Blockchain for Industrial Engineers: Decentralized Application Development

บล็อกเชนสำหรับวิศวกรอุตสาหการ: การพัฒนาแอปพลิเคชันแบบกระจายศูนย์

Decentralized application (DApp)

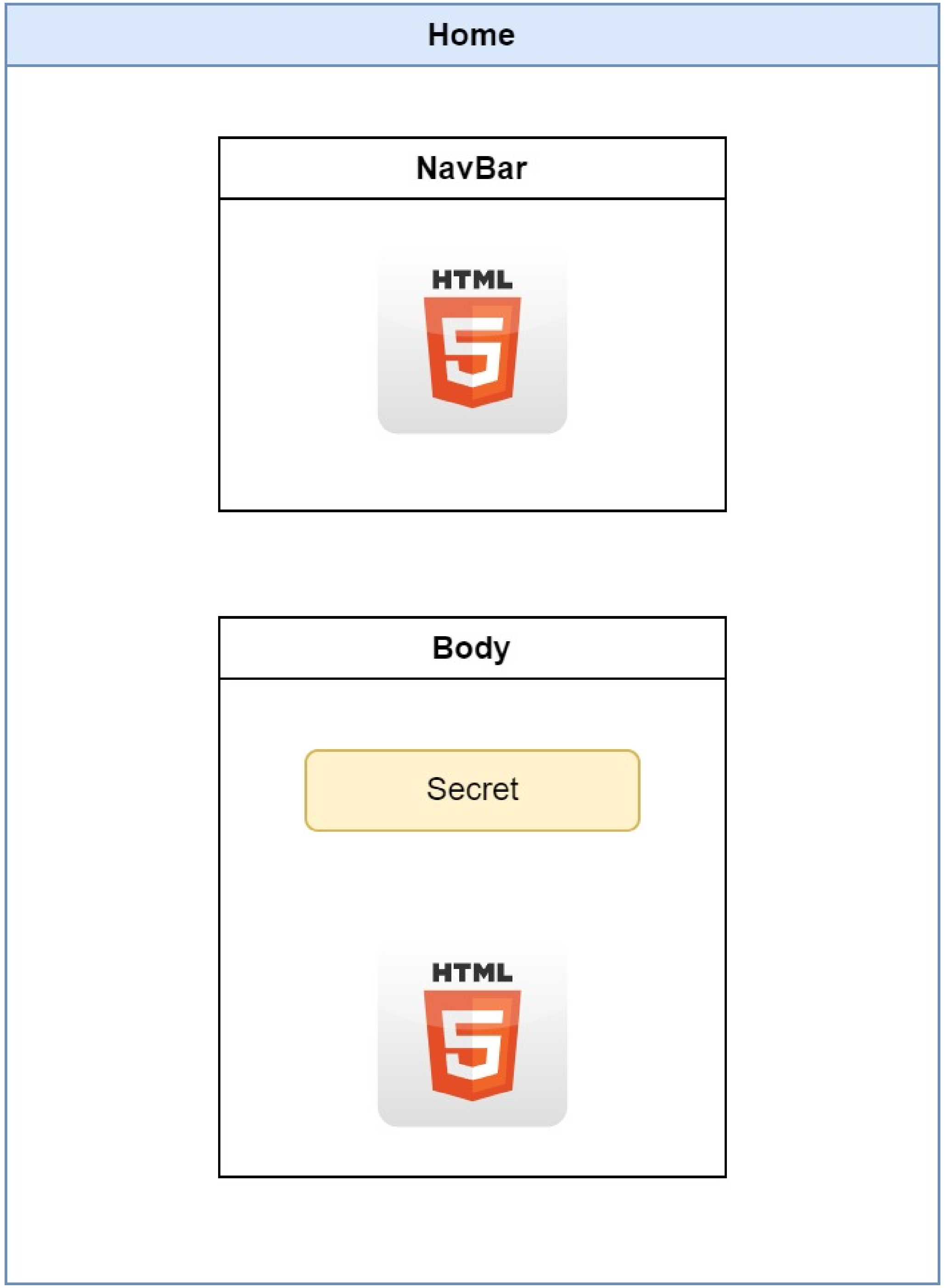
Part 1

Template

- Fork the following repository
  - https://github.com/nnnpooh/blockchain-class-dapp
- Clone your new forked repository to local machine
  - `git clone ......`
- Open the newly cloned folder in `VSCode`
- Install library
  - `npm install`
- Run
  - `npm run dev`

Structure

- `Home` component
  - `NavBar` component
  - `Body` component
- Reactive variable
  - `secret`



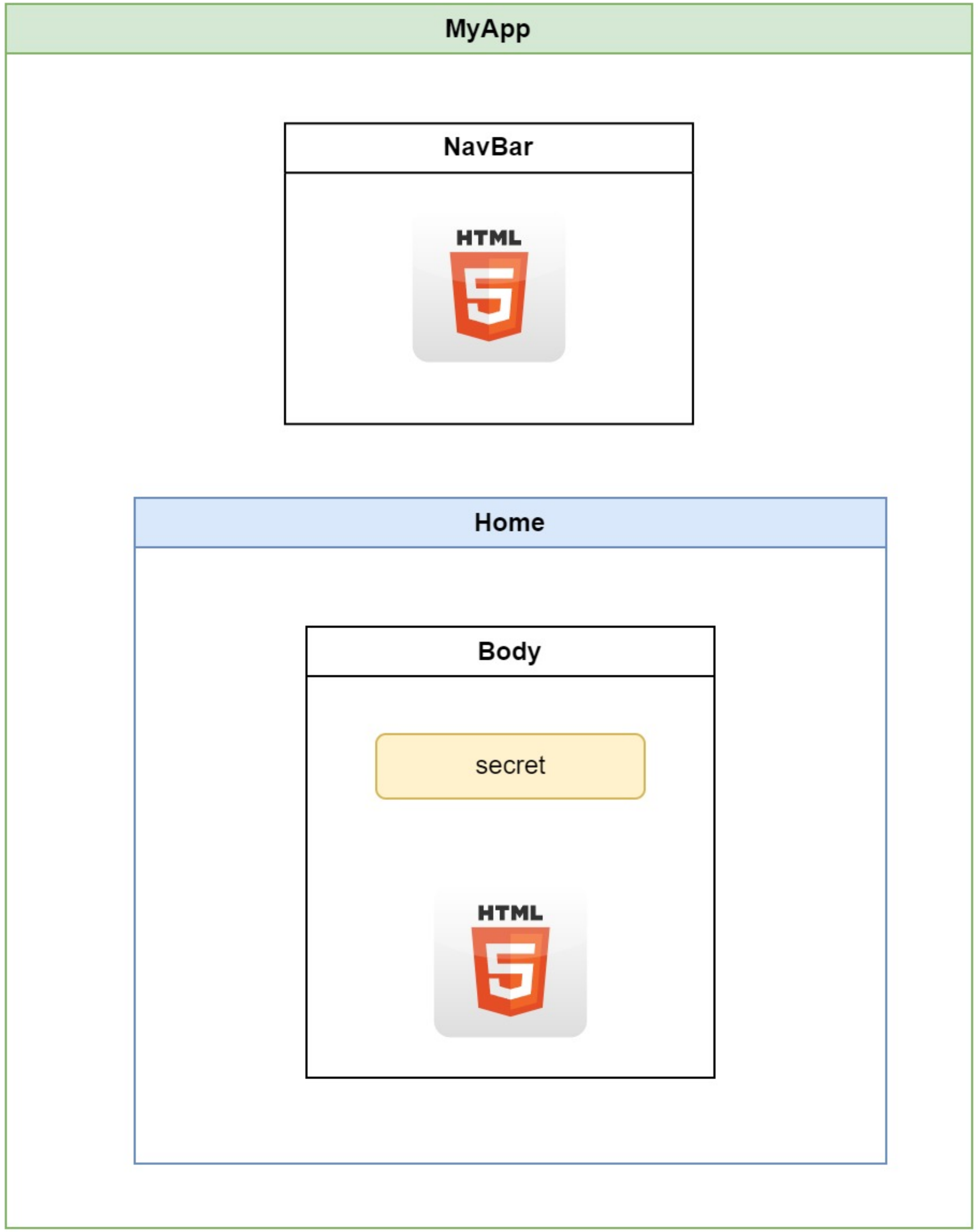
Part 2

Library

- Install `ethers` library
  - `npm install ethers@5.7.2`

`_app` file

- Manage layout



Centralized store

Stores

- `useWorkingStore()`
  - `secret`
- `useMetaMaskStore()`
  - `account`
  - `balance`
  - `chainId`
  - `isEthereumAvailable`
  - `provider`
- `./src/utils/stores.ts`

Startup Logics

- Open MetaMask
- Setup listener
  - Change account
  - Change network
- `./src/utils/useEthereum.ts`
  - Using `useEffect` function
- Insert `useEthereum()` into `_app.tsx`

Fix `TypeScript` Error

- `types/index.d.ts`

```
import { ExternalProvider } from "@ethersproject/providers";
import { MetaMaskInpageProvider } from "@metamask/providers";

declare global {
  interface Window {
    ethereum?: MetaMaskInpageProvider;
  }
}

export {};
```

- `tsconfig.json`

```
...
"typeRoots": ["./node_modules/@types", "./src/types"],
...
```

React to change in state

- When you change account and network in Metamask, the application should react to it.
  - `useEffect` with dependency.
- ./src/utils/useMetaMask.ts
- Insert `useEthereum()` into `_app.tsx`
- Insert `useMetaMask()` into into `_app.tsx`

Update `AccountButton`

- AccountButton