**🧱 GreenLedger Ethereum Build Plan (3-Day Step-by-Step)**

**✅ Before You Start (Prep)**

**🔧 Install These:**

| **Tool** | **Purpose** |
| --- | --- |
| Node.js (v18+) | JavaScript runtime |
| npm | Package manager |
| Hardhat | Ethereum dev framework |
| MetaMask | Wallet (browser extension) |
| Ganache (optional) | Local test blockchain (GUI/CLI) |
| Infura/Alchemy | Deploy to public testnet (e.g., Goerli/Sepolia) |

**🟩 DAY 1 — Project Setup + Smart Contract**

**✅ Step 1: Create the Hardhat project**

bash

CopyEdit

mkdir greenledger-eth && cd greenledger-eth

npm init -y

npm install --save-dev hardhat

npx hardhat

Choose: **"Create a basic sample project"**

**✅ Step 2: Write Smart Contract**

Create contracts/GreenLedger.sol:

solidity

CopyEdit

// SPDX-License-Identifier: MIT

pragma solidity ^0.8.20;

contract GreenLedger {

struct Log {

string item;

string suggestion;

string action;

string result;

string store;

string timestamp;

}

mapping(string => Log) private logs;

function addLog(string memory id, string memory item, string memory suggestion, string memory action, string memory result, string memory store, string memory timestamp) public {

logs[id] = Log(item, suggestion, action, result, store, timestamp);

}

function getLog(string memory id) public view returns (string memory, string memory, string memory, string memory, string memory, string memory) {

Log memory l = logs[id];

return (l.item, l.suggestion, l.action, l.result, l.store, l.timestamp);

}

}

**✅ Step 3: Compile the contract**

bash

CopyEdit

npx hardhat compile

**🟨 DAY 2 — Local Blockchain + Contract Deployment**

**✅ Step 4: Start Local Ethereum Blockchain**

Option 1: Use Hardhat Node

bash

CopyEdit

npx hardhat node

Option 2 (GUI): Use Ganache (download from https://trufflesuite.com/ganache/)

**✅ Step 5: Deploy Contract**

Create scripts/deploy.js:

js

CopyEdit

const hre = require("hardhat");

async function main() {

const GreenLedger = await hre.ethers.getContractFactory("GreenLedger");

const ledger = await GreenLedger.deploy();

await ledger.deployed();

console.log(`GreenLedger deployed to: ${ledger.address}`);

}

main().catch((error) => {

console.error(error);

process.exitCode = 1;

});

Run this:

bash

CopyEdit

npx hardhat run scripts/deploy.js --network localhost

**✅ Step 6: Interact via Script**

You can create a small script to call addLog or test using the console:

bash

CopyEdit

npx hardhat console --network localhost

js

CopyEdit

const contract = await ethers.getContractAt("GreenLedger", "DEPLOYED\_ADDRESS");

await contract.addLog("log1", "lettuce", "donate", "price\_cut", "sold", "store\_1", "2025-07-05T21:00");

const log = await contract.getLog("log1");

console.log(log);

**🟦 DAY 3 — REST API Integration or Frontend (Optional)**

**✅ Step 7: Build a REST API**

Install:

bash

CopyEdit

npm install express ethers dotenv

Create server.js:

js

CopyEdit

require("dotenv").config();

const express = require("express");

const { ethers } = require("ethers");

const app = express();

app.use(express.json());

const CONTRACT\_ADDRESS = "DEPLOYED\_ADDRESS";

const ABI = require("./artifacts/contracts/GreenLedger.sol/GreenLedger.json").abi;

const provider = new ethers.providers.JsonRpcProvider("http://localhost:8545"); // or Infura

const signer = provider.getSigner(); // uses account[0]

const contract = new ethers.Contract(CONTRACT\_ADDRESS, ABI, signer);

app.post("/add-log", async (req, res) => {

const { id, item, suggestion, action, result, store, timestamp } = req.body;

await contract.addLog(id, item, suggestion, action, result, store, timestamp);

res.send("Log added.");

});

app.get("/get-log/:id", async (req, res) => {

const log = await contract.getLog(req.params.id);

res.json(log);

});

app.listen(3000, () => console.log("Server running on port 3000"));

**✅ Step 8: (Optional) Deploy to Goerli/Sepolia**

* Get free test ETH from a [Goerli faucet](https://goerlifaucet.com/)
* Get your private key and Infura URL
* Add to .env

Update hardhat.config.js:

js

CopyEdit

require("@nomicfoundation/hardhat-toolbox");

require("dotenv").config();

module.exports = {

networks: {

sepolia: {

url: process.env.INFURA\_URL,

accounts: [process.env.PRIVATE\_KEY]

}

},

solidity: "0.8.20",

};

Deploy:

bash

CopyEdit

npx hardhat run scripts/deploy.js --network sepolia

**📦 Final Deliverables**

| **File/Component** | **Purpose** |
| --- | --- |
| GreenLedger.sol | Smart contract |
| scripts/deploy.js | Deployment script |
| server.js | REST API |
| hardhat.config.js | Config for local + testnet |
| Demo or screenshots | Hackathon submission |

**🗣️ Hackathon Pitch (Ethereum Version)**

"GreenLedger records what AI suggested, what the store did, and what the outcome was — all transparently on the Ethereum blockchain. This makes actions auditable and outcomes shareable across stores to improve sustainability."

Ready to generate this full project ZIP or GitHub repo for you?