**What You Need to Change for Real-Time (Production) Deployment**

1. **Blockchain Network**

* **Change:**
* Update your Flask app to use a mainnet (e.g., Ethereum Mainnet) or a public testnet (e.g., Sepolia, Goerli) endpoint from Infura/Alchemy instead of Ganache or localhost.
* **How:**
* Change the Web3 provider URL in your code from http://127.0.0.1:7545 (Ganache) to your Infura/Alchemy endpoint.

1. **Smart Contract Deployment**

* **Change:**
* Deploy your smart contracts to the mainnet or testnet, not to Ganache.
* **How:**
* Use Truffle/Hardhat with the correct network configuration.

1. **API Keys and Secrets**

* **Change:**
* Use production API keys for Infura/Alchemy, Firebase, and Pinata.
* Store them securely (environment variables, not hardcoded).

1. **File Storage**

* **Change:**
* Use production Pinata or Infura IPFS accounts if you need more storage or reliability.

1. **App Hosting**

* **Change:**
* Deploy your Flask app to a public server or cloud platform (Render, Railway, Fly.io, AWS, etc.).
* Use a production-ready server (Gunicorn, Nginx, etc.).

1. **Domain and SSL**

* **Change:**
* Use a real domain and enable HTTPS (SSL certificate).

**What You Do NOT Need to Change**

* Your core application logic (Flask routes, certificate generation, QR code logic, etc.) can remain the same.
* Your database structure and Firebase logic (just update credentials if needed).
* Your IPFS/Pinata integration logic (just update API keys if needed).

**In summary:**

* Only update network endpoints, API keys, and deployment targets for production.
* Do NOT change your core app logic unless you want to add new features or fix bugs.

If you want a checklist or code snippet for any of these changes, let me know!

Cmd to setup : secret\_key

Remove-Item -Recurse -Force node\_modules

Remove-Item -Force package-lock.json

npm install

npm install @truffle/hdwallet-provider

truffle migrate –reset

After i*nstalling Dependencies in requirements.txt*

*Use cmd to run Application: python flask\_app.py*

*Use Remix to get smart contract by uploading solidity files*

*Install Ganache to test this in local……. By starting ganache server*

*Use RPC server address*