**FINAL REPORT**

**1. Installation guide**

**Install forge.** Forge is used to test and deploy a Solidity smart contract

|  |
| --- |
| curl -L https://foundry.paradigm.xyz | bash |

**Deployment.** This command aims to deploy a smart contract written by Solidity to specific Ethereum network.

|  |
| --- |
| forge create --rpc-url <your\_rpc\_url> --private-key <your\_private\_key> src/MLModels.sol:MLModels |

**Build server.** Flask framework is used in the project to create a web service which allows people to interact with the deployed smart contract effortlessly via function calls.

|  |
| --- |
| source contract\_connector/connector/bin/activate flask --app app.py run --host=0.0.0.0 --port=8000 |

**2. The sample data**

This is the data generated by AI models. The data is then stored in the blockchain

|  |
| --- |
| {  "timestamp": 171370555,  "threatLevel": 2,  "ipAddress": "172.14.56.70" } |

**3. The sample output**

This is data returned by interacting with a deployed smart contract through a function call

|  |
| --- |
| {    "message": "Model pushed successfully",    "txHash": "da0ae1e22ed3d08c6f056c48f43d929a025d34a4030e16a09f74d0e72aff0f83" } |

**4. The edge cases**   
The messages below indicate errors when the system makes function calls that result in errors.

A screenshot of a computer

AI-generated content may be incorrect.