README.md

nwen304-project3-team-sweet

this holds an simple e-commerce website written in node + express + postgres which has a heroku deployment: quiet-cove-52767.herokuapp.com and the repo is here

how to run on a local dev machine

database

you can either run postgres locally which you will need change `connectionString` in `models\index.js` or use docker-compose which instructions on how install are here if you do that then each time you want to run the server you will need to run `docker-compose up` and then if you want to destroy database then run `docker-compose down`

server

firstly you will need to run these commands:

```
npm install node-pre-gyp passport passport-facebook passport-local
npm install -g nodemon
```

then you can run either `npm test` or `npm start` if you running the database using the docker-compose you will need to wait a few seconds

api endpoints

user login and auth

url endpoint	request type	description
/register	POST	registers a user by passing a json object which has the user information e.g {username: gygy, password: jhfttf}
/login	POST	logins a user in by passing a json object in the body e.g {username: gygy, password: jhfttf}
/logout	POST	logouts a user

user cart api

url endpoint	request type	description
/user	GET	accesses the currently logged in user info in user table
/addtocart	PUT	adds a product to the users that is logged in by passing a json object in the body e.g {prodId: 1, amount 2}
/user/cart/:id	GET	gets info of a product in the users cart based of the product id e.g /user/cart/2 will get the product that has an product id of 2
/user/cart/amount/:id	POST	changes the amount of a certain product that is in the database based of the url parameter which will have the product id that you want to change
user/cart/delete/:id	DELETE	deletes a product from the cart based of its product id

products api

url endpoint	request type	description
/products	GET	gets all products in the products table
/products/:id	GET	gets a product based off its id

api curl requests

local user registration:

```
curl -X POST \
  http://localhost:8080/register \
  -H 'cache-control: no-cache' \
  -H 'content-type: application/json' \
  -H 'postman-token: 95dcbe00-3ed6-2620-1810-a540953c0bf4' \
  -d '{"username":"ydfygd","password":"uhgy"}'
```

login user:

```
curl -X POST \
  http://localhost:8080/login \
  -H 'cache-control: no-cache' \
  -H 'content-type: application/json' \
  -H 'postman-token: f7b9e5a9-5fea-ea2b-b0d4-044867ff583e' \
  -d '{"username":"ydfygd","password":"uhgy"}'
```

lists all products in db:

```
curl -X GET \
  http://localhost:8080/products \
  -H 'cache-control: no-cache' \
  -H 'postman-token: 397f334d-1103-95a7-46ea-e6de8f50bcf9'
```

logout user:

```
curl -X POST \
  http://localhost:8080/logout \
  -H 'cache-control: no-cache' \
  -H 'content-type: application/json' \
  -H 'postman-token: 4ba34190-eeec-0d51-b9af-373ee8f12954' \
  -d '{"user":"ydfygd","pass":"uhgy"}'
```

lists all products in user cart:

```
curl -X GET \
  http://localhost:8080/user/cart \
  -H 'cache-control: no-cache' \
  -H 'postman-token: 75c86803-319d-7f61-cee4-a9fa0e3fb303'
```

get product based of id

```
curl -X GET \
  http://localhost:8080/products/11 \
  -H 'cache-control: no-cache' \
  -H 'postman-token: 17b7cd81-294a-0653-881d-8feeb66524ed'
```

adds product to cart:

```
curl -X POST \
  http://localhost:8080/addtocart \
  -H 'cache-control: no-cache' \
  -H 'content-type: application/json' \
  -H 'postman-token: 3fe472e3-5b39-fa46-d3a3-5ee2a10db124' \
  -d '{"prodId":1,"amount":1}'
```

search for a product:

```
curl -X POST \
  http://localhost:8080/search \
  -H 'cache-control: no-cache' \
  -H 'content-type: application/json' \
  -H 'postman-token: 915dba11-0b28-315e-669d-f3bba1b3cc99' \
  -d '{"product_name": "Unbranded Concrete Keyboard"}'
```

database schema

users

holds the password and username of each user

user_id SERIAL PRIMARY KEY NOT NULL	username TEXT	password TEXT NOT NULL	fbid TEXT NOT NULL	token TEXT NOT NULL	fbname TEXT NOT NULL

(user)_cart

hold all the items that are in the users cart

id SERIAL PRIMARY KEY NOT NULL	product_id BIGINT	amount BIGINT

products

holds info about all the products that are in the store

products_id SERIAL PRIMARY KEY NOT NULL	product_name TEXT NOT NULL	product_des TEXT NOT NULL	price TEXT	in_stock INTEGER	picture_dir TEXT