Blockchain Certificate Verification System:

This project implements a blockchain-based certificate verification system with two roles: -

1.) Government officials can upload certificates.
2.) Users can verify certificates using either the certificate ID or uploaded documents.

Prerequisites:

Before you begin, ensure that you have the following installed on your machine: -

Python 3.x (You can download it from: https://www.python.org/downloads/)
Pip (Python's package manager, which comes with Python 3.x)
Postman (Optional for testing APIs) - [Download Postman: https://www.postman.com/downloads/]

Setup Instructions:

1.) Clone the Repository Start by cloning this repository to your local machine:
```bash
git clone https://github.com/your-username/blockchain-certificate-verification.git cd blockchain-certificate-verification
```

2.) Open cmd, and enter the following commands:

    1.) cd project\address\on\your\local\machine

        eg. cd C:\user\username\Documents\Programs\Blockchain verification system

    2.) python -m venv venv

    3.) venv\Scripts\activate (if windows)
        source venv/bin/activate (if mac/linux)

4.) pip install -r requirements.txt

5.) python app.py

6.) The application should now be running at http://127.0.0.1:5000

Test the Application:

You can now access the application through your browser by visiting:

1.) **For certificate upload (Government officials)**: http://127.0.0.1:5000/official
2.) **For certificate verification (Users)**: http://127.0.0.1:5000/user

Posting Through Postman:

1.) POST /official (for uploading certificates):

Headers:

Role: government

Form data:

certificate_id, department, issuer, pdf_file

2.) POST /verify_certificate (for verifying certificates):

Form data:

certificate_id or pdf_file

Stopping the Server:

To stop the Flask server, simply press Ctrl+C in the terminal where it's running.

Directory Structure:

1.) app.py: The main Flask application with routes for uploading and verifying certificates.

2.) blockchain.py: Contains the blockchain implementation to store and retrieve certificates.

3.) templates/: Contains HTML templates for the user and official pages.

4.) static/: Contains CSS and JavaScript files for the frontend.