SecureVault – Cryptographic Data Vault

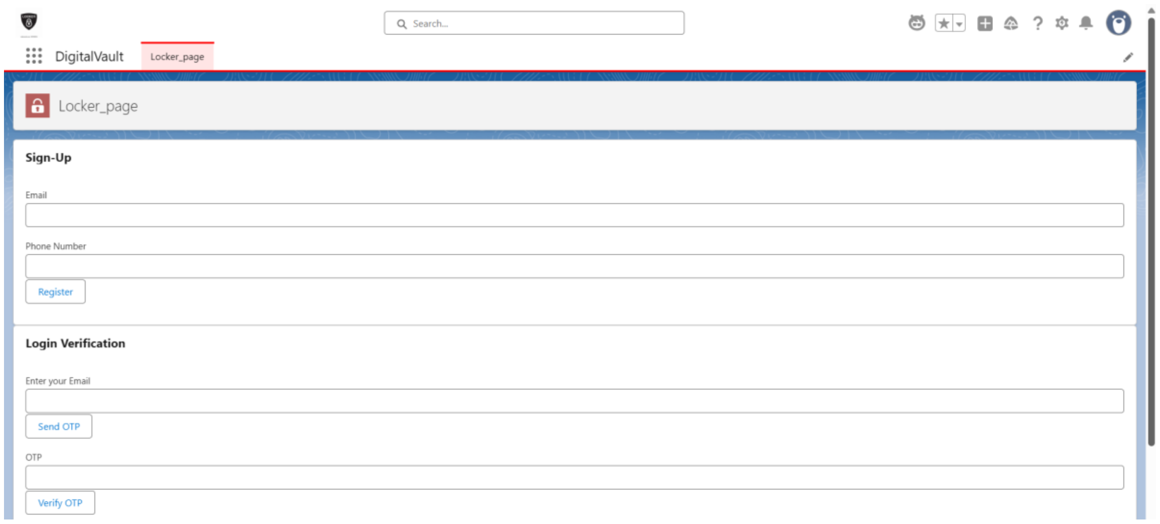
Phase 1: Problem Understanding & Industry Analysis

- **Requirement Gathering**
  - Users need to securely store and retrieve sensitive data.
  - The company must never have access to decrypted content.
  - OTP-based login required for authentication.
  - Users should be able to delete their files and accounts.
- **Stakeholder Analysis**
  - **End Users** → Encrypt, decrypt, and manage their own files.
  - **Admin/Org** → Provides the platform but cannot access decrypted data.
  - **Evaluators** → Validate the zero-trust architecture.
- **Business Process Mapping**
  - Sign-Up → OTP Login → Encrypt File → Store Encrypted Data → Decrypt with User Key → Delete File/Account.
- **Industry Use Case**
  - Relevant to industries like healthcare, finance, and SaaS where **data privacy and compliance** are critical.

Phase 2: Org Setup & Configuration

- Developer Org created for building and testing.
- Company profile and time zone configured for the company.
- Users set up for users.
- Deployment managed through **VS Code + SFDX** with rollback plans.
- Created two apps one for user end and another for company end.

USER END


DigitalVault Locker_page

Locker_page

Sign-Up

Email

Phone Number

Register

Login Verification

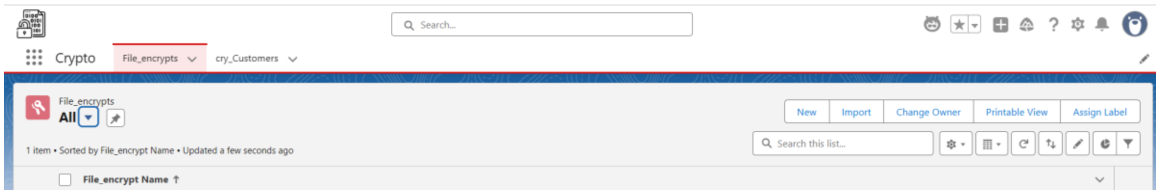
Enter your Email

Send OTP

OTP

Verify OTP

COMPANY/SERVICE END


Crypto File_encrypts cry_Customers

File_encrypts

All

New | Import | Change Owner | Printable View | Assign Label

1 item • Sorted by File_encrypt Name • Updated a few seconds ago

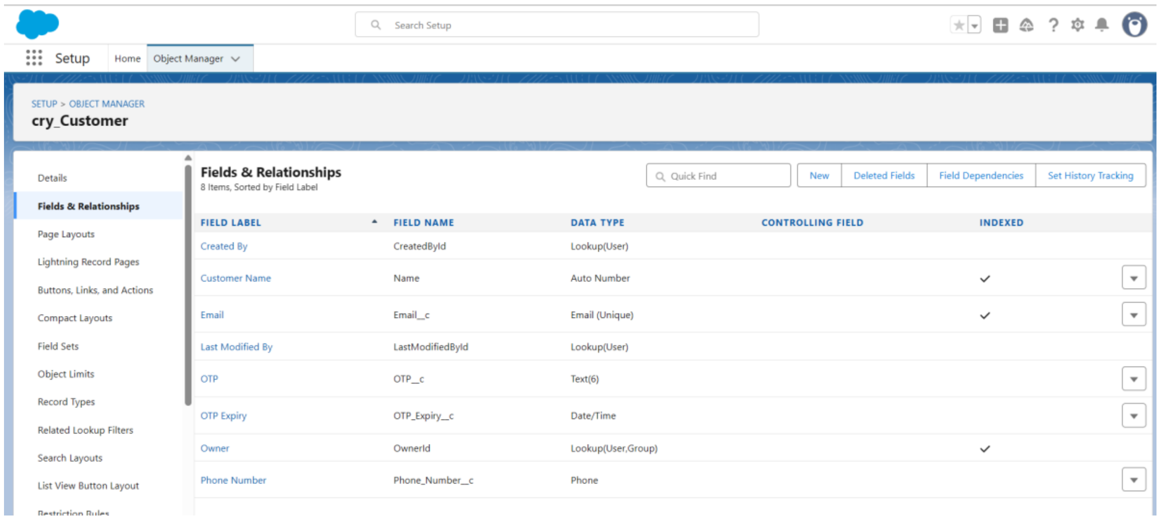
Search this list...

File_encrypt Name ↑

Phase 3: Data Modeling & Relationships

- **Custom Objects**
  - **Customer** → Stores Email, Phone, OTP, OTP Expiry.
  - **File Record** → Stores File Name, Encrypted Content, linked to Customer.
- **Fields**
  - Customer: Email, Phone, OTP, OTP Expiry.
  - File Record: File Name, Encrypted Content, Owner Email.
- **Relationships**
  - Lookup relationship between File Record and Customer.

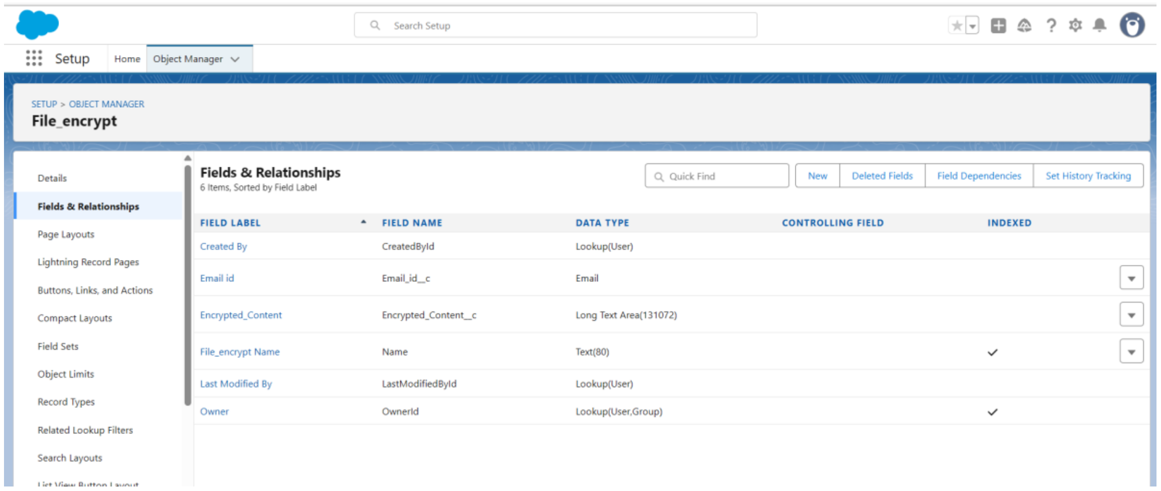
Two custom objects and their Fields and Relationships



cry_Customer

Fields & Relationships
8 Items, Sorted by Field Label

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedById	Lookup(User)		
Customer Name	Name	Auto Number		✓
Email	Email__c	Email (Unique)		✓
Last Modified By	LastModifiedById	Lookup(User)		
OTP	OTP__c	Text(6)		
OTP Expiry	OTP_Expiry__c	Date/Time		
Owner	OwnerId	Lookup(User,Group)		✓
Phone Number	Phone_Number__c	Phone		



File_encrypt

Fields & Relationships
6 Items, Sorted by Field Label

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedById	Lookup(User)		
Email id	Email_id__c	Email		
Encrypted_Content	Encrypted_Content__c	Long Text Area(131072)		
File_encrypt Name	Name	Text(80)		✓
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		✓

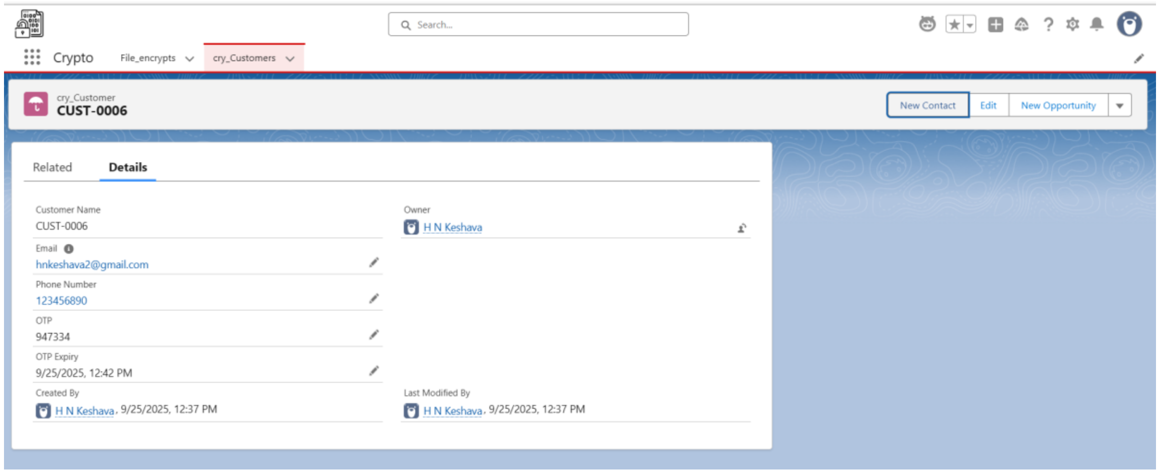
- **Validation**
  - Enforced uniqueness on Email and Phone.
  - OTP expires after 5 minutes.
- **Design Principles**
  - Minimal fields to reduce attack surface.
  - Zero-trust: no decrypted data or keys stored.
  - Used strong Encryption and Decryption Algorithms(AES-256).
  - Scalable for future features.

Phase 4: Process Automation (Admin)

- **Validation Rules** → Prevent duplicate accounts.
- **Email Alerts** → OTP delivery and encryption key sharing.

- **Field Updates** → OTP expiry set automatically.

Admin side for a customer with Customer id = CUST-0006 the otp and the expire of otp is shown and also storing the customer's gmail id and phone number



cry_Customer
CUST-0006

Customer Name
CUST-0006

Email
hnkeshava2@gmail.com

Phone Number
123456890

OTP
947334

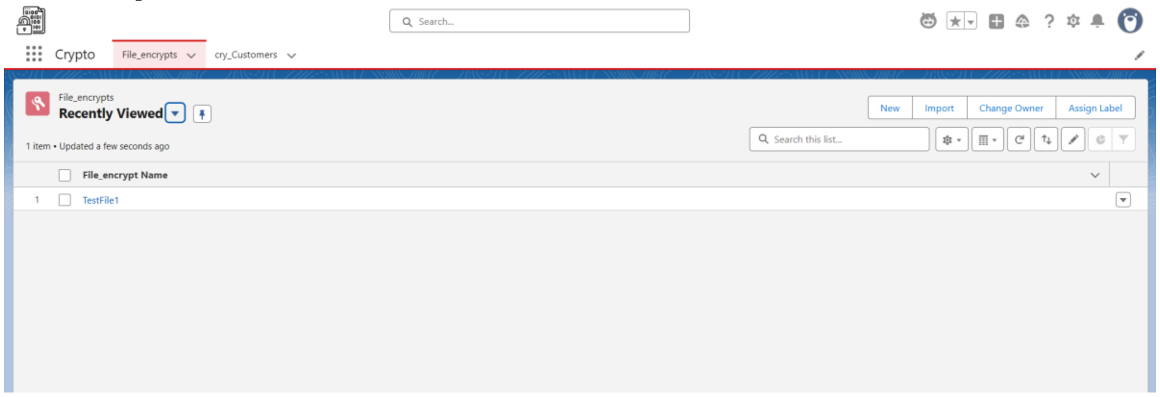
OTP Expiry
9/25/2025, 12:42 PM

Created By
H N Keshava, 9/25/2025, 12:37 PM

Owner
H N Keshava

Last Modified By
H N Keshava, 9/25/2025, 12:37 PM

Another part what admin sees is the encrypted files of user there can be multiple users with multiple files

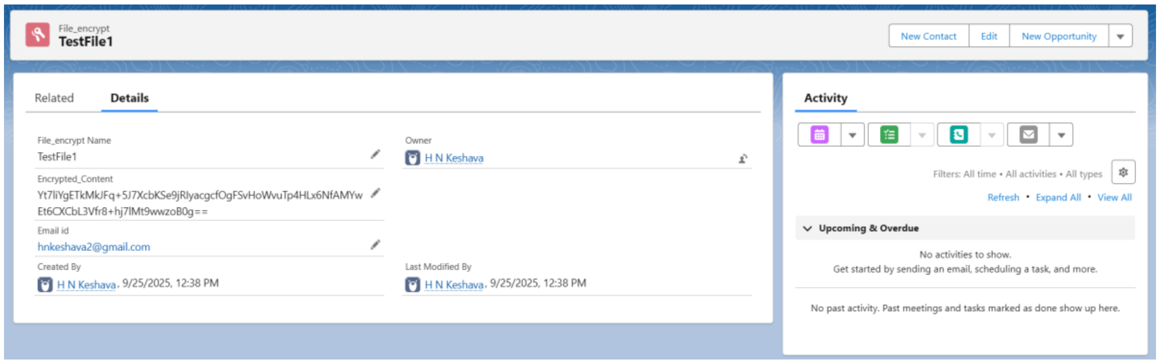


File_encrypts
Recently Viewed

1 item • Updated a few seconds ago

	File_encrypt Name
1	TestFile1

This is each encrypted file data and its details


New Contact
Edit
New Opportunity
Related
Details
Activity
Filters: All time • All activities • All types
Refresh • Expand All • View All
Upcoming & Overdue

File_encrypt
TestFile1

File_encrypt Name
TestFile1

Encrypted_Content
Yt7liYgETkMkJFq+5J7XcbKSe9jRlyacgcfOgFSvHoWvuTp4HLx6NfAMYw
Et6CXCbL3Vfr8+hj7lMt9wwzoB0g==

Email id
hnkeshava2@gmail.com

Created By
H N Keshava, 9/25/2025, 12:38 PM

Owner
H N Keshava

Last Modified By
H N Keshava, 9/25/2025, 12:38 PM

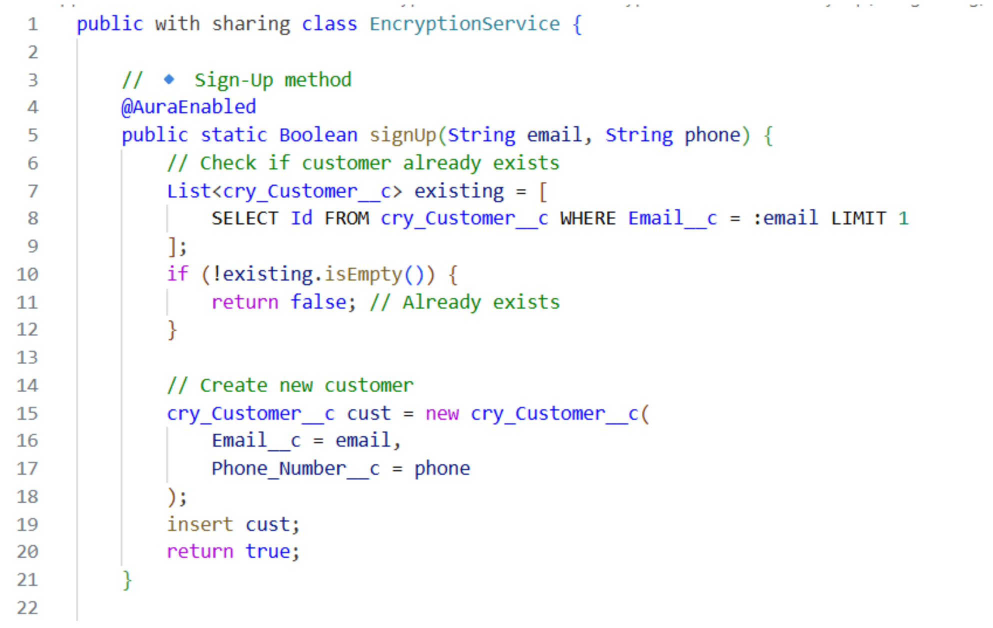
No activities to show.
Get started by sending an email, scheduling a task, and more.

No past activity. Past meetings and tasks marked as done show up here.

Phase 5: Apex Programming (Developer)

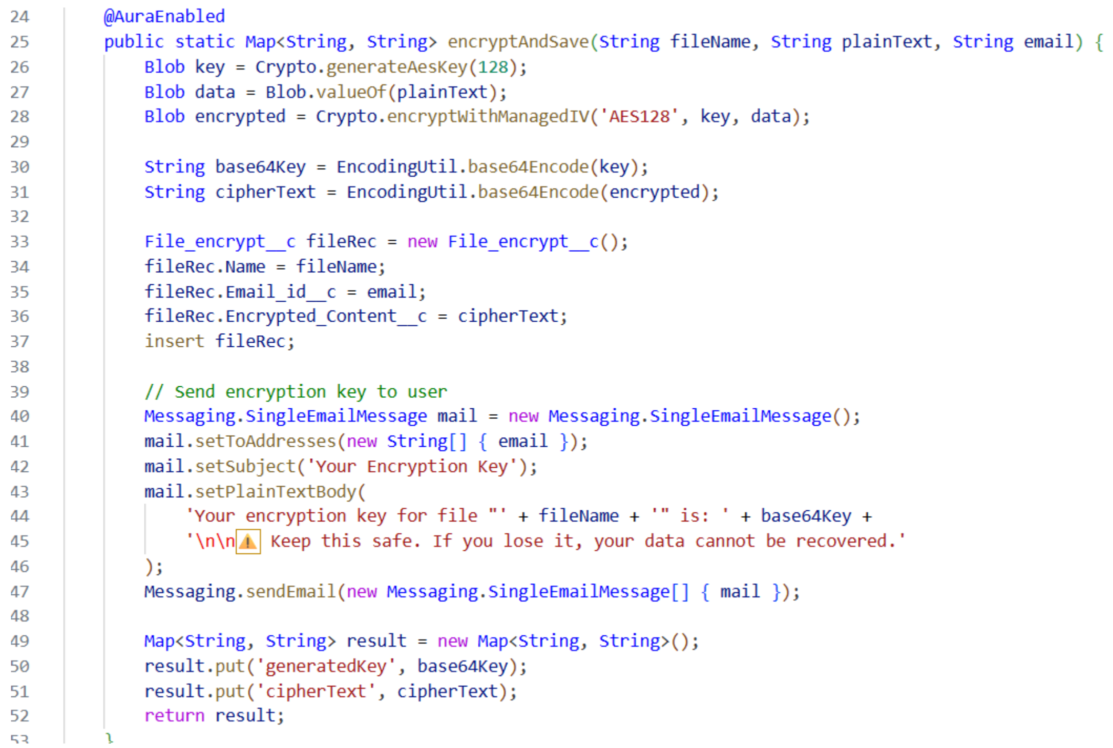
- **Classes** → Core logic for sign-up, login, encryption, decryption, file management.
- **Logic** → Ensure data integrity and enforce uniqueness.
- **Exception Handling** → Prevent invalid OTP or decryption attempts.
- **Test Classes** → Validate encryption, OTP, and file workflows.

This below code sees of the customer exists or not



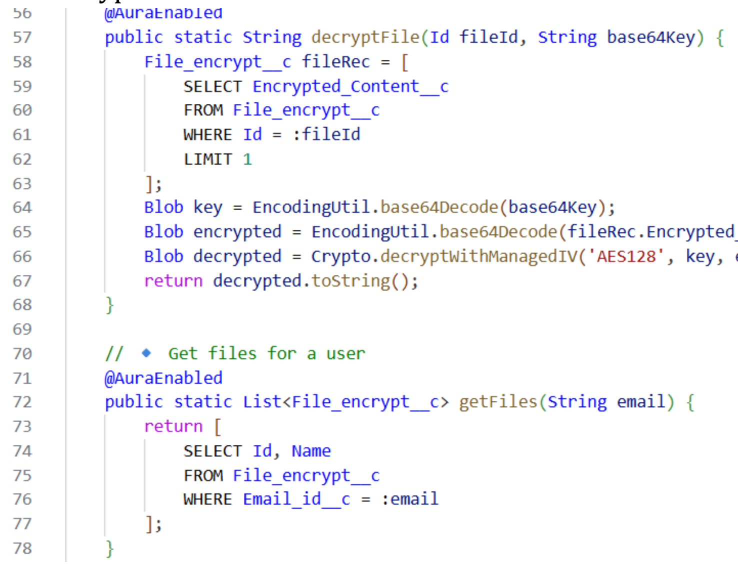
```
public with sharing class EncryptionService {

    // 🔹 Sign-Up method
    @AuraEnabled
    public static Boolean signUp(String email, String phone) {
        // Check if customer already exists
        List<cry_Customer__c> existing = [
            SELECT Id FROM cry_Customer__c WHERE Email__c = :email LIMIT 1
        ];
        if (!existing.isEmpty()) {
            return false; // Already exists
        }

        // Create new customer
        cry_Customer__c cust = new cry_Customer__c(
            Email__c = email,
            Phone_Number__c = phone
        );
        insert cust;
        return true;
    }

```

This the core Function 1 THE ENCRYPTION of data using one of the strongest algorithm AES-256. This also generates a Key used for encrypting the data and same key is used to decrypt also so this Key is sent to user’s gmail and this key will not be stored by the Service/Company side for Zero-Trust Architecture. Also used some SOCL/SOSL queries to store and retive.



```
    @AuraEnabled
    public static Map<String, String> encryptAndSave(String fileName, String plainText, String email) {
        Blob key = Crypto.generateAesKey(128);
        Blob data = Blob.valueOf(plainText);
        Blob encrypted = Crypto.encryptWithManagedIV('AES128', key, data);

        String base64Key = EncodingUtil.base64Encode(key);
        String cipherText = EncodingUtil.base64Encode(encrypted);

        File_encrypt__c fileRec = new File_encrypt__c();
        fileRec.Name = fileName;
        fileRec.Email_id__c = email;
        fileRec.Encrypted_Content__c = cipherText;
        insert fileRec;

        // Send encryption key to user
        Messaging.SingleEmailMessage mail = new Messaging.SingleEmailMessage();
        mail.setToAddresses(new String[] { email });
        mail.setSubject('Your Encryption Key');
        mail.setPlainTextBody(
            'Your encryption key for file "' + fileName + '" is: ' + base64Key +
            '\n\n⚠️ Keep this safe. If you lose it, your data cannot be recovered.'
        );
        Messaging.sendEmail(new Messaging.SingleEmailMessage[] { mail });

        Map<String, String> result = new Map<String, String>();
        result.put('generatedKey', base64Key);
        result.put('cipherText', cipherText);
        return result;
    }
```

This will decrypt that data based on the Key given as input by the user and data selected to decrypt. This is also the Core Function 2 THE DECRYPTION



```
    @AuraEnabled
    public static String decryptFile(Id fileId, String base64Key) {
        File_encrypt__c fileRec = [
            SELECT Encrypted_Content__c
            FROM File_encrypt__c
            WHERE Id = :fileId
            LIMIT 1
        ];
        Blob key = EncodingUtil.base64Decode(base64Key);
        Blob encrypted = EncodingUtil.base64Decode(fileRec.Encrypted_Content__c);
        Blob decrypted = Crypto.decryptWithManagedIV('AES128', key, encrypted);
        return decrypted.toString();
    }

    // 🔹 Get files for a user
    @AuraEnabled
    public static List<File_encrypt__c> getFiles(String email) {
        return [
            SELECT Id, Name
            FROM File_encrypt__c
            WHERE Email_id__c = :email
        ];
    }
```

The Delete file option for User

```
// ◆ Delete file
@AuraEnabled
public static Boolean deleteFile(Id fileId) {
    delete [SELECT Id FROM File_encrypt__c WHERE Id = :fileId LIMIT 1];
    return true;
}
```

send otp code

```
// ◆ Send OTP
@AuraEnabled
public static Boolean sendOtp(String email) {
    Integer raw = Math.abs(Crypto.getRandomInteger());
    Integer modVal = Math.mod(raw, 1000000);
    String otp = String.valueOf(modVal);
    while (otp.length() < 6) {
        otp = '0' + otp;
    }

    cry_Customer__c cust = [
        SELECT Id FROM cry_Customer__c WHERE Email__c = :email LIMIT 1
    ];

    cust.OTP__c = otp;
    cust.OTP_Expiry__c = System.now().addMinutes(5);
    update cust;

    Messaging.SingleEmailMessage mail = new Messaging.SingleEmailMessage();
    mail.setToAddresses(new String[] { email });
    mail.setSubject('Your One-Time Password');
    mail.setPlainTextBody('Your OTP is: ' + otp + ' (valid for 5 minutes)');
    Messaging.sendEmail(new Messaging.SingleEmailMessage[] { mail });

    return true;
}
```

Finally the verication of OTP code

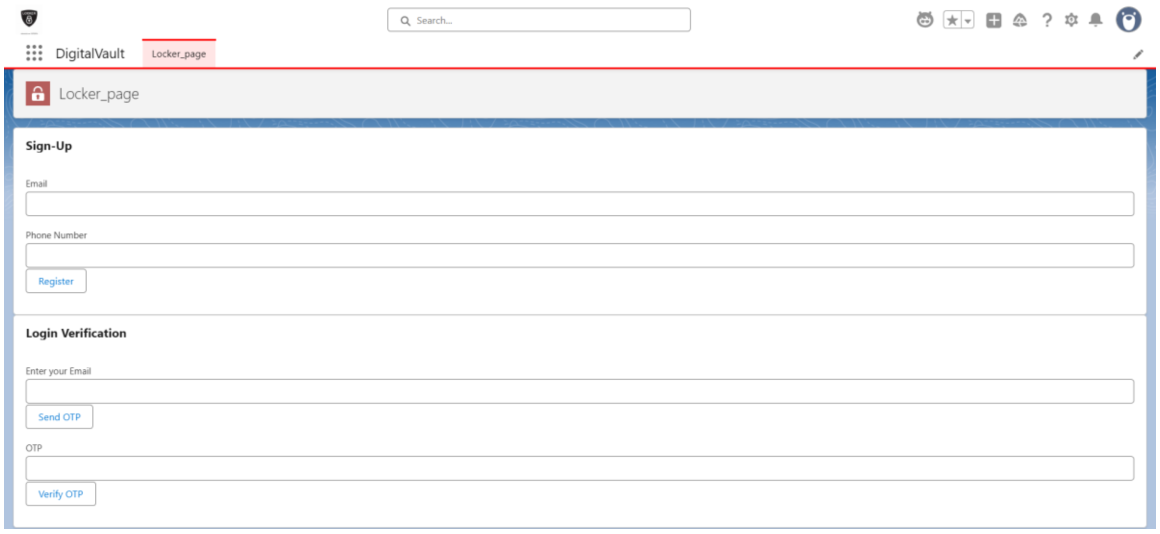
```
    // 🔹 Verify OTP
    @AuraEnabled
    public static Boolean verifyOtp(String email, String otp) {
        cry_Customer__c cust = [
            SELECT Id, OTP__c, OTP_Expiry__c
            FROM cry_Customer__c
            WHERE Email__c = :email
            LIMIT 1
        ];
        if (cust != null && cust.OTP__c == otp && cust.OTP_Expiry__c > System.now()) {
            return true;
        }
        return false;
    }
}
```

Phase 6: User Interface Development

- Lightning Web Component → `fileEncryptor`.
- Features:
  - Sign-Up form.
  - OTP login verification.
  - File list display.
  - Encrypt & Save new file.
  - Decrypt & Delete existing file.

- **Conditional Rendering** → Sections visible only after OTP verification.

The Frontend for user looks like



Locker_page

Sign-Up

Email

Phone Number

Register

Login Verification

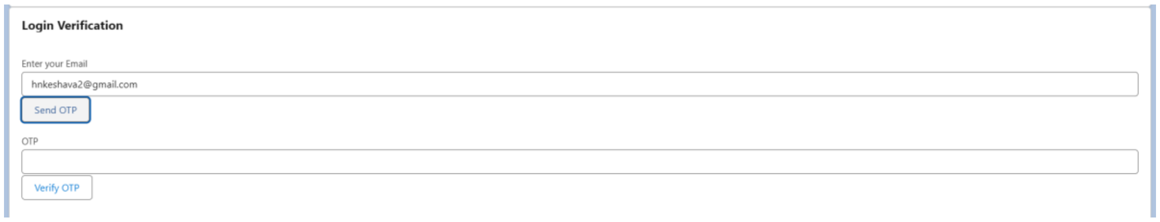
Enter your Email

Send OTP

OTP

Verify OTP

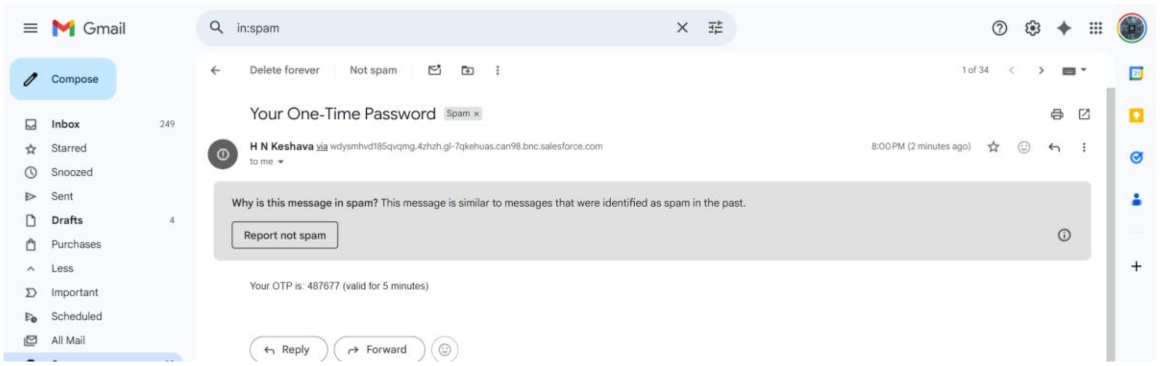
Suppose the users is already there then user enters the gmail and the verifies it using the OTP sent to the user's gmail sent



Login Verification

Enter your Email

hnkeshava2@gmail.com



Your One-Time Password Spam ×

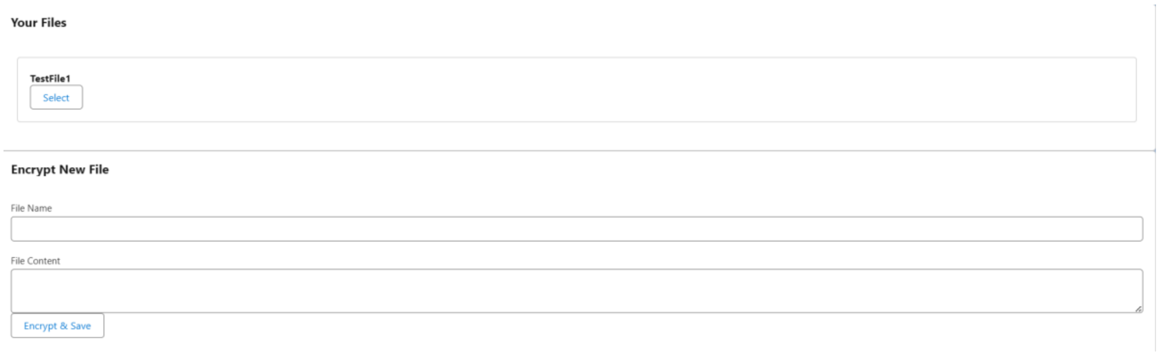
H N Keshava via wdysmhwd185qvqmg.4zhzh.gl-7qkehuas.can98.bnc.salesforce.com
to me
8:00 PM (2 minutes ago)

Why is this message in spam? This message is similar to messages that were identified as spam in the past.

Report not spam

Your OTP is: 487677 (valid for 5 minutes)

Then the user's files list will be there which can be decrypted and if the user want to encrypt a new file and save it then user can enter it in the "Encrypt New File" section.

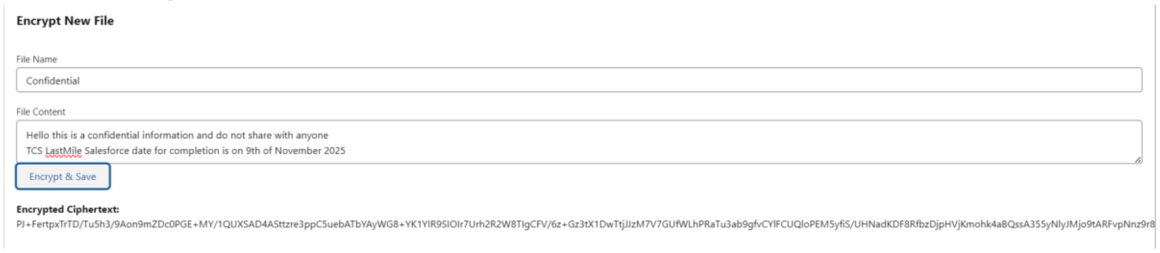


Your Files

TestFile1

Encrypt New File

Like the example user enter the file name that he/she will like it and then the key is sent to the user's gmail



Encrypt New File

File Name

Confidential

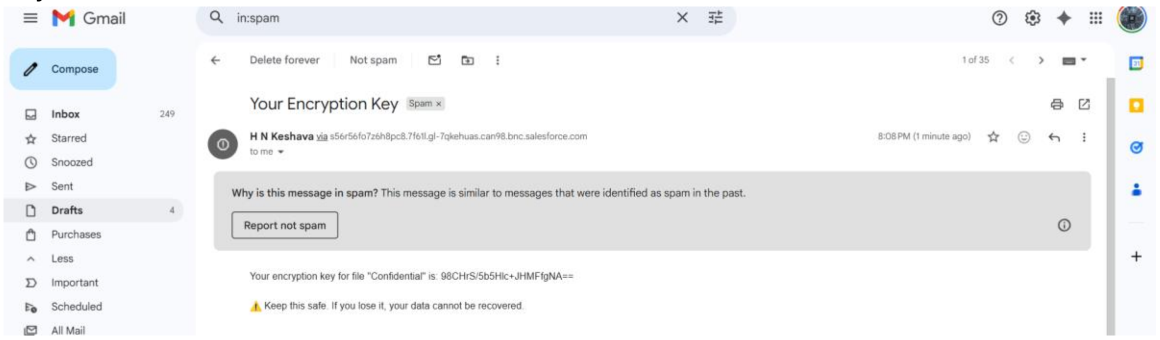
File Content

Hello this is a confidential information and do not share with anyone
TCS LastMile Salesforce date for completion is on 9th of November 2025

Encrypted Ciphertext:

PJ+FertpxTrTD/Tu5h3/9Aon9mZDc0PGE+MY/1QUXSAD4ASttzre3ppC5uebATbYAyWG8+YK1YIR9SIOIr7Urh2R2W8TIgCFV/6z+Gz3tX1DwTtjJJzM7V7GUfWLhPRaTu3ab9gfvCYlFCUQloPEM5yfiS/UHNadKDF8RfbzDjpHVjKmohk4aBQssA355yNlyJMjo9tARFvpNnz9r8

Key sent to the user’s email



Your Encryption Key Spam ×

H N Keshava via s56r56fo7z6h8pc8.7f61l.gl-7qkehuas.can98.bnc.salesforce.com
to me

8:08 PM (1 minute ago)

Why is this message in spam? This message is similar to messages that were identified as spam in the past.

Your encryption key for file "Confidential" is: 98CHrS/5b5Hlc+JHMFfgNA==

⚠️ Keep this safe. If you lose it, your data cannot be recovered.

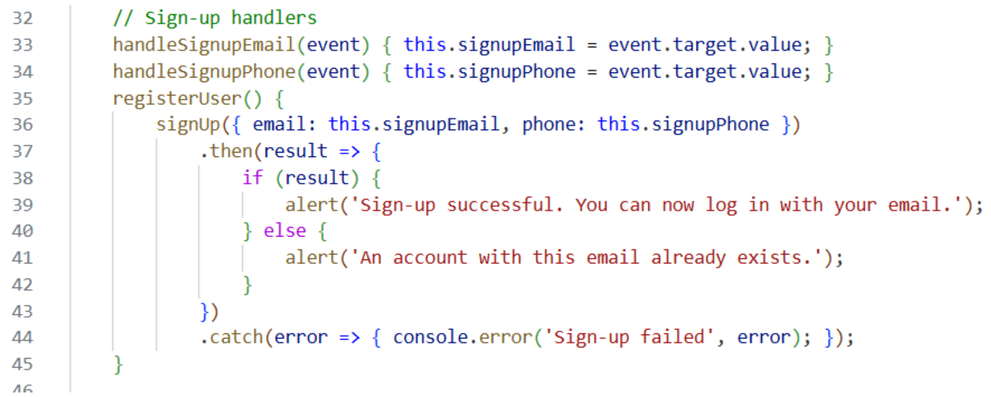
Phase 7: Integration & External Access

- **Email Service** → Used for OTP and encryption key delivery.

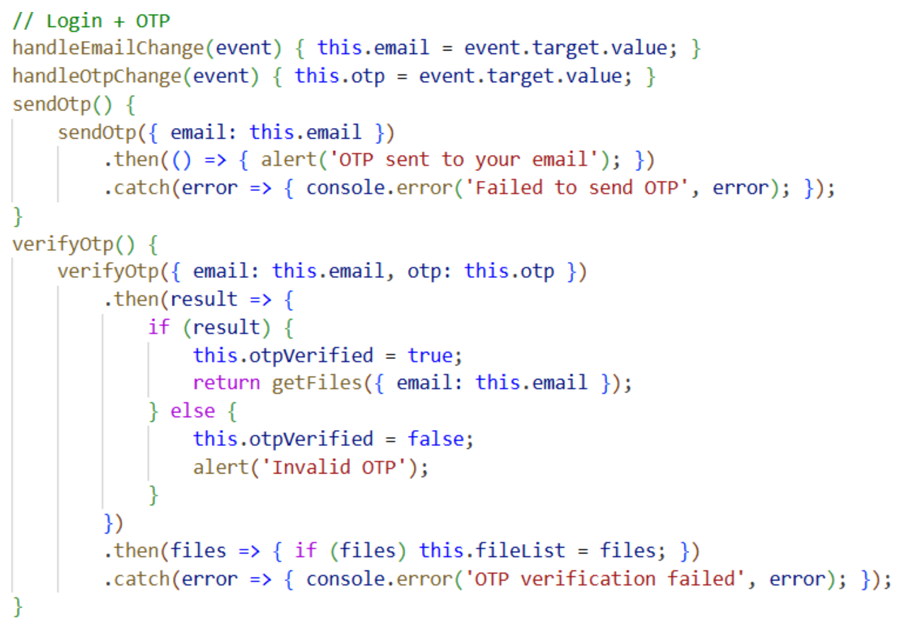
Phase 8: Data Management & Deployment

- **VS Code + SFDX** → Used for deployment and rollback (show in the previous section).
- **Rollback Plan** → Always maintained a known-good baseline.

This will handle the errors and rollbacks

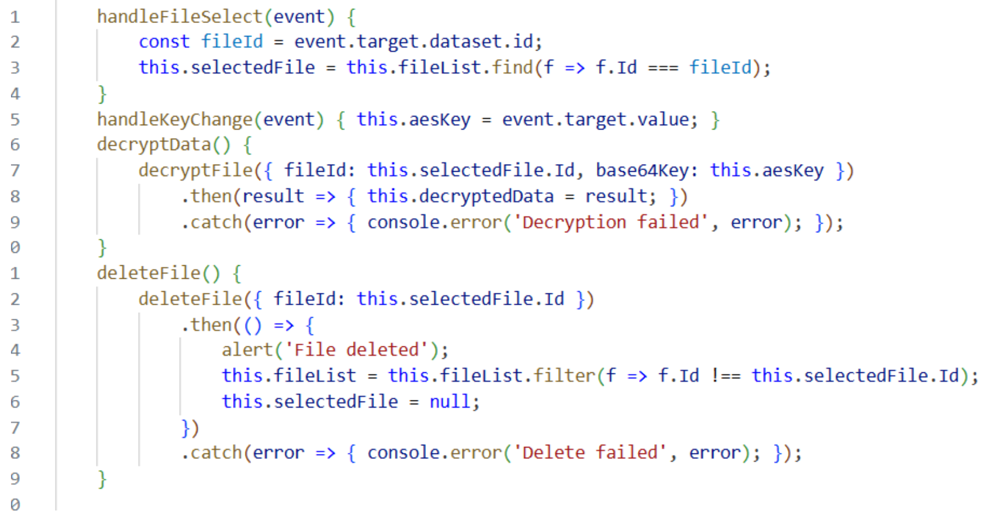


```
// Sign-up handlers
handleSignupEmail(event) { this.signupEmail = event.target.value; }
handleSignupPhone(event) { this.signupPhone = event.target.value; }
registerUser() {
    signUp({ email: this.signupEmail, phone: this.signupPhone })
    .then(result => {
        if (result) {
            alert('Sign-up successful. You can now log in with your email.');
        } else {
            alert('An account with this email already exists.');
        }
    })
    .catch(error => { console.error('Sign-up failed', error); });
}
```



```
// Login + OTP
handleEmailChange(event) { this.email = event.target.value; }
handleOtpChange(event) { this.otp = event.target.value; }
sendOtp() {
    sendOtp({ email: this.email })
        .then(() => { alert('OTP sent to your email'); })
        .catch(error => { console.error('Failed to send OTP', error); });
}
verifyOtp() {
    verifyOtp({ email: this.email, otp: this.otp })
        .then(result => {
            if (result) {
                this.otpVerified = true;
                return getFiles({ email: this.email });
            } else {
                this.otpVerified = false;
                alert('Invalid OTP');
            }
        })
        .then(files => { if (files) this.fileList = files; })
        .catch(error => { console.error('OTP verification failed', error); });
}
```

Encryption/Decryption Key handling

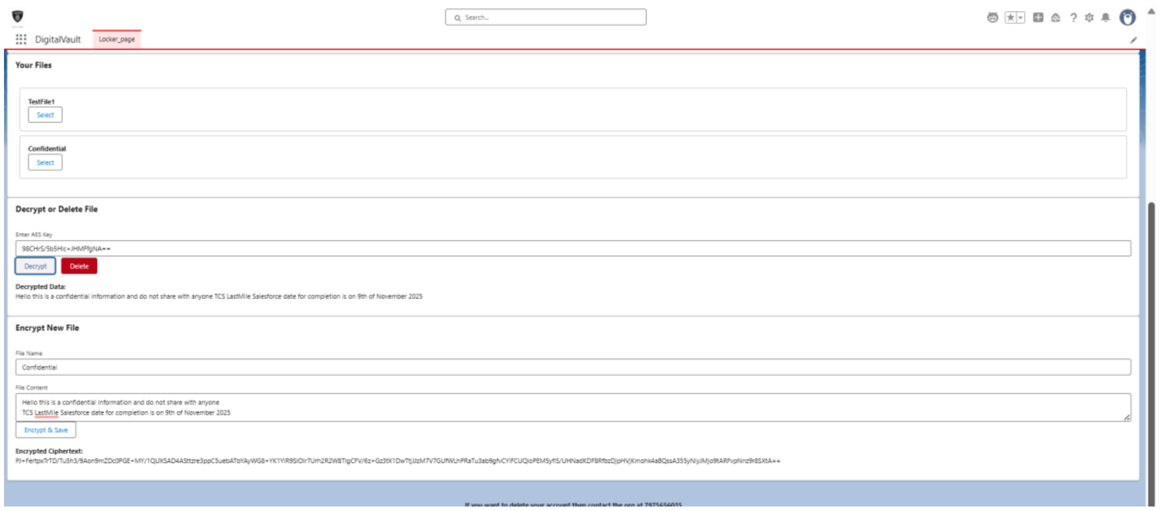


```
handleFileSelect(event) {
    const fileId = event.target.dataset.id;
    this.selectedFile = this.fileList.find(f => f.Id === fileId);
}
handleKeyChange(event) { this.aesKey = event.target.value; }
decryptData() {
    decryptFile({ fileId: this.selectedFile.Id, base64Key: this.aesKey })
        .then(result => { this.decryptedData = result; })
        .catch(error => { console.error('Decryption failed', error); });
}
deleteFile() {
    deleteFile({ fileId: this.selectedFile.Id })
        .then(() => {
            alert('File deleted');
            this.fileList = this.fileList.filter(f => f.Id !== this.selectedFile.Id);
            this.selectedFile = null;
        })
        .catch(error => { console.error('Delete failed', error); });
}
```

Phase 9: Reporting, Dashboards & Security Review

- **Sharing Settings** → Encrypted files private to owner .
- **Field Level Security** → Sensitive fields hidden from unauthorized users, no service/company can see the users data as the key is not stored.
- **Audit Trail** → Tracks changes to customer and file records.
- **Zero-Trust Review** → Confirmed no decrypted data stored in system.
- **Key losing**→ If the user loses the Key given through the mail then it is **Impossible** to recover the message.

  Complete User dashboard after login/sign up



Your Files

TestFile1

Confidential

Decrypt or Delete File

Enter AES Key: 98CHrS/5b5Hic+JHMFfgNA==

Decrypted Data:
Hello this is a confidential information and do not share with anyone TCS LastMile Salesforce date for completion is on 9th of November 2025

Encrypt New File

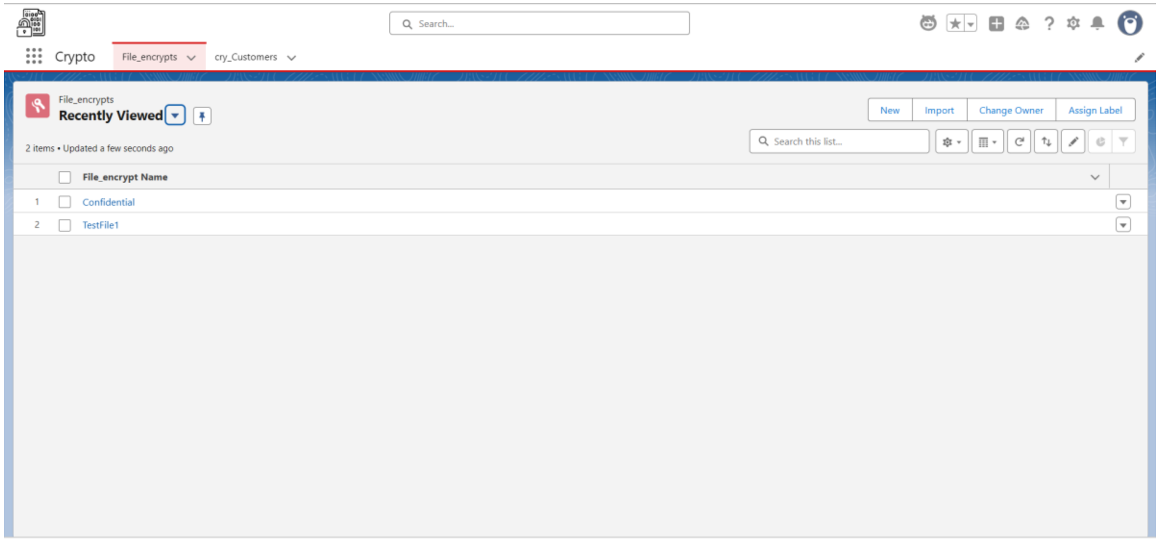
File Name: Confidential

File Content:
Hello this is a confidential information and do not share with anyone
TCS LastMile Salesforce date for completion is on 9th of November 2025

Encrypted Ciphertext:
PJ+FertpxTrTD/Tu5h3/9Aon9mZDc0PGE+MY/1QUXSAD4ASttzre3ppC5uebATbYAyWG8+YK1YIR9SIOir7Urh2R2WBTlgCFV/6z+Gz3tX1DwTtJJzM7V7GUfWLhPRaTu3ab9gfvCYIFCUQIoPEM5yfIS/UHNadKDF8RfbzDjpHVJKmohk4aBQssA355yNiyJMjo9tARFvpNnz9r8SXtA==

If you want to delete your account then contact the org at 7975656035

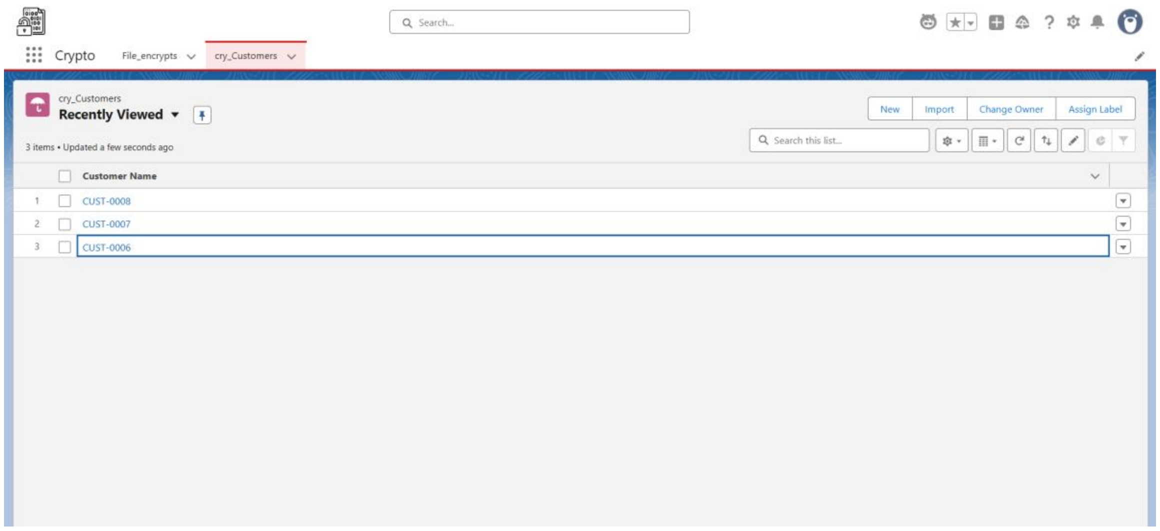
Company Dashboard number – 1 this has encrypted files of multiple users. Files created will be reported here.



2 items • Updated a few seconds ago

	File_encrypt Name
1	Confidential
2	TestFile1

Company Dashboard number – 2 this has All the users and their information in it. Users login or signup will be reported here.



cry_Customers
Recently Viewed
3 items • Updated a few seconds ago

	Customer Name
1	CUST-0008
2	CUST-0007
3	CUST-0006

Phase 10: Final Presentation & Demo Day

- **Demo Walkthrough** → Showed Sign-Up, OTP Login, Encrypt, Decrypt, Delete.
- **Zero-Trust Messaging** → Clear warnings in UI footer.
- **Documentation** → Phase mapping, screenshots, and explanation prepared.
- **Portfolio Showcase** → Ready for evaluator and internship submission.