Pre-requites

* install MySQL (user/password: root/jing9193; database: nodedb);
* create folder: backend/upload

1. Download packages:

git clone https://github.com/yannonghuang/shinshin

1. Install packages:

cd backend; npm install;

cd react-app; npm install;

1. Run

cd backend; node server;

cd react-app; npm start;

1. Initial db instance

Use mysql client tool (i.e., MySQLWorkbench) to import a db dump

1. Initial account

Username / password : shinshin / shinshin

1. Build & Deploy & Production

cd react-app; npm run build

cd backend; node server (only backend server is necessary, frontend is bundled in backend. Port number is defined with environment variable PORT, default 8080).

1. Secure Run

cd react-app; npm run build

cd backend; **node server sure** (only backend server is necessary, frontend is bundled in backend. Port number is defined with environment variable SECURE\_PORT, default 443. Secure run needs a private key key.pem and a certificate cert.pem stored under folder backend/security).

A sample pair of private key and certificate files are in security.zip. To use, unzip security.zip to backend/security folder.

git commands

* check out:

cd shinshin

git pull

* add/push new file

git add<file>

git commit -m “…”

git push

db migration tips

* execute after db schema changes

cd backend/app; npx sequelize-cli db:migrate

* change db schema

cd backend/app; npx sequelize-cli migration:create --name <file name, ex: modify\_users\_add\_new\_fields>

*edit* .js files under migrations/ …

Generate private key & certificate

openssl generates a self-signed certificate valid for 9999 days:

cd backend/security;

openssl genrsa -out key.pem;

openssl req -new -key key.pem -out csr.pem;

openssl x509 -req -days 9999 -in csr.pem -signkey key.pem -out cert.pem