

Readme explaining how to use the e- commerce platform

This e- commerce platform created by team of third year students for NWEN304 as part of group project:

- Elaf Almusaid
- Hayden Pomare
- Kitt McEvoy

GitHub link: <https://github.com/kittmcevoy/NWEN304-GroupProject>

Site is hosted on Heroku: <https://nwen304-group-host.herokuapp.com/>

To run:

1. Navigate to root directory (where README is located)
2. npm install
3. npm start
4. open browser and go to "localhost:3000"

Current features:

- Create user account
- Password encryption and decryption
- Sign up feature working
- Google account login
- Login feature working
- Password reset
- Dynamic web pages
- Read and display items in browser
- Add new items
- Delete items
- Update items
- API items / API users
- Cart
- Order
- Session and cookies
- Timeout after inactivity
- Error handling functionality
- MVC architecture
- Hosted on cloud platform
- Mocha and chai tests

Rest commands included for testing

URL/HTTP verbs	Description	Response state when Successful	Response state when Failed
'/' + get	Home page routes	Listening on port 3000 connected	500 internet service error
'/login' + get	Login page	200 OK	500 internet service error
/local-login + post	Login	Render to home page	401 Unauthorized response status
/local-signup + post	Signup	Render to home page	401 Unauthorized response status
/logout + get	Logout	Render to home page	Send and error message
/auth/google + get	Login page	Authenticate with google login	
/google/callback + get		Render to home page	401 Unauthorized response status
/add-item + get	Add item	Render add item page + 200 OK	500 internet service error
/edit/:id + get	Update item	Render update item page	
/edit-item/:id + post	edit item	Render home page + 200 OK	500 internet service error
/browse + get	browse	Render to the browse page	
/delete/:id + get	Delete item	Without a GUI	500 internet service error
/401 + get	Error handling	Render 401 page	
/403 + get	Error handling	Render 403 page	
/cart + get	cart	Render cart page	404 error page not found
/order + get	Order	Render order page	404 error page not found
/api/items + get	Items	Render all items without GUI	404 error page not found
/api/users + get	Users	Render all users without GUI	

/api/items/:size + get	Random item size	Render the item size randomly	
/api/users/:size + get	Random user size	Render the user size randomly	404 error page not found
/api/items/id/:id + get	Get item by ID	Render the item if exists	
/api/users/id/:id + get	Get user by ID	Render the user if exists	
/api/items/update/:id + get	Update item price	Render updated item	

Error handling has been implemented in our system:

For example:

```
app.get('/401', (req, res) => {
  res.render('401.ejs');
});

app.get('/403', (req, res) => {
  res.render('403.ejs');
});
```

```
/* 404 page if a user typed the wrong URL */
app.use((req, res, next) => {
  res.status(404).render('404.ejs');
});
```



