SocioHack

SocioHack is a proof-of-concept decentralized identity and credential platform combining Ethereum smart contracts, IPFS, and a Node/React full-stack server and frontend.

Status: Development

Contents:

- Smart contracts (Truffle) in the Blockchain folder
- Server and tests in the Server folder
- React frontend in the src folder
- Utilities in the Utils and src/utils folders

Key Features

- DID and identity registry smart contract
- Issuer / Holder / Verifier flows implemented in backend and frontend
- IPFS integration for storing artifacts

Prerequisites

- Node.js (>=16 recommended)
- npm
- Ganache (for local Ethereum test node)
- IPFS (optional — tests may require IPFS)
- Truffle (for migrations)

Quickstart (local development)

1. Start Ganache (local Ethereum):

```
ganache
```

2. (Optional) Start IPFS daemon offline (if testing IPFS flows):

```
cd D:\kubo_v0.39.0_windows-amd64\kubo
.\ipfs.exe daemon --offline
```

3. Deploy smart contracts (from project root):

```
cd Blockchain
truffle migrate --reset
```

4. Start the backend server (Server folder):

```
cd Server
npm run dev
```

5. Start the frontend (from project root):

```
npm install
npm start
```

Alternatively use the included VS Code tasks (see `.vscode/tasks.json`) — run the "5. Launch Full Stack" task to run the sequence.

Running Tests

- Server tests are in Server/Test. Run from the `Server` folder:

```
cd Server
npm test
# or use jest on a single test
npx jest ./Test/7_CredentialIssue.test.js
```

Folder Overview

- Blockchain: Truffle config, contracts, and migrations.
- Server: Node backend, tests in `Test/`, utilities in `Utils/`.
- src: React frontend, main components in `src/components`.
- public: Frontend static assets and `index.html`.

Notable files

- Contract build artifact: Blockchain/build/contracts/IdentityRegistry.json
- Server entry: Server/Server.js
- Frontend entry: src/index.js

Development Notes

- If a test fails related to IPFS, ensure an IPFS daemon is running (or mock/adjust tests).
- Tests and server rely on the contracts being migrated to the local Ganache network.
- If ports conflict, adjust `Server` or frontend port settings.

Next steps

- Run migrations and start the stack, then exercise Issuer/Holder flows in the UI.
- I can also add a CONTRIBUTING guide and environment variable examples if you want.

Generated and updated by GitHub Copilot assistant.