Setup Environment

I am using WSL2(Ubuntu 18.04 LTS) on my machine. I think this tutorial can work on both WSL and WSL2. If you want to set up WSL2, please check out this link https://docs.microsoft.com/en-us/windows/wsl/install-win10.

Install NodeJs:

> Important:
>
> 1. Do not use root user to install `npm` , each user should have its own `npm`
> 2. Install your Nodejs in the partition of your WSL. Install it in your Windows partition will cause permission error later.
> 3. Your project directory must be in the partition of WSL too.

1. Open your Ubuntu 18.04 command line.
2. Install cURL with: `sudo apt-get install curl`
3. Install nvm, with: `curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.36.0/install.sh | bash` To verify installation, enter: `command -v nvm` This should return 'nvm', if you receive 'command not found' or no response at all, close your current terminal, reopen it, and try again.
4. List which versions of Node are currently installed (should be none at this point): `nvm ls`
5. Install the current release of Node.js (for testing the newest feature improvements, but more likely to have issues): `nvm install node`
6. Install the latest stable LTS release of Node.js (recommended): `nvm install node --lts`
7. List what versions of Node are installed: `nvm ls` Now you should see the two versions that you just installed listed.
8. Verify that Node.js is installed and the currently default version with: `node --version`. Then verify that you have npm as well, with: `npm --version` (You can also use `which node` or `which npm` to see the path used for the default versions).
9. change your directory to the user side, use `npm install` to install all the dependencies
10. use `npm start` to run the project

Work on your project with VS code

1. Install **Remote - WSL** extension for your VS code. (https://marketplace.visualstudio.com/items?itemName=ms-vscode-remote.remote-wsl
2. Use the extension to connect to your WSL remotely
3. Work on your project as usual!!

Ant Design Framework

The UI framework the project is currently using is the Ant Design. Please use the following links for more detailed documentation.

https://ant.design/

https://pro.ant.design/

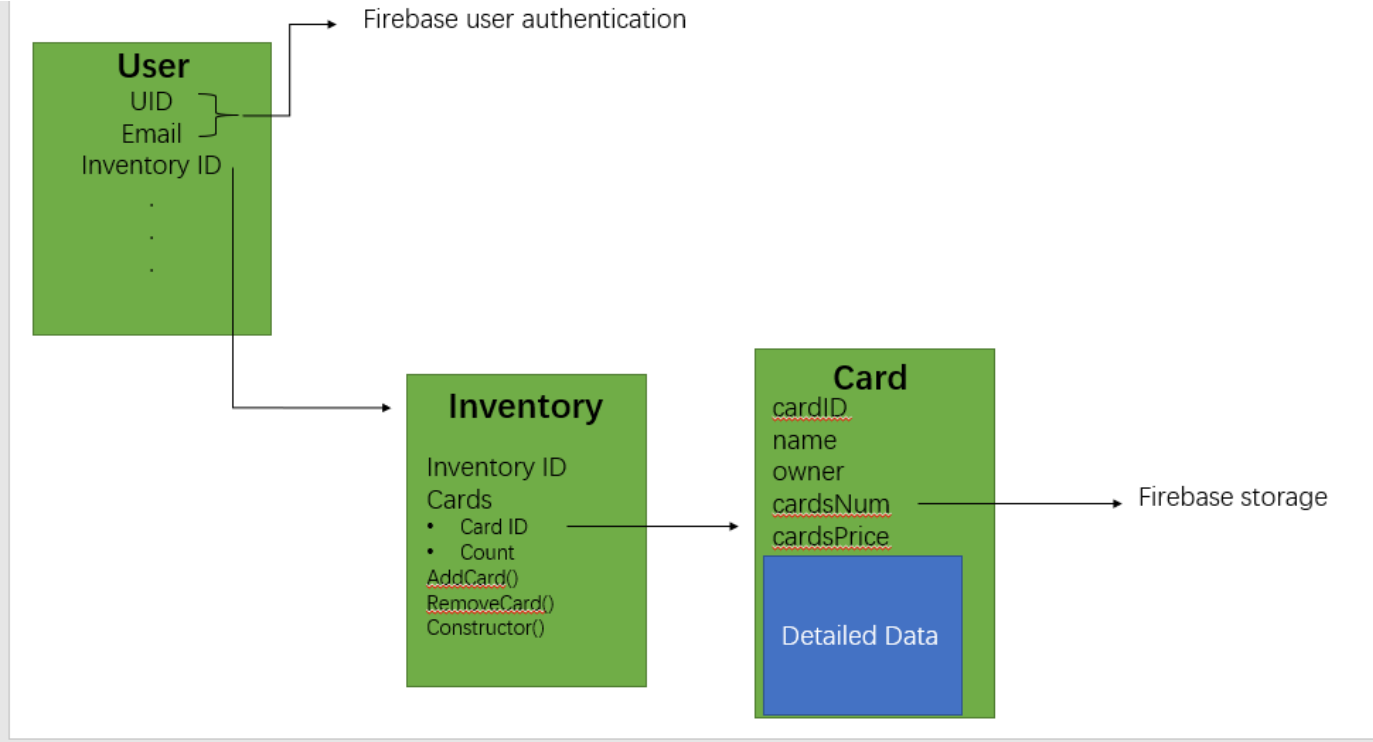
https://procomponents.ant.design/en-US

We have kept the boilerplate code in the project code base for ease of understanding and future extension. If you find Chinese in the code, no need to worry. They are part of the boilerplate code and have little connection with the application. However, be very careful if you want to remove or modify boilerplate code. It may break the application.

Data modal

The data of the application is located at google firebase Fire store.

- Currently, the database has threat data modal. User, inventory, and Card.
- The Inventory ID field of the user point to the inventory modal of that modal,
- Each inventory modal has an array of a pair of Cards ID and count.
- Each Card ID points to the corresponding cards



Login & password

- Currently, we use firebase user authentication to manage user information.
- default user name: zzm@gmail.edu
- default password: 123456

ToDo:

- Rework the `buyCard` method in src/services/ant-design-pro/api.ts
  - Reviewing the above data diagram for a general idea to interact with the data modal
  - The information of the user will be stored by using a customized hook plugin provided by the framework. Check out this link for detailed information.

    ```
    • const {initialState, setInitialState} = useModel
    ('@@initialState');
     const uid = initialState?.currentUser?.uid;
    ```

    This code snippet shows how to use the global InitailState. https://beta-pro.ant.design/docs/initial-state
  - Currently, you need to fetch the user information from the firebase to get the user's inventory id. And then use that inventory id to fetch the detailed information of the inventories. In the future, you can save the inventory ID as part of the InitialState to avoid one async operation.
- Create a new middleware in the remote and move the implementation of the APIs from the front end to the middleware.
  - Currently, all the connections between local and firebase are included as part of the web applications. It would be better to create a middleware to connect the front end and the database for better encapsulation and performance.