# Protected Document Store (Credentials Store)

## Overview

1. This project is a document store. The document is to be stored in git service such as GitHub, BitBucket, GitLab in an encrypted form.
2. The project is envisioned as a Front-end application.
3. The application should have a log in screen. The login screen should be backed by the Git Service provider.
4. Post logging in to the application, the encryption phrase is set.
5. This phrase is used to encrypt/decrypt the text documents.
6. As the authentication is already completed, the encrypted documents in Git repository.
7. It is important to note each encryption/decryption happens on the browser. Unencrypted document does not leave the browser.

## Use Cases

1. Create a document.
2. View the list of document names available.
3. View the content of one document.
4. Update the content of one document.
5. Delete a document.
6. Search a document from the list based on the name.

## Supporting Use case

To meet the main use cases, the following supporting use cases are necessary.

1. Login to Git Service provider
2. Setting the encryption phrase

## Design decisions made

**Project Framework**: ReactJS, Create React App

**Code Repo**: GitHub : <https://github.com/nachiappannk/doc-store>

**Deployment**: Azure (Static WebApp): [React App (wonderful-desert-0c742fe00.3.azurestaticapps.net)](https://wonderful-desert-0c742fe00.3.azurestaticapps.net/)

**CICD pipelines**: GitHub actions

**Git Repo for storing documents**: GitLab

**Authentication Provider:** GitLab (GitLab though is not as popular as GitHub does provide PCKE authentication – not provided by Github)

## Work done

1. Project structure is created.
2. Authentication is implemented.
3. CICD pipeline is set up.
4. POC for encryption and decryption using crypto is done
5. Validated APIs of GitLab via postman/curl

## Work to be done

1. Improve the screens to pleasing UI.
2. Should support different resolution.
3. Implementing “Set Encryption Context screen”
4. Implementing “Listing all documents screen”
5. Implementing “Viewing a document”
6. Implementing “Update to a document”
7. Implementing a ReadMe page
8. Implementing Routing in react
9. Testing the product