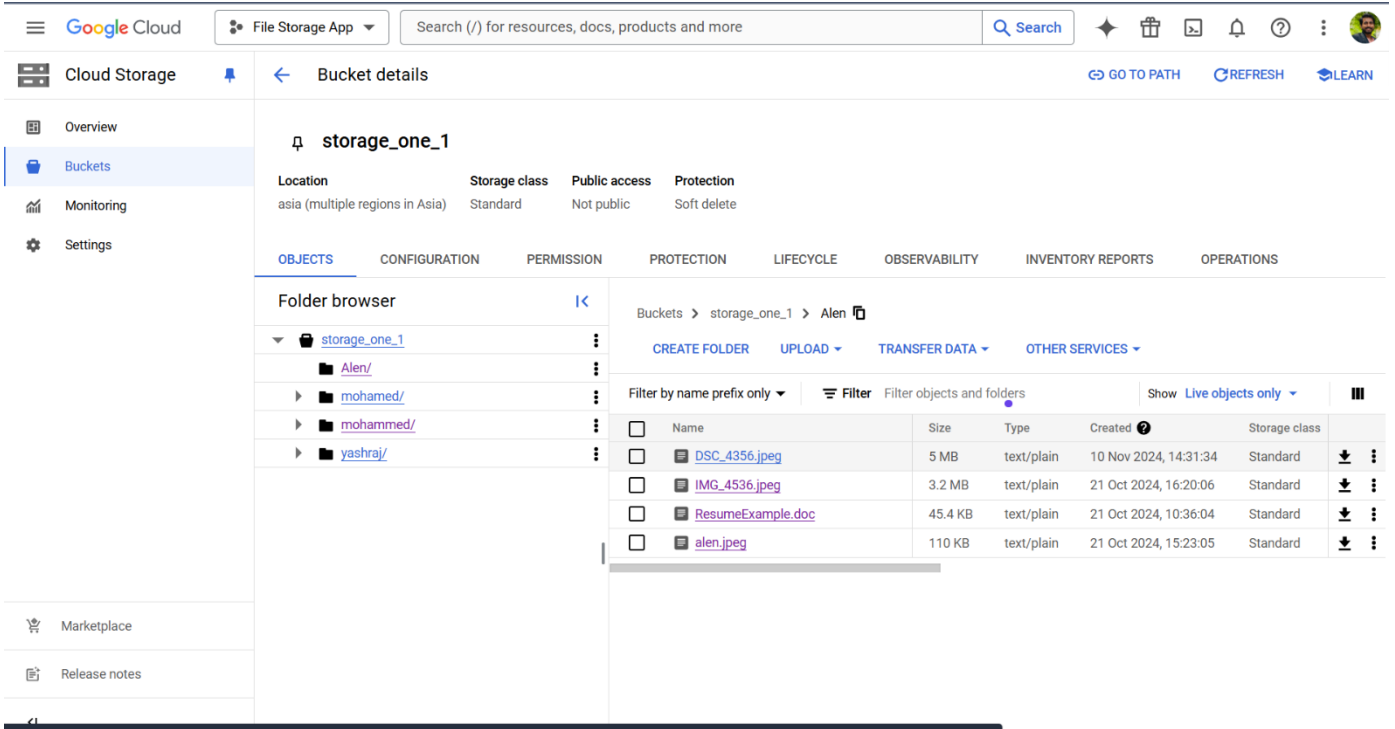


# Acesssing Through Email



# Backend , Google cloud





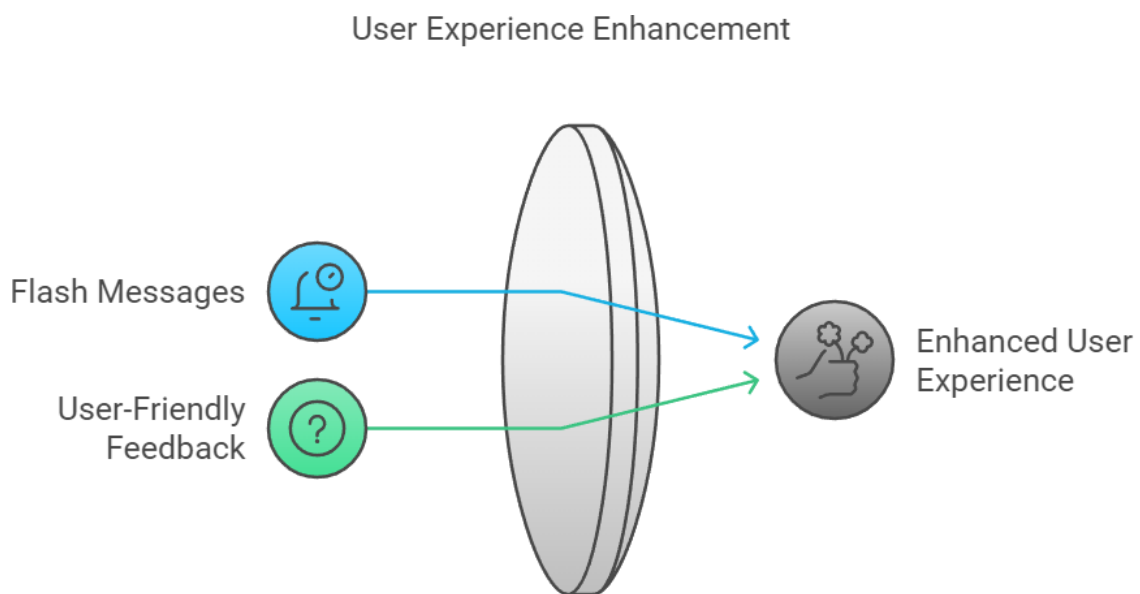
## File cannot be decrypted by Admin

gAAAAABnHZZINg6zhxa-BhEFeu-3u02b29Vx4R5W\_40gHXNikyqgV7Gzyj6hXXY1GEdxkJo61pdixGiTtYmNsUjpy-jAvisFEW5SHwKpTg6bXpKf66PPTJKWUm6\_Tl3I9cDMWjsw07IYMic2\_E2Cg3hMoof5Jpodd-Su9Uy\_zWsyUyubgA6k6\_JA5hs5jZxpydltangS0nmfJbklfwi1Bshu4Cr0mXlQremN7e8xQIEbkUuLaBgZEmGmLasZWLZgA1fAQEGKXTMGCFR4ByC2nyPF122\_oG59mEXUuWfWbXNessL7g0D2A3CBj9s2me\_NUUC\_mCjQcFabnsIa1a4m6N6cOfvrtVtcZp-50p\_ky6wZ2u7PjI30VRLtr3XJAxyxbC6w71nwrIP310Tzsh76471iof8EKkgNcL17KL-Qk0NRCL0p78u8mW1LZ9X052Tdeh61LXCvFC7J1u6GUKjshKv9ghZIXH761RAET533C7deZFLF2U4eWJ1JmgwA6Bz8YnjtZc5MTKF9qv2Y9H25K0hJ42FHU8P6bNwyhmLwZ711M0mTPMRRYedAw4678u55jmvw\_c9Ii1jLfcDl0cb2UmjD17FxbQw6G18TFbmKvHkdeEhn-XxPAjMke0MgJ08h-n0Esv701c7iKkAog9bpg35gBqtr92qNRPuIvcdDIRVKJ54GAD0eIeJvnhse7ZxxuRqtut6KXW42xvpndxte3qYtXiZAdnUwAFbERRAKm5ro17u8N8yVujpH9qMq3aYuUzj1StI6TeDTSkZT63nb84BbwRFHScMndKVPfVpdt7hLE1kd-IxiFFXTqUPPjIBKGrK7V5K08pKZXUk-XpQ8K7eYkZ5rr4\_T0gJ8X3T2cdLrJH8t0nb12FbTqmS5jbv4u\_FzsqgKCIUXADu1jaU9HJRTIgg9VYzrrZn\_CiyvDUpdgGBzF78Y\_syzn0vYIqVKnFv7DmUuH9UjdU3wVAUkQk0lhd0PSm7cF02ERP4XgMg77T-VhgEPV02IEQec6BHX77h2KvYd0u6vWm8-Duxew7jocmydbHb10GvYUN9zrg0Fr382AA0ew\_Ffhqi\_KyspUg3RjmvRiY8\_hf33LrSycav2h0CDTgkr2Rfglbrjmg8myxnn9IDgcalI433m0qrboe4p5CQmLkZC1EL-pv1I0RcB8M9L0h3tBDKA0gcm0ia-TlZjgCFNZTw2Z5X8trb0SPyEa3tnBQrQEVN90cDm0f8nu1Jq0Q\_5N3Gdgq3TMA-s5qKnxhK3t0v\_7XJ\_U0J0a08yKvYVQf2gthQI1knjokbNlYt1cekd1f7dQzm\_FnJ3u1Jtcg3cPNZxVQoAYRZMTBhYnmFRp62LWMOQE\_tpSwGRnenE1wG47VklKw4XLv364qxmZej0dTkrvHufG41Ysr9\_U-vMy-ebY586E\_FZ4F0PmWdKQYQR21aEccqofIqEhyalrsGEMZ1Z7CCfCL-ETVMLQvur02ZFAX8\_DQ\_wqM3AuFCfATC\_Ail-d\_AqLUSQdHx3ygm9x8W4SyyeXm\_lvyvZK6k91tyGp32CW6gJEWJH3u3RuM3RTE708w60QFPHL535Xj-ZX2rpZgAaAR0XMYQHnbnh12xwZi06\_s3TPIhyzq01zK0bjMqmdotpcL2AUKofITZPYaeag7I9HnMep9mXAwH14HEeStMKfwq2LfyPdp4r42KCI45YH5UKcT6fANm1PA0uN3\_NN1RYNtV-Rr20Zxued7JW1HmhbnxfqhL11E68mVYxdgd210b911-vfcsEqLkXafHIdw1s2ndfeqHdNY4Gx0q101xwN9pJhNIDgmx7UW\_FUX\_U5riicGZ3ciTzC0pA9K0UFc8N1gP77h2Uj0zPQLB8\_EXvoey\_be76bcyxmfw41Z\_nmhm5SQRoYDnBbj3Y3R00v0EnfPbM8eqJ4-INVATZc9udv-FexINfEKnrSPQX424QXAH9M0AQNi2-R0cf6Z-ik3wBDJvg1R0PwK0Gj1IIFmG-U0bXmQ\_HcZ9K3EY13DR15eXatX\_Fza3Ame\_jz6bht2orVjsxzq7or0B3ME9HpCEQXNBERgZ8YXGbisFv02kr8p1Z7F7U\_cBEHPhng56VATnesjCQwus3CQVg3ibNtW4X7T0k14ma1Ts70Rendga08Gu90tdAhpzKvQvohdf4Fh-YQdht0eyxms9U1EUJg9bn0d1kNly79v20Cw73mUttel1VhuicfT500XTG7\_Kep1YNeowegf1mJImymnfY4THR211-dATUpnrQ3JeFdvXw5h8K18A1mwfqdrFqkdvHm810vexX964uvJZIme0V8WwJ0J1d5H4G1s4H4BkN1gK3G52y0QuyyJ2fW1VtV1B9KxAM9KwB1\_80x810Vkv0gI3HC\_c01gkFb16h1\_aXJ4XtZj8BfgosF0N11DUWqZC8tVADZ3ff5ujM0D26wJ7jN5cgmK36uqVfR60j173yQ3K6MLNwKglz-1Tm0CYX9\_ExmWenI1hu09wAH2ZzX35YfF83b5kwd1Xk8e0ZIDWfay1fj1mCLelubQ007hygoXoT1p7aUaBtmrcLexBm1md3EY1V7d8KT2yKSmuruj-KscrUzCtJ01DEH5vBMBK0CK1ZZJN95Ygnqt3ASDHCE9meZSMZ7BfW43TMQfnul3VJfGbj09HABYV0iZCbwhanw-1H8bQ18ymf-wiqw10x-EHD055hwJ7ZdYVxkUHGQfjmbFCY2Z61NRZb-017RandF606LZET7tbb8UfDeKU1rEdqsyi-ITjztZJt93COPH34dKasPmF71L8JMA6LhUCG1gm10BPRH8yGTM8UBH8yflUwqMHTZhusuSeka53CfV31RdfHmH2FVTKAD07st1n824R-b0PM08rrS8A0sT4\_7r3yMAKZ\_B\_Kd\_sMAI0VEjtpmbbnzysvD2f0b5HeMf27eh\_0TPmP0PRkq195AuZdHXXrIvETHjwHgvNvX1MIdE5CnBPuII4HsQ7YnrPmskvf10xtI6L4HagV6VVLVAXM4U-4HvUXIF-x5Mm23S80ack47g8L12t0KvPlm190u2J528taX9M11xfdrpfD7IPHD0A0UJY56jwc0TE0Za1N8d65AXP1Ug2Nc1ZVPaArdcumjwckPm\_gcQTq5NQ4qy017KXLIWgEikwTq4nDgm9GhdV6nFhwAm79L52hL5pp\_fg5AgK6E9HKYX5Tetx3Sm78cmadeW27Xs10mwk3952DGZu062\_Mccan205x0iff7FRQ5X11Aua8W\_L85ZERmDvY17ur-5xmhFjCUB1c3iivEdsoqQhH7g7L2xcPvS\_mf0aYfV\_uTSZNLZ0M9p2fy0Z7jd6NHZmowj6TzPn2326ego\_2M2IJ09aMR10n\_c9rYgtjHnxuxfE-T8QyuAK3ig1ccclgegeyKrJRXV61C7fJah15yaa8pRyt8B2IKr6cNEZRI1MfQvr7u56gJGsuBc060Bvnp1Yktoc70Hmsq51C\_R6hsdeZfYqXScwLxjNRu5wgx55FAd7PHGc7mas17sDjWrfJ7h0oQ0L0ks00H2yKzQ4Q0Q0RM4g5SMVWk4pXK1w0Z-Ep9ym1-01rLx0uWcGM13x02p27d5HUKVg\_CIR4bg-IzP\_f7nTFH8gH1mZ\_gK9Eaoh4Tjp8v8r3P0XEDfMwJHvKXfHQ4beetUsT5etCUUSTJH4NuLEpe6GLw0jild1M19MYG8X4q5d82B5bnVT014DNDp6erG7aJG1-F30XXmZn6gkd8BNE1Z5YA3Va6a7NK96v1h7EdJvU-Nyaph\_sMrdcsCDptm19PAQ8B1GHf\_Lccy\_IqysyPFIlyfe7J7F18dLGit4H8B2Cgz7HmV15Ucmacgq40pvrPqPwKz3j06wYDHx9j6tvP41TLDFf5pKv2QXaR3sEDGEMwTFQQQQ-pvY1ZwFzAmE1LTK0t0e1eBf8-12F18ReeMLrxYXZJ32ez2bKZ2Pz6f83Uak1taoffFXZjKs6a8LXFN5\_0a55p0ZGXiCpJb-omecTCGSMTRSL295mpagBRNjZyQeQFC7nnjXows7C-eP62L-FX1T34KMEsG034c38A515K5rkq7bv1tktytGZIdw76XQ05R9Cv\_Fln58mb04P0X7dAy0FMLHJ10Sb5U75q5mhrZdyPLHCc2U215AUAEZchmG1-ftoqX60n80M5Qezps0T0TB31Mpxykwbykyio3NS1KrbW\_QE0vCmC-dusxZ3VnCr10wsRravoxA0b8Hq3Q8G02hdLSKYCsuhtXnR1R8DR6Xjy4c9vmdgJvZy7t0BMSR9HVOuyg1KgfPk8E18UJfarg1T0AQ7XhHLB-tlesz9Fkw1-dqv6A3Zag5npJgPjue7Kdp6AKxbC78n4de\_UvqQ1kct10B0p99mFmaE1KRYDwu27vcGjURPphdWYf1\_2D0Z48CkaotSvYqE10Gf4YD81RV7TqX-YdwQe1lgvM3R3erJUBabAKseypQke7Wj9G-Xuk6e65S859D8KXSGutono1B8p30ZTEKTS1Cnd4a4XqAphtClupQm2J0Qj01V5xutJgRGL9vCPXQZsqa01wvpPQFmVR4U1MXVPLYjwE390URL-2E3ocaTnkB9ACHgXcr3G1PvV4uNOCwQUT950rJgKewTouVQ9JwgKD08uUuYhY8K14dm-Qghp-ye5AoffHnZd2XDKHC0i630tSmvH32PB8BbhcUc5APHTN6wJ9eRh4Ygh7jdxqM1dRgvsE34cM3fUy1s4TbN6DzB7qohV1Yhh4CnctCfQ1H2L4qy6w6esaQDf-4aw3So\_1K0bP8urd2t0rM1EXN5G3mH5nHPVRYfBmHurnTY\_Y50K1K7w8YPL1Yh8-7B18u1\_T55vTb7214fy161fjwV1u0ey6eAtJ\_1VJ0cI0k4tyGdXG1vLHm1K11b30fN5K1To1oJh1f6PLXue5ZrntAaC1Jhe0DuXhF3u2hgQJVB81\_EK8X819H9HC5KZJW71\_UCAqQk13YpYyIVaG0G4Huoc0tACL2f4Z6x2nsDK0cZf-50R16GubqJ9Bf0G5ZB7V00R601f-TUGeCZLkvw8F8KSHJh10710NAK378gJ7H5QXTu02qWJH1Rc80VNI7840Sg1bl77IeyU2v5KQrB7bBW\_IWuVYNGD\_FSEE\_PbzJ1\_TQUqH3s1d03uA52dfBM3AfMGXELPAVkvon3bs0Ueqid90ARR\_UwF9XSuaVC4NmQe0pw50Sdw1tPwAGnJKSmpA1K35S08Z18WPyhsyGHTZ\_NLX8K0337fno2wZKysj-V0\_R3cVmw-6K0hp4Fkl2exrP0mWk3jyH1Z0-L0JLWYH1K8XegXsYujChta01lEvoJ310xaoVgXfD-SW08ZrnhE10uPrKcEV\_atdev4uKMEtNgU0HhN7uX6\_X2AZZjN010pYGrT5\_Z8GL130CQeZcfsoD85yhkJnxvP1Z1Qzyak7I2yE1EmcSfN4tk5c\_zksaDa0acQQZBTk7M215mhkL7osK1L8c4fuFty3PMXGRYtd-M0Cw7X14CnQpX5NKRGSgE1wbuR-acAjaP25UymB0za9R155A818M2H13gc8WbH4Z7YK8SP1BZncl0F7ZSAkAeGfVWQ5gmRCRHzwe9nvaq-nzX5dc2N9D7VvW6j9nHr-Mbxc7G0L30uLksh1T851MLd2\_Kq1VW0X35ST1T0uF1ZSA91T50\_noycwhrd7Tme1qZ36fFL42ZhdhgZf65vTbC72mLgQv2eaykfqalW31u5C2E76GwQrGAZc8wTFNun-eCH1UjvWw0a00ndmNYJLczmf\_3QpdyJCT\_3n1j3q\_g08b15v7n7N1mp1fDyCgW7daQ3jwpoFyCB31Hj5M4P31La11G1AaGkbwvaQd3jAMFue0e70DHzg00P0S6LdqwvyK578H4Z7WQeGmXcJg\_fJ2T15LVU5B56m5FUYIOMi1An6G3jGfy2TrEMpihus-huezUHLTZg-Bk87D9E1xpgst\_xRBEfjbjat6kArwz\_EizkshW0R1yvQkqhH8R1qLB00ITf\_gKbP1KfYhmG0p3-CvzwK15\_MdWuH1JL-K5m5ekbP1aTChapiT0SeUoySuoX10H1gtz9LmK4j0P8HJG9J1zn6117P64a7fVjVrTKZ8BhU7D0Cpde1J17GjMR3Cra2ZhxxBjgc\_EnuZEsQ706dmsXU3k1PfhM4qgPKsvb31MoQAGxiGnqBBXUy1vqEmdsDhQ04KwQwznjAt5BukiKDKC0JWqfD2NwQuvYzGvPtF43Bmp19w2z1ATD0zXRS5HnKEZRH2EBN-Hnx0bnE9Fas3jUSRXhQf6GPAU1Aw4j320TfZ69yCmgZ3w55Xm1v1NA041j0p64NH3-awd51RNX05KxyR16210hPw1Jm139eK4ufuG5Z1ScfFHPAX53mH8Bh08H08UJHm8AKZerK1P1PXAHR5Z2EKQ5aps9E1M8Jum7dgwEC20-FEgR61zt-2FK-b5kmf-fPrVfK39a3dXayEG0pynbWQ11\_0zd1LYw370W0Z\_6MYlyF2AvYK-t1s2s7FACXvud5T311mHvJwL1hUvLV7W0Luz36F5Uv1XJ631-RVNS13VnmTRX713EU3wH5mK6H6X1VfX09mYrXrbvdmK2HHNGB-0uad08RVmGmLZCOKsc0E4D13ecXbX9m1R5gmKRTf-w55E90vJ5VMDze6T0GdY283UZ90zRLma-

## Error Handling and User Feedback

The application includes comprehensive error handling mechanisms to ensure a smooth user experience:

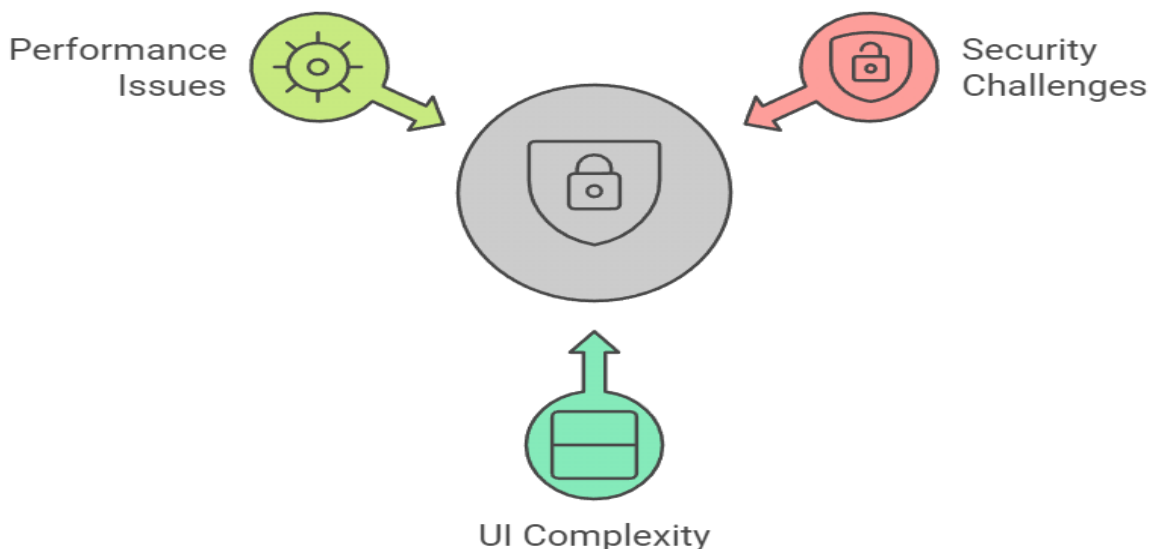
- **Flash Messages:** Users receive notifications for successful actions, errors, or failed operations such as incorrect decryption keys or missing files.
- **User-Friendly Feedback:** Keeps users informed about the outcome of their actions, enhancing trust and usability.



## Challenges Faced and Solutions Implemented

- **Security Challenges:** The biggest challenge was securing user data. This was resolved by implementing file encryption and OTP-based authentication.
- **User Interface Complexity:** Maintaining simplicity while offering advanced functionality was challenging. Using **Bootstrap** helped create a responsive and intuitive design.
- **Performance Issues:** The need for efficient file upload and download was addressed by optimizing file I/O operations and employing asynchronous handling.

### Challenges and Solutions in System Enhancement



## Future Enhancements and Roadmap

This project lays a foundation for future advancements, including:

- **User Roles:** Introducing different user roles (e.g., admin, user, guest) to control access and permissions.
- **Cloud Integration:** Extending the project to use cloud services like AWS or Google Cloud for scalable storage solutions.
- **Mobile App Development:** Creating mobile applications to enhance accessibility.
- **Enhanced Notifications:** Improving email notifications and logs to provide users with detailed activity tracking.



## CONCLUSION

The Cloud Storage Application offers a comprehensive, secure solution for users looking to store, manage, and share their files. With a strong focus on encryption, OTP authentication, and user-friendly design, this project ensures both ease of use and high-level security. The groundwork established here provides a pathway for future enhancements, including mobile development, cloud service integration, and more. This application represents a significant step toward developing robust cloud storage systems.

## REFERENCE

- Qian, L., Luo, Z., Du, Y., & Guo, L. (2009). Cloud computing: An overview. In *Cloud Computing: First International Conference, CloudCom 2009, Beijing, China, December 1-4, 2009. Proceedings 1* (pp. 626-631). Springer Berlin Heidelberg.
- Bisong, E., & Bisong, E. (2019). An overview of google cloud platform services. *Building Machine learning and deep learning models on google cloud platform: a comprehensive guide for beginners*, 7-10.
- Kulkarni, J., Sidnal, N., & Dandagi, V. (2024, June). Fernet Encryption: A Secure Approach for Edge Data Protection. In *2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT)* (pp. 1-4). IEEE.
- Bello, S. A., Oyedele, L. O., Akinade, O. O., Bilal, M., Delgado, J. M. D., Akanbi, L. A., ... & Owolabi, H. A. (2021). Cloud computing in construction industry: Use cases, benefits and challenges. *Automation in Construction*, 122, 103441.
- Alouffi, B., Hasnain, M., Alharbi, A., Alosaimi, W., Alyami, H., & Ayaz, M. (2021). A systematic literature review on cloud computing security: threats and mitigation strategies. *Ieee Access*, 9, 57792-57807.

- Khashan, O. A. (2020). Secure outsourcing and sharing of cloud data using a user-side encrypted file system. *IEEE Access*, 8, 210855-210867
- Bhargav, A. J. S., & Manhar, A. (2020). A review on cryptography in cloud computing. *International Journal of Scientific Research in Computer Science Engineering and Information Technology*, 6(6), 225-230.
- Gupta, K., Gupta, D., Prasad, S. K., & Johri, P. (2021, March). A Review on Cryptography based Data Security Techniques for the Cloud Computing. In *2021 International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE)* (pp. 1039-1044). IEEE
- Tsai, C. L., Chen, C. J., & Zhuang, D. J. (2012, June). Secure OTP and biometric verification scheme for mobile banking. In *2012 Third FTRA International Conference on Mobile, Ubiquitous, and Intelligent Computing* (pp. 138-141). IEEE.
- Lutz, M. (2010). *Programming python*. " O'Reilly Media, Inc.
- Dabbish, L., Stuart, C., Tsay, J., & Herbsleb, J. (2012, February). Social coding in GitHub: transparency and collaboration in an open software repository. In *Proceedings of the ACM 2012 confere*
- [https://github.com/yash8877/cloud\\_project/tree/main](https://github.com/yash8877/cloud_project/tree/main)