**ShareSafely - Secure File Sharing Web Application**

**Overview**

ShareSafely is a web application that allows users to securely upload files to Azure Blob Storage. Once uploaded, the application generates a unique, time-limited link for secure sharing. This ensures that only authorized users with the link can access the uploaded file for a specified duration.

**Features**

* Secure file uploads to Azure Blob Storage.
* Automatic generation of unique, time-limited download links.
* Protection of sensitive credentials using Azure Key Vault.
* Hosted on Azure Web Apps for scalability and security.
* Monitoring and automatic cleanup of expired files.

**Technologies Used**

* **Backend**: Flask (Python)
* **Frontend**: HTML, CSS
* **Cloud Services**:
  + Azure Blob Storage
  + Azure Key Vault
  + Azure Web Apps

**Project Structure**

ShareSafely/

│── app.py # Flask backend application

│── templates/

│ ├── upload.html # HTML file upload form

│── static/

│── requirements.txt # Required dependencies

│── README.md # Project documentation

**Setup Instructions**

**Prerequisites**

* Python 3.8+
* Azure account
* Azure Storage Account & Container
* Azure Key Vault (for secure credential storage)

**Step 1: Clone the Repository**

git clone <repository-url>

cd ShareSafely

**Step 2: Install Dependencies**

pip install -r requirements.txt

**Step 3: Configure Azure Blob Storage**

* Create an Azure Storage Account.
* Create a container named files.
* Store your connection string securely in Azure Key Vault.

**Step 4: Set Environment Variables**

export AZURE\_STORAGE\_CONNECTION\_STRING="your-connection-string"

export AZURE\_KEY\_VAULT\_NAME="your-keyvault-name"

**Step 5: Run the Flask Application**

python app.py

The application will be available at http://127.0.0.1:5000.

**Deployment to Azure Web Apps**

1. **Create an Azure Web App**:
2. az webapp create --name ShareSafelyApp --resource-group <resource-group> --plan <app-service-plan>
3. **Deploy using Git or Azure CLI**:
4. az webapp up --name ShareSafelyApp --resource-group <resource-group>

**Usage**

1. Open the web app.
2. Upload a file using the provided form.
3. Receive a secure, time-limited link for sharing.

**Security Best Practices**

* Use Azure Key Vault to store credentials.
* Set storage container access to private.
* Enable logging and monitoring for tracking activities.

**License**

This project is licensed under the MIT License.

**Author**

Tafadzwa Mwerenga