National College of Ireland

Configuration Manual

MSc Research Project
M.Sc. Cyber Security

Shreyas Joshi
Student ID: 23270071

School of Computing
National College of Ireland

Supervisor: Zakaria Sabir

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Shreyas Sharad Joshi
Student ID:	23270071
Programme:	M.Sc. Cyber Security **Year:** 2025
Module:	Practicum
Lecturer:	Zakaria Sabir
Submission Due Date:	11-08-2025
Project Title:	A Blockchain-Based dApp for Secure Communication and Prevention of Phishing Attacks
Word Count:	1045 **Page Count:** 13

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Shreyas Sharad Joshi
Date:	11-08-2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Shreyas Joshi
Student ID: 23270071

1 Introduction

This Configuration Manual gives a detailed, procedural installation, configuration, and operational methods of the blockchain-based secure files sharing platform designed in a bid to counteract the AI-enhanced phishing attacks. The guide is aimed towards a technical and also non-technical audience.

Project Overview
One is able to do the following on this dApp:
Carry out secure transmission of files and messages by using blockchain
Convenient sporadic encrypting of sensitive material using AES encryption
IPFS (Pinata) storage of files Non-removal
Source proof-of-work (block chain as source hash-checks of the material)
Sender and receiver can only share the hash and it does not allow phishing, smishing or deepfake impersonation attacks.

2 System Requirements and Technologies

2.1 System Requirements

Component	Requirement
Operating System	Windows 10/11, macOS, Linux
Node.js + npm	Node.js v18+, npm v9+
Blockchain Tools	Truffle v5.8+, Ganache CLI (latest)
Wallet	MetaMask (browser extension)
Storage Service	Pinata (with API key)
Text Editor	VS Code / Notepad++
Browser	Chrome / Firefox

2.2 Technology Stack

Layer	Technology
Frontend	React.js
Styling	Standard CSS
Smart Contract	Solidity (Via Truffle)
Blockchain	Ganache CLI (Local)
Storage	IPFS via Pinata
Encryption	AES
Backend	Node.js + Express
Wallet	Metamask

3 Project Setup and Configuration

3.1 Folder Structure

```
SecureBlockchain-dApp/
    backend/           # Express server for IPFS and encryption
    contracts/         # Solidity smart contracts
    frontend/          # React frontend
    migrations/        # Truffle migration scripts
    build/contracts/   # ABI and deployed contract artifacts
    .gitignore         # Files/folders to be ignored by Git
    package.json       # Project metadata
    truffle-config.js  # Truffle configuration
```

3.2 Installation Guide

3.2.1 Install Node.js and npm

- Download and install the LTS version from: https://nodejs.org
- Confirm installation:
    1. node -v
    2. npm -v

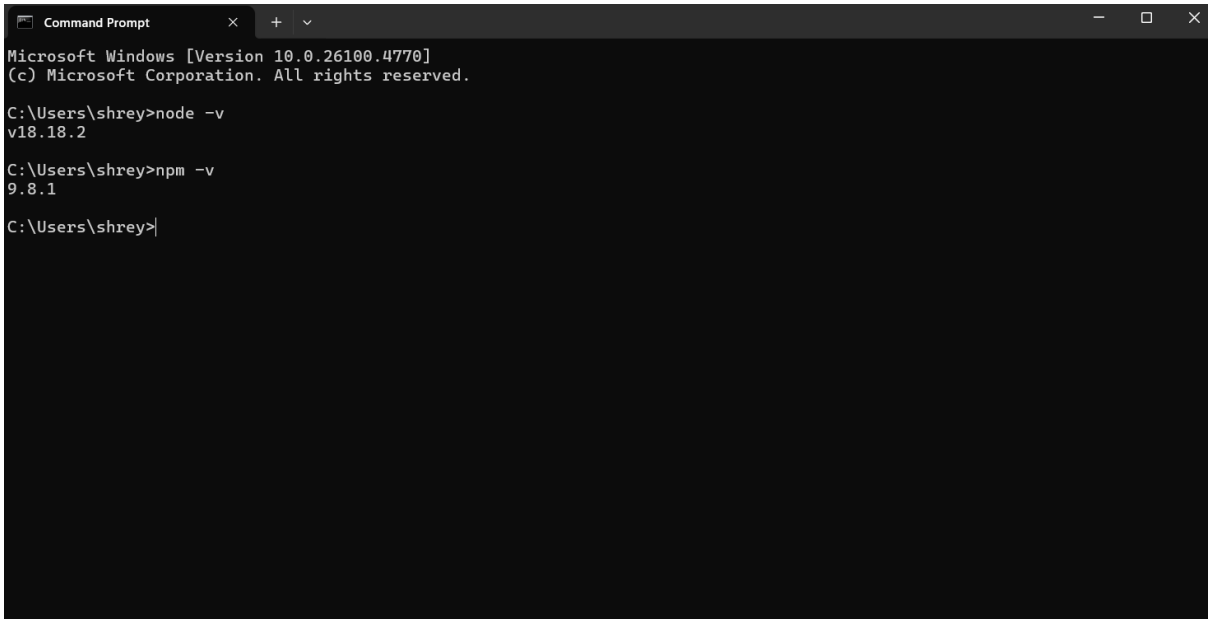


Fig : 1

3.2.2 Install Truffle (Solidity Framework)

- npm install -g truffle

Check version:

- truffle version

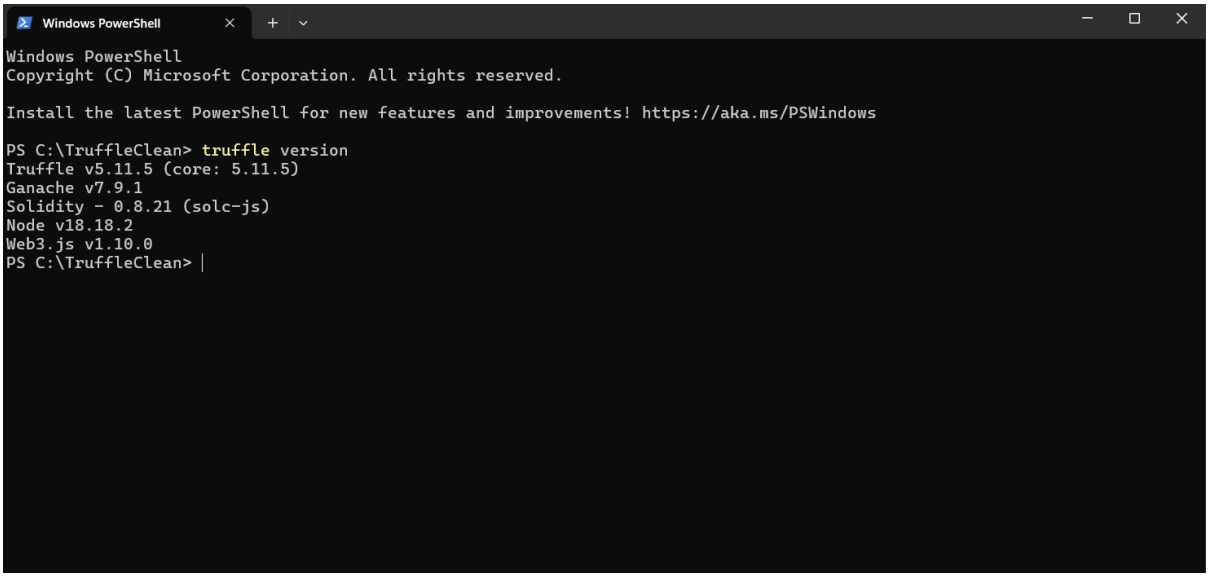


Fig :2

3.2.3 Install Ganache CLI (Blockchain Emulator)

- npm install -g ganache

3.2.4 Start Ganache

To launch Ganache locally and start a blockchain node with a custom chain and network ID:

npx ganache --chain.chainId 1338 --chain.networkId 1338

It will start to enlist the default Ethereum blockchain at http://127.0.0.1:8545, and indicates 10 test accounts with their individual private keys and balances. And keep such native terminal up.

Here, in bootstrap, this will trigger the local Ethereum chain at that moment on, `http://127.0.0.1:8545` and will list out 10 test accounts, their keys and balances. And keep such native terminal up.

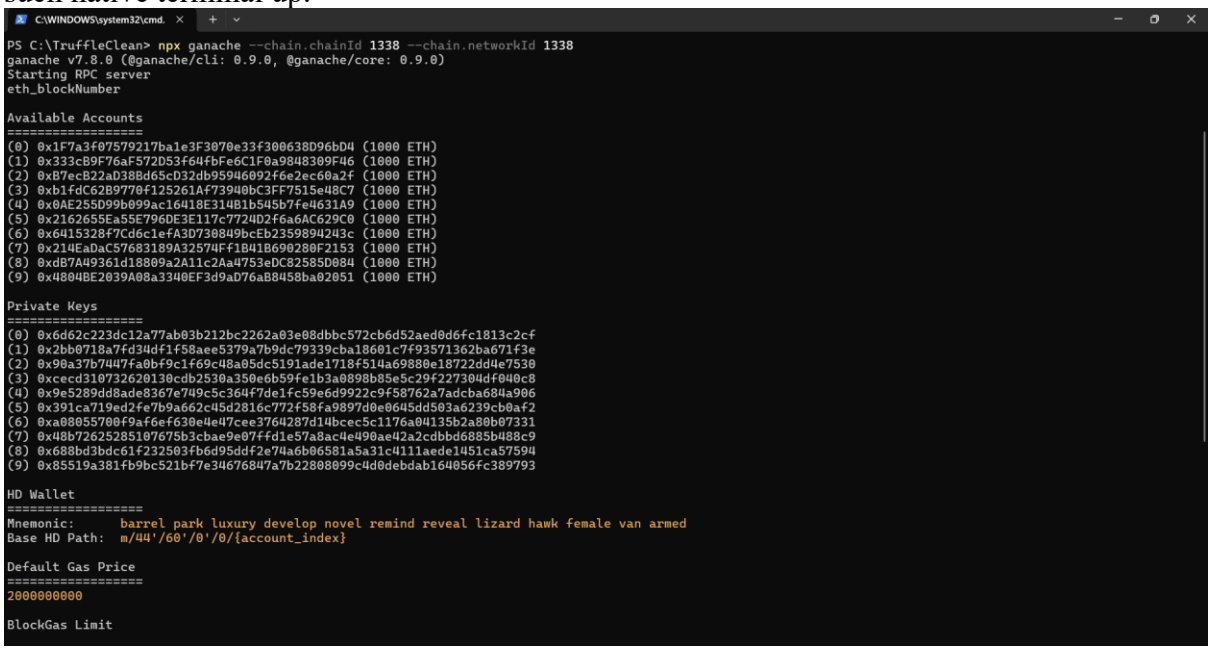


Fig :3

3.5 Install MetaMask (Wallet)

- Download MetaMask: https://metamask.io
- Create a new wallet and save your recovery phrase securely

3.4 Create a Custom Network in MetaMask for Ganache

1. Open MetaMask > Settings > Networks > Add Network
2. Enter:
    - Network Name: Localhost 8545
    - New RPC URL: http://127.0.0.1:8545
    - Chain ID: 1337 (or leave default)

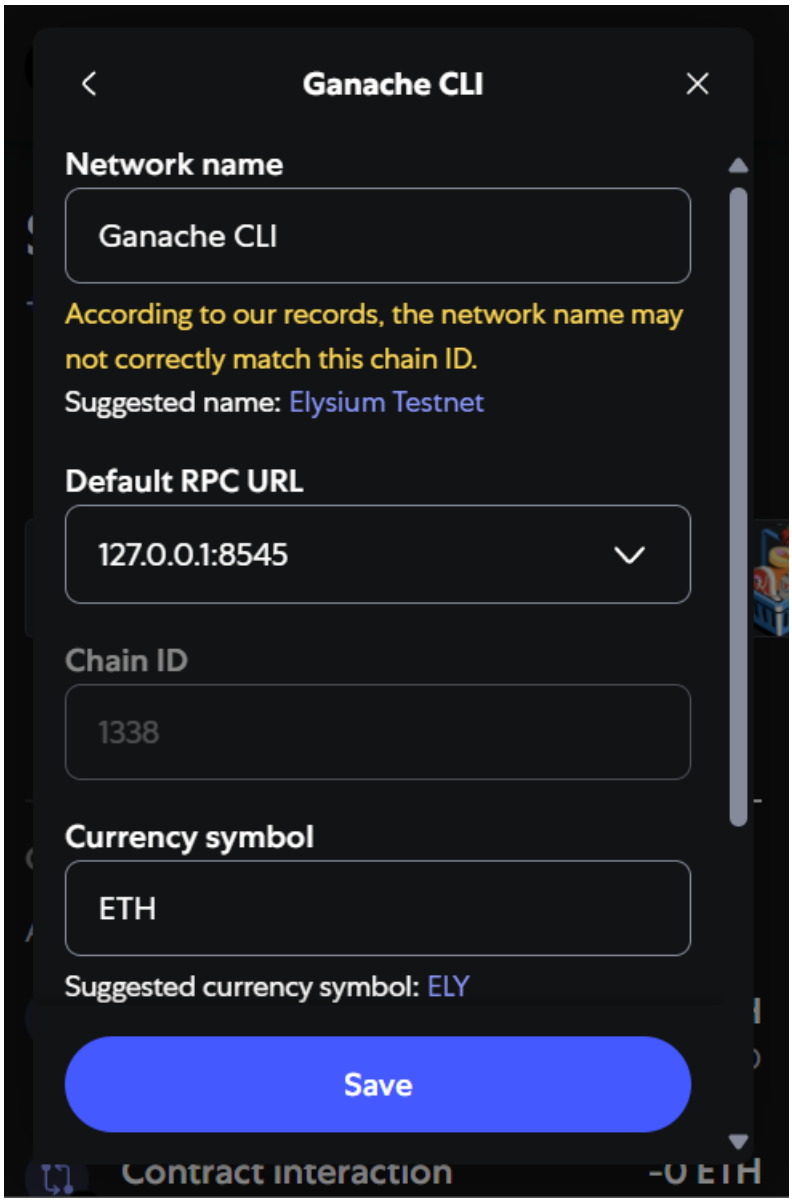


Fig :4

- Save

3. Import a private key from Ganache into MetaMask to use test ETH

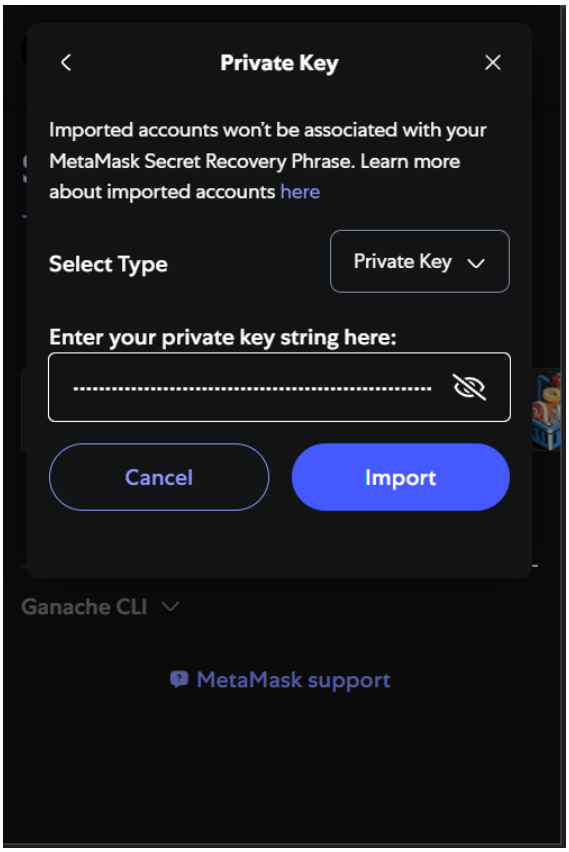

Private Key
Imported accounts won't be associated with your MetaMask Secret Recovery Phrase. Learn more about imported accounts here
Select Type
Private Key
Enter your private key string here:
Cancel
Import
Ganache CLI
MetaMask support

Fig :5

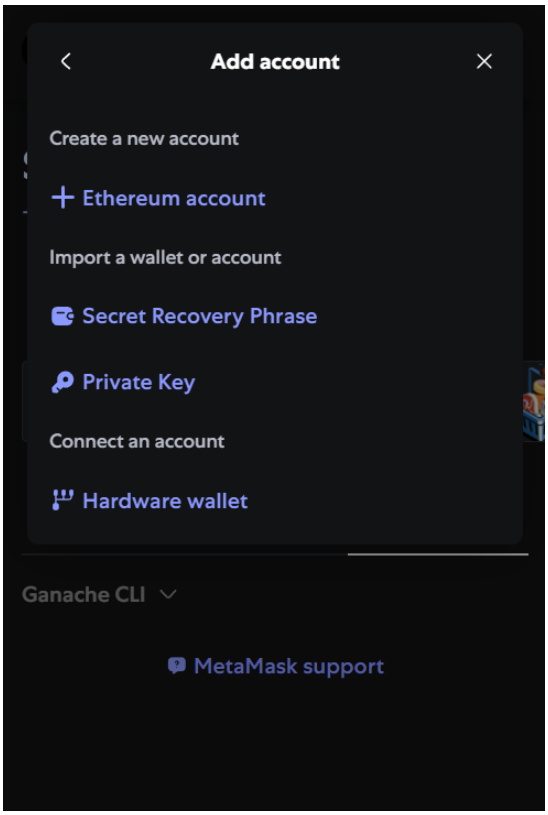

Add account
Create a new account
Ethereum account
Import a wallet or account
Secret Recovery Phrase
Private Key
Connect an account
Hardware wallet
Ganache CLI
MetaMask support

Fig :6

3.6 Create a Pinata Account (for IPFS)

4. Go to: https://app.pinata.cloud
5. Register/login
6. Navigate to API Keys
7. Create a new key and copy:
    - API Key
    - API Secret

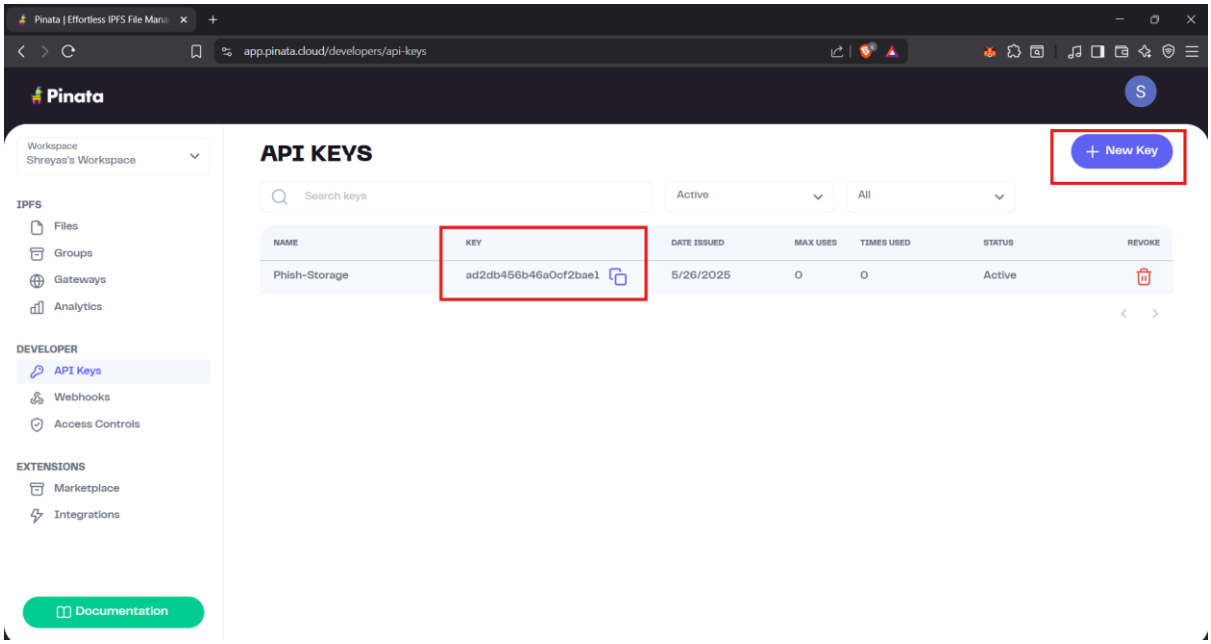


Fig :7

3.7 Install Express and Backend Dependencies

- cd backend
- npm install express cors dotenv axios

This sets up the Express server for encryption and IPFS integration.

3.8 Install Frontend Dependencies

- cd ../frontend
- npm install

Ensure dependencies for React are installed.

4. Configuration Steps

4.1 Backend Setup

1. cd backend
2. npm install
3. Create .env file:
    - PINATA_API_KEY=your_key
    - PINATA_SECRET_API_KEY=your_secret

4.2 Frontend Setup

- cd ../frontend
- npm install

4 Smart Contract Deployment

8. Smart Contract Deployment

8.1 Compile & Deploy Contracts

- cd ../
- truffle compile

```
PS C:\TruffleClean> truffle compile

Compiling your contracts...
===========================
> Compiling .\contracts\Migrations.sol
> Compiling .\contracts\SecureMessage.sol
> Artifacts written to C:\TruffleClean\build\contracts
> Compiled successfully using:
   - solc: 0.8.21+commit.d9974bed.Emscripten.clang
PS C:\TruffleClean>
```

Fig :8

- truffle migrate --network development

```
PS C:\TruffleClean> truffle migrate --network development

Compiling your contracts...
===========================
> Compiling .\contracts\Migrations.sol
> Compiling .\contracts\SecureMessage.sol
> Artifacts written to C:\TruffleClean\build\contracts
> Compiled successfully using:
   - solc: 0.8.21+commit.d9974bed.Emscripten.clang


Starting migrations...
======================
> Network name:    'development'
> Network id:      1338
> Block gas limit: 30000000 (0x1c9c380)


1_initial_migration.js
======================

   Replacing 'Migrations'
   ----------------------
   > transaction hash:    0x195baf9533bfc46fefde24619133822c5c416d973db7eb70d0e81a0dabb88796
   > Blocks: 0            Seconds: 0
   > contract address:    0xB5dCA95d2Ec6D94c7D871bF3C1f5Ce87AEBEdaA5
   > block number:        1
   > block timestamp:     1754782702
   > account:             0x1F7a3f07579217ba1e3F3070e33f300638D96bD4
   > balance:             999.999171501625
   > gas used:            245481 (0x3bee9)
   > gas price:           3.375 gwei
   > value sent:          0 ETH
   > total cost:          0.000828498375 ETH

   > Saving migration to chain.
   > Saving artifacts
   -------------------------------------
   > Total cost:      0.000828498375 ETH
```

Fig :9

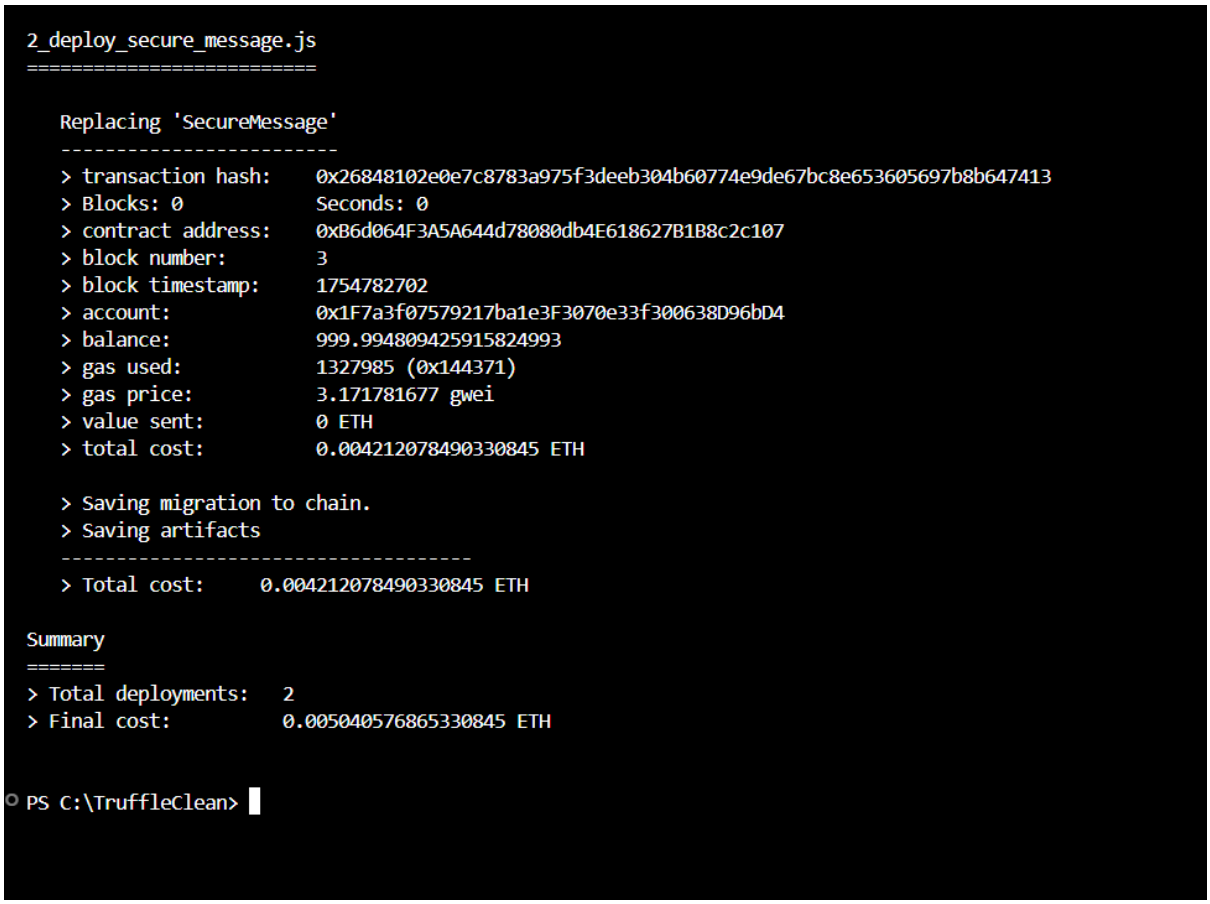


Fig :10

Make sure Ganache is running locally before deploying.

8.2 Copy ABI and Contract Address

After deployment, Truffle generates ABI JSON:

- Path: build/contracts/SecureMessage.json
- Copy the entire ABI object and deployed contract address
- Update the frontend file where ABI and contract address are imported (usually frontend/src/config/contract.js)

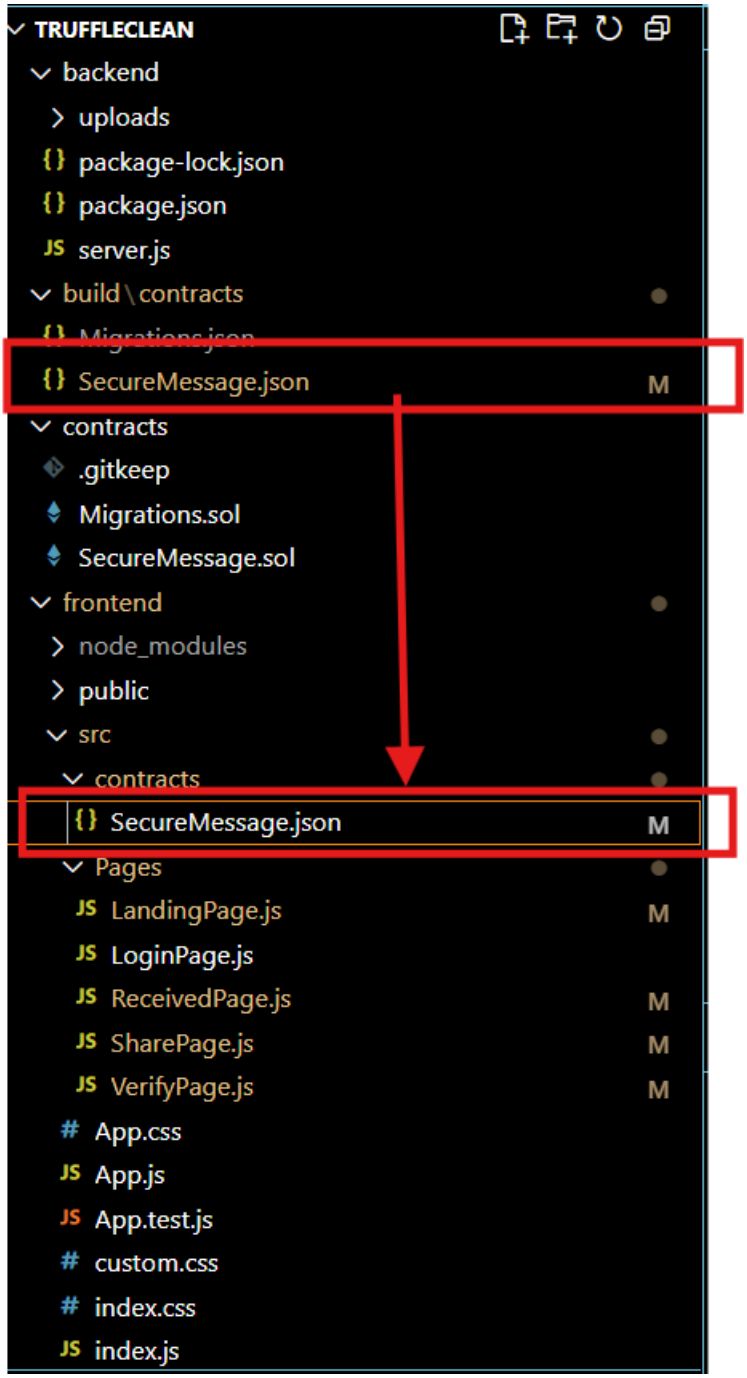


Fig :11

5 Running the Application

5.1. Running Backend Server

To start the backend Express server:

- cd backend
- node index.js

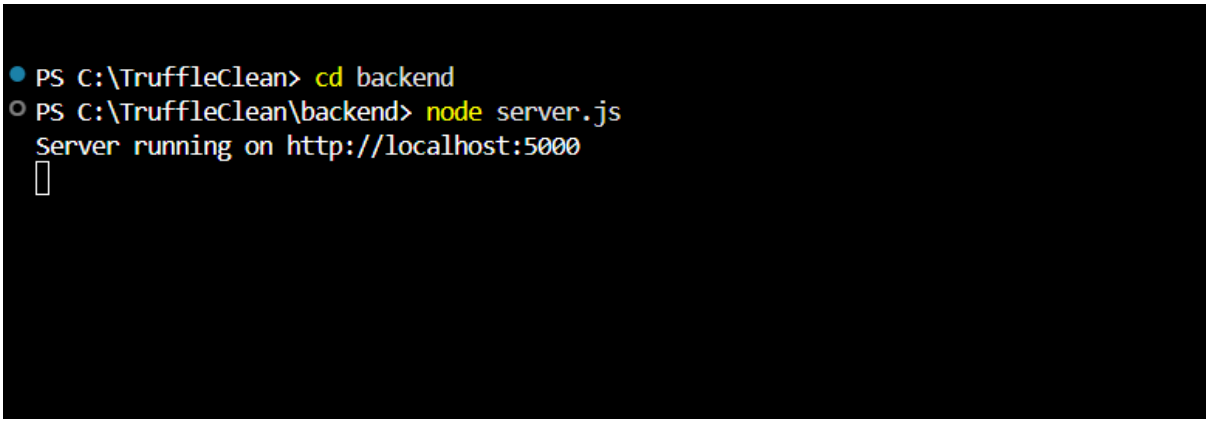


Fig :12

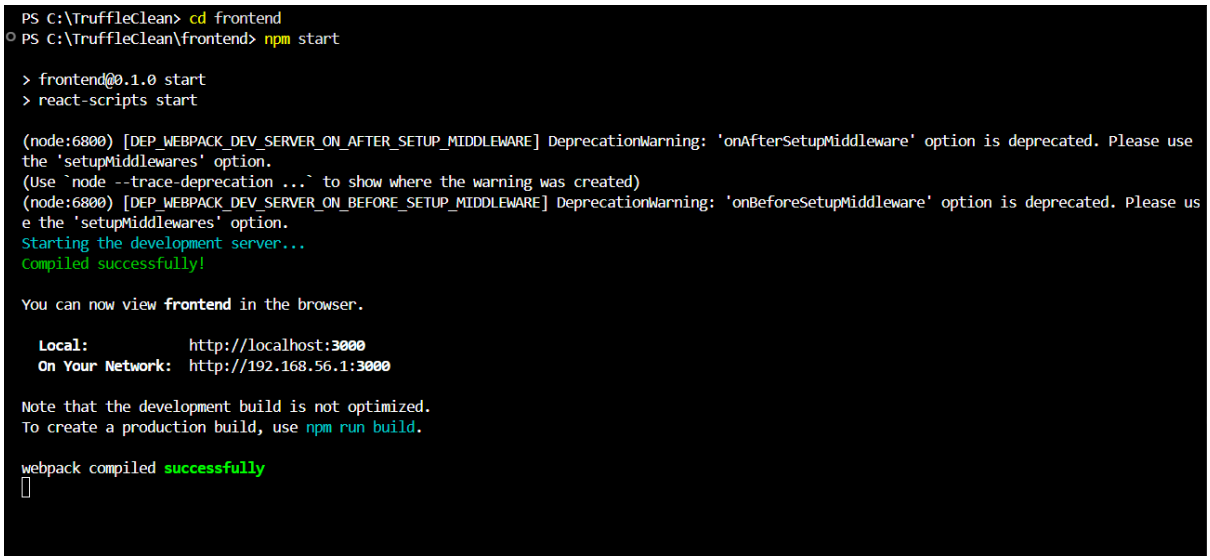
If your entry point is named something else (like server.js), use:

- node server.js
- npm start

This server:
- Encrypts uploaded files (if selected)
- Pins encrypted file to IPFS using Pinata
- Returns IPFS hash to be stored on blockchain

5.2. Running Frontend Application

- ```bash
- cd ../frontend



- npm start

Fig :13

1. Open browser at: http://localhost:3000
2. Connect MetaMask (set to Localhost 8545)
3. Upload the ABI and contract address

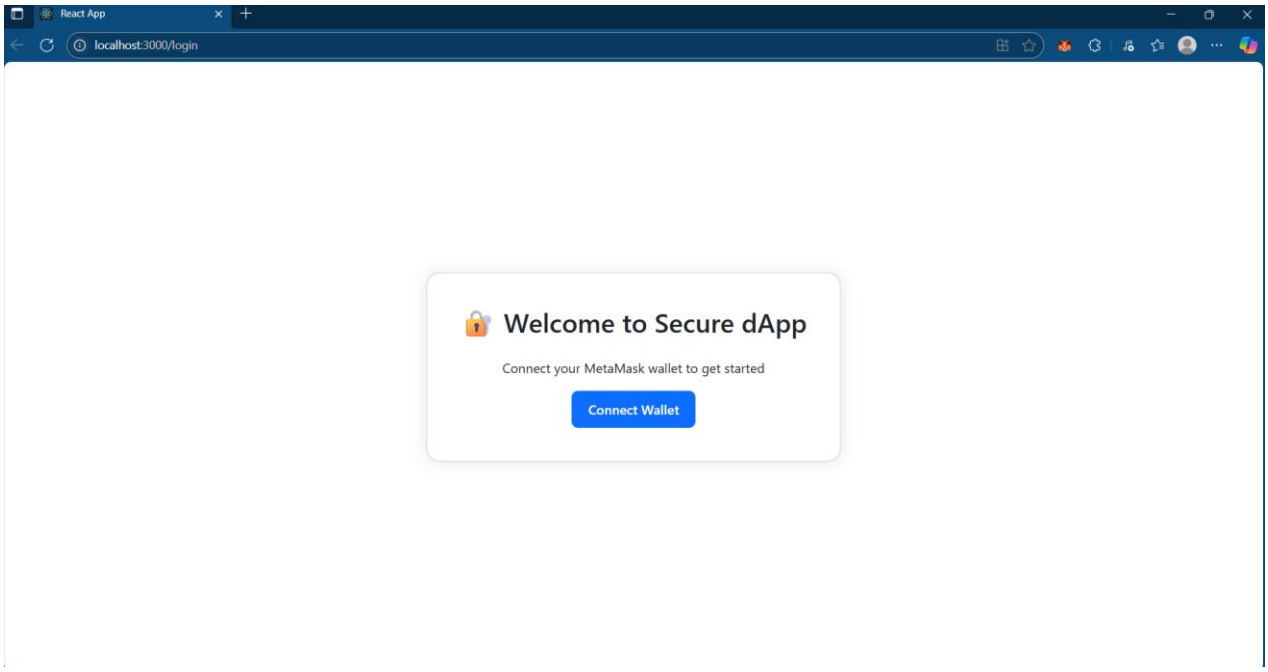

React App
localhost:3000/login
Welcome to Secure dApp
Connect your MetaMask wallet to get started
Connect Wallet

Fig :14

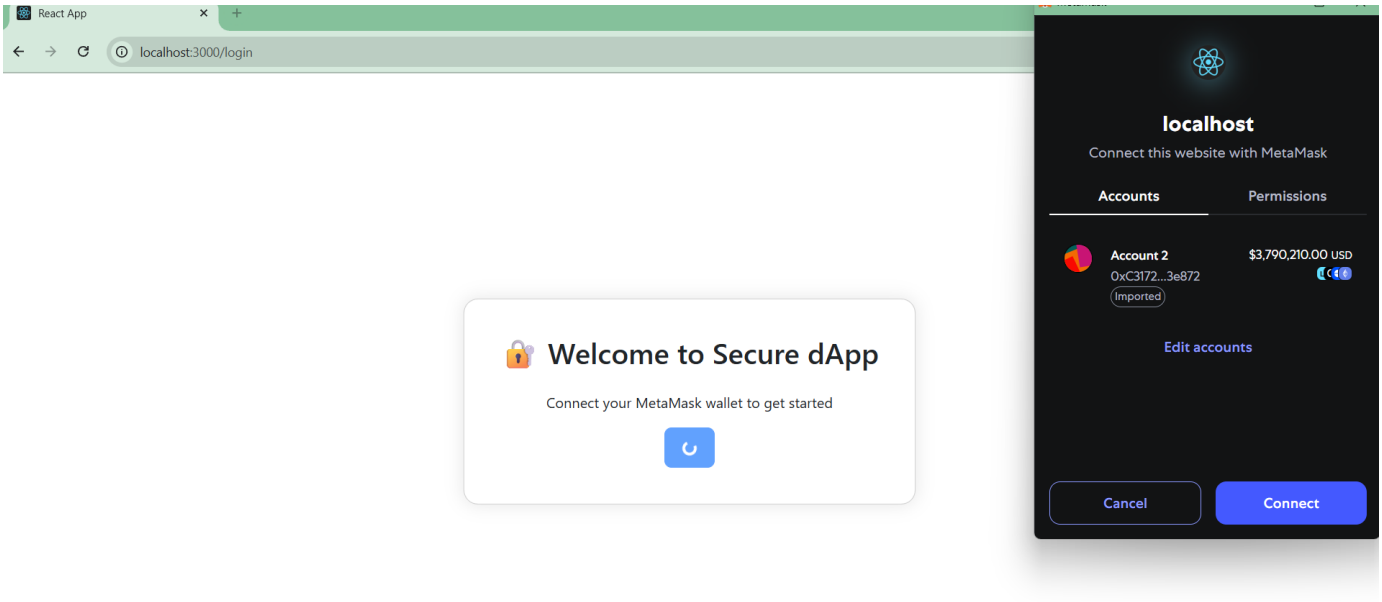

React App
localhost:3000/login
Welcome to Secure dApp
Connect your MetaMask wallet to get started
localhost
Connect this website with MetaMask
Accounts
Permissions
Account 2
0xC3172...3e872
Imported
$3,790,210.00 USD
Edit accounts
Cancel
Connect

Fig : 15

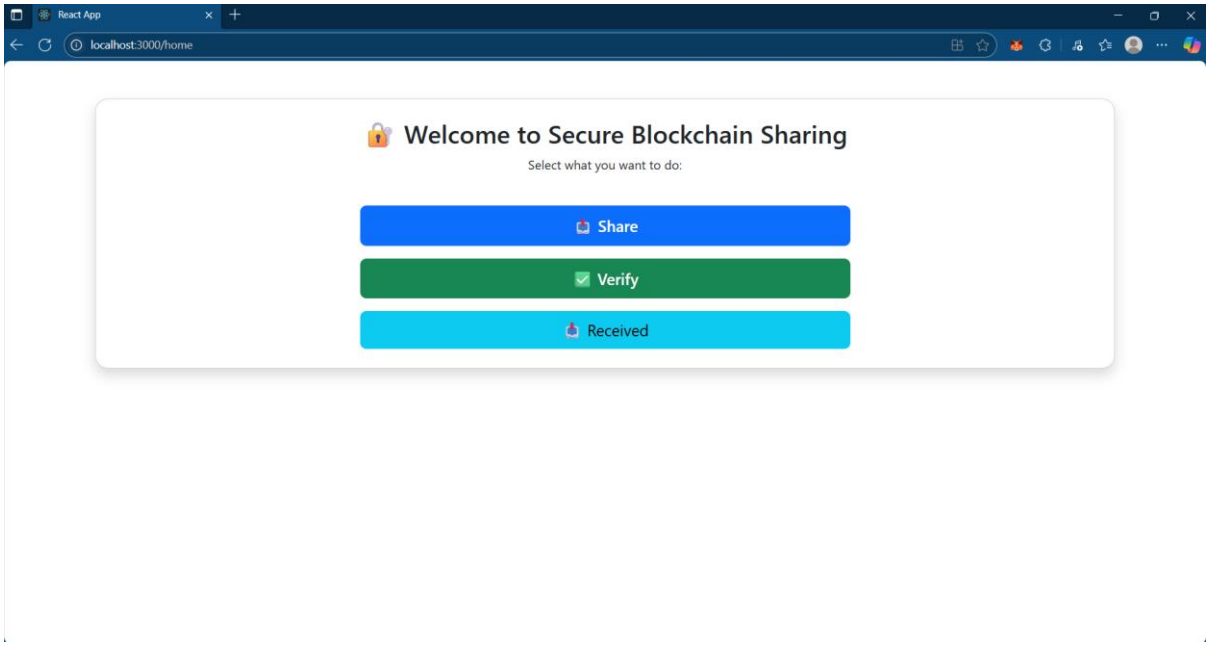


Fig : 16

6 Application Usage

6.1 Using the Application
Such application can be applied locally in single-user application, and two machines (sender and receiver) in multi-user mode. In cases where it is used by two different users, then one should adhere to the extra setup steps included below.

6.2 For Multi-User Usage (Two Different Machines):

6.2.1 Receiver Machine Setup:

1. Clone the GitHub repo:

git clone https://github.com/ShreyasJoshii/SecureBlockchain-dApp.git

6.1.1.1 Option B — Using a ZIP File or Local Copy

1. If you have a .zip file:
   - Right-click on the .zip → **Extract All** → choose a folder location.
   - Once extracted, open the folder in your terminal or VS Code.
2. If you already have the full project folder from another source (e.g., USB, email, shared drive):
   - Place it in your preferred location (e.g., C:\Projects or ~/Projects).
   - Open the folder in VS Code or navigate there using:

bash
cd path/to/SecureBlockchain-dApp

2. Install Node.js and npm (if not installed):

3. node -v
   npm -v

4. Set up the frontend:

5. cd SecureBlockchain-dApp/frontend
   npm install
6. Start the frontend:
   npm start

7. Ensure MetaMask is installed and connected to the same Ganache network:
   - Import a Ganache account (from sender or separate instance)
   - Add custom network if not already configured (same Chain ID and RPC URL)

8. Request the sender to share:
   - The blockchain hash
   - The encryption key (if used)
   - Optional: the contract ABI and address (only needed if not hardcoded in frontend)

References

developer.mozilla.org. (2023). *AES Encryption - SubtleCrypto: encrypt() method - Web APIs | MDN*. [online] Available at: https://developer.mozilla.org/en-US/docs/Web/API/SubtleCrypto/encrypt.

Express (2017). *Express - Node.js web application framework*. [online] Expressjs.com. Available at: https://expressjs.com.

metamask.io. (n.d.). *MetaMask - A crypto wallet & gateway to blockchain apps*. [online] Available at: https://metamask.io.

Node.js (n.d.). *Node.js*. [online] Node.js. Available at: https://nodejs.org.

Pinata.cloud. (2025). *Pinata | Crypto's file storage*. [online] Available at: https://www.pinata.cloud.

Reactjs.org. (2019). *React – A JavaScript library for building user interfaces*. [online] Available at: https://reactjs.org.

Trufflesuite.com. (2020). *Home - Truffle Suite*. [online] Available at: https://trufflesuite.com.

trufflesuite.com. (n.d.). *Ganache - Truffle Suite*. [online] Available at: https://trufflesuite.com/ganache/.