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.
- Users set up for testing flows.
- Deployment managed through **VS Code + SFDX** with rollback plans.

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.
- **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.
    - 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.

Phase 5: Apex Programming (Developer)

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

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.

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.
- **Rollback Plan** → Always maintained a known-good baseline.

Phase 9: Reporting, Dashboards & Security Review

- **Sharing Settings** → Encrypted files private to owner.
- **Field Level Security** → Sensitive fields hidden from unauthorized users.
- **Audit Trail** → Tracks changes to customer and file records.
- **Zero-Trust Review** → Confirmed no decrypted data stored in system.

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.