NWEN304 final group document

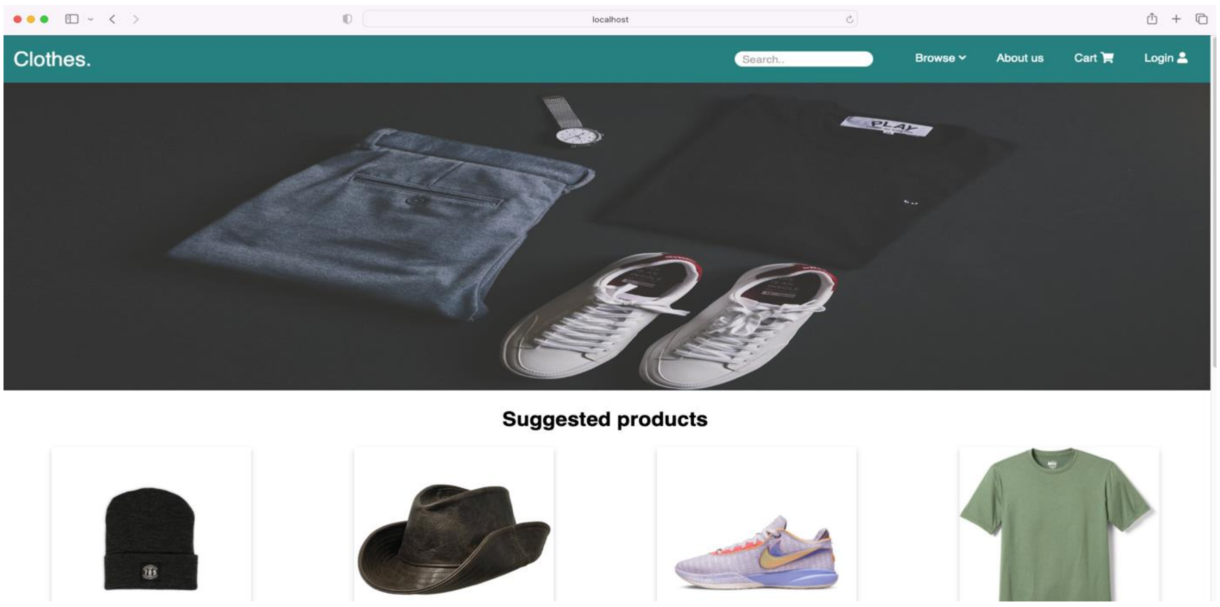
This e- commerce platform created by team of third year students for NWEN304 as part of group project:

- Elaf Almusaid
- Hayden Pomare
- Kitt McEvoy

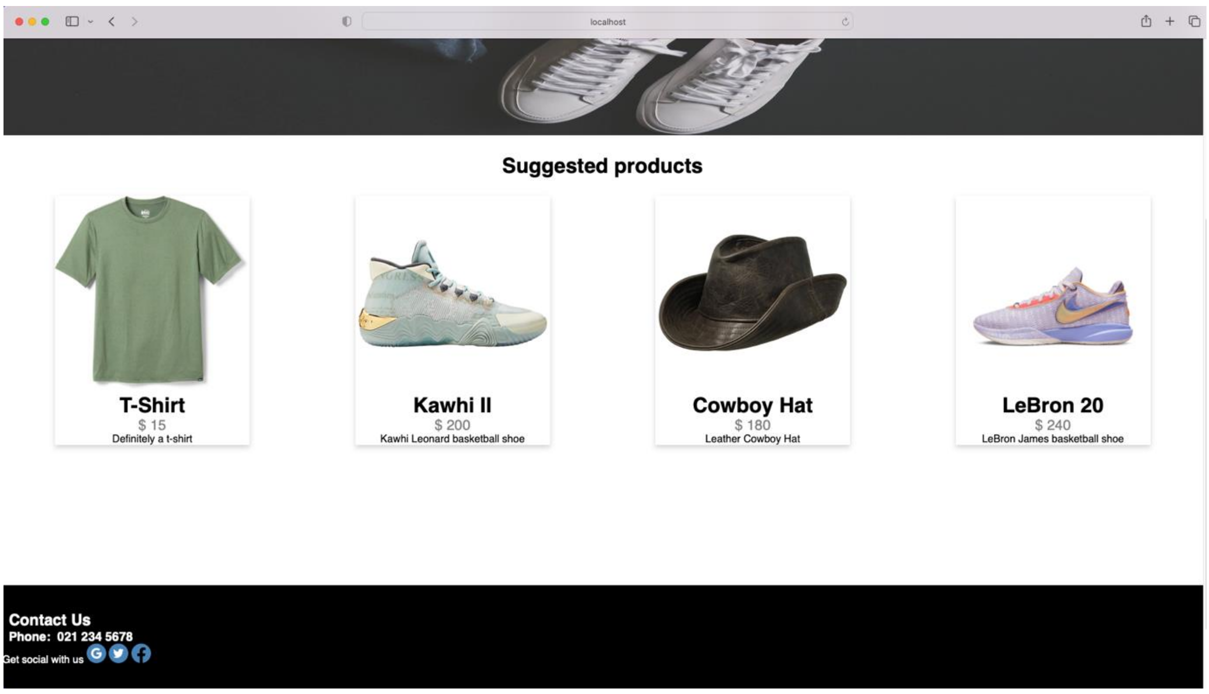
1.0 Introduction

This document will draw upon and evaluate the overview of the e- commerce platform that allow user to browse shopping items such as clothing, pants, shoes and accessories. The application requirements were to design a secure, private, RESTful, has navigation functionality and scalable application. The design of e-commerce platform was formed to meet the application requirement by using various ranges of resources and programming languages techniques. These include, third party libraries such as mongoose, express, ejs, passport, cookie-parser, chai, mocha and aws-sdk. User can sign in through their email, username and password, the system can also suggest new secure password and agree to the terms & condition to the e-commerce platform. We also have login/logout system were user can login through their Google API. Furthermore, the application development process was divided into stages. The first stage of the application was the security which is one of the major requirements of the application. Decision on security of the application was users needed to be authenticated before using the application and this was achieved by two options, local signup or using media website such as Google to login. Additionally, one of the application's main functions is that user need to login and stay login to be able to browse items and shop.

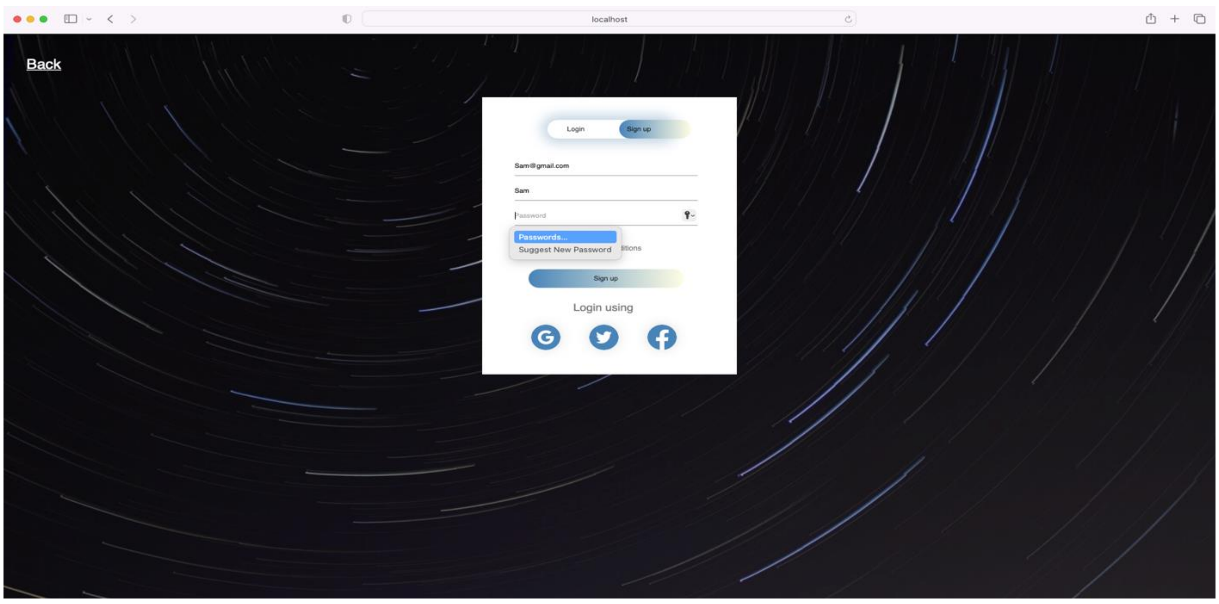
2.0 How to use the platform

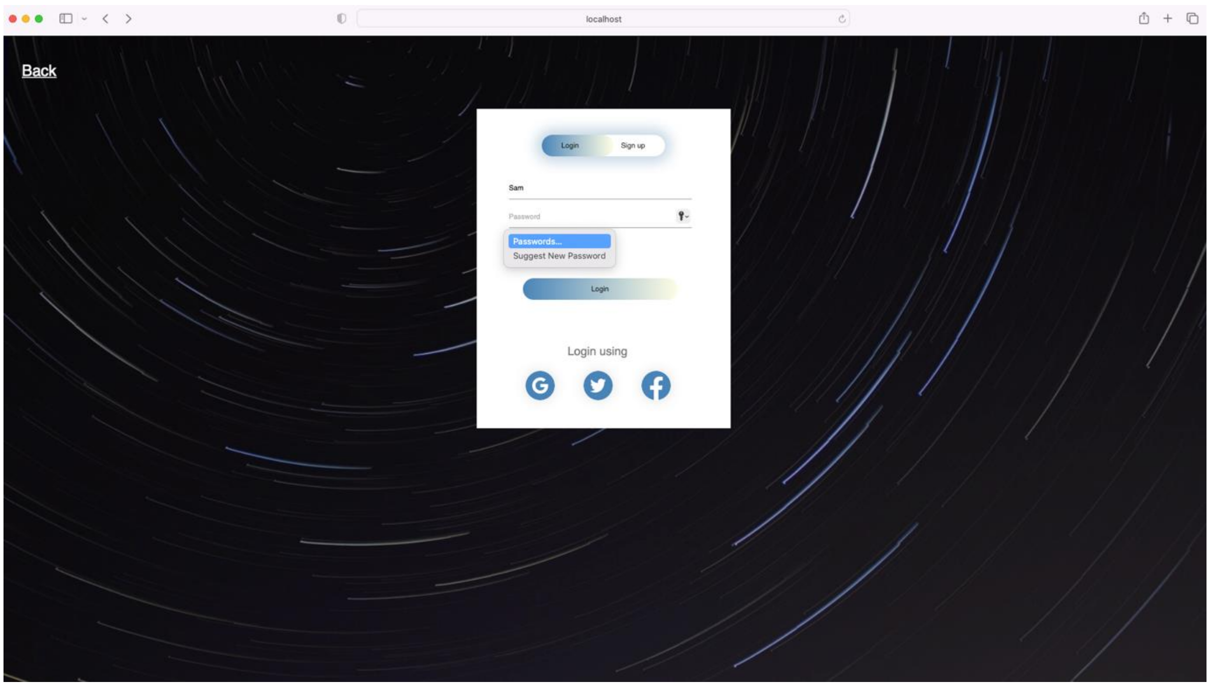


when you first load up the page, you will be led to our main page there are number of ranges things you can do. At the top there are navigation bar with dropdown Browse, About us information, Cart and login. Where the Cart only available once user login and have more accessible features like Orders. If you scroll down you view random items with their title, price and description. With some details in how to contact us which is either by contacting our phone number or visit our social platform.

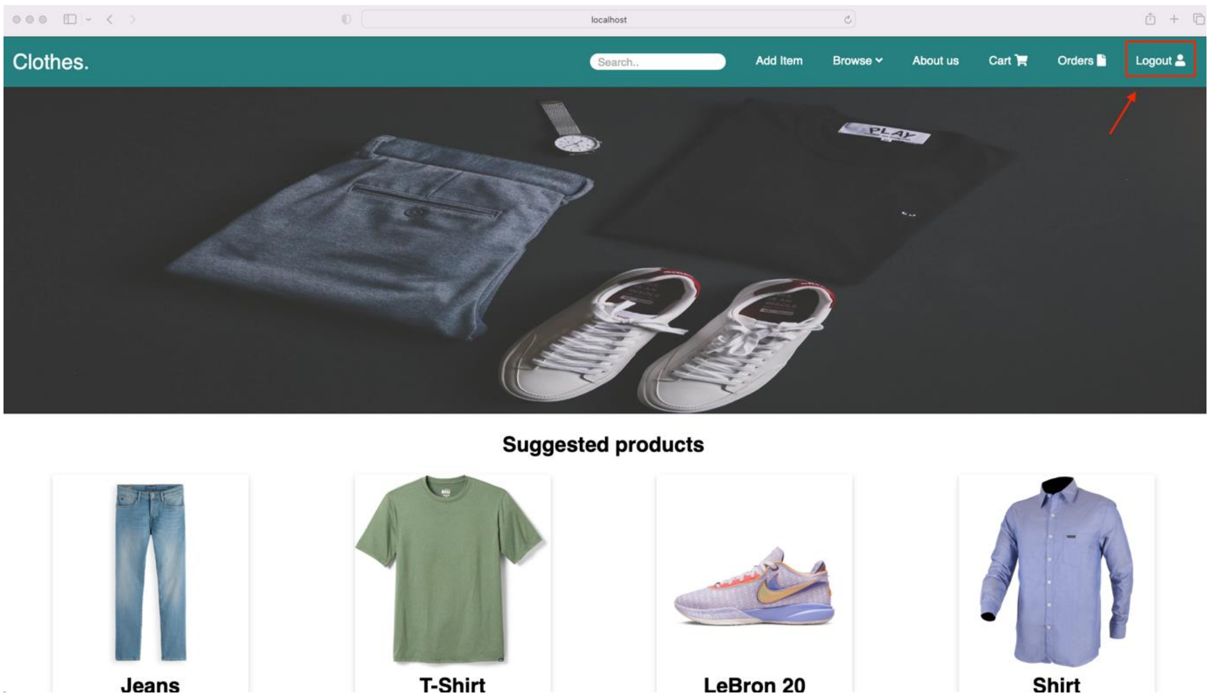


We are then taken to sign up / login page and ask to either sign in locally with email address, username and password. Or through social media platform like Google. If we entered the correct information here we successfully will be taken back to the main page but we will now be logged in.





Once we login, will have a user id in the mongoDB and can login next time easily and be able to browse and shop. We can now see from the navigation bar that we have been logged in, and can add items to the cart and see our orders.



3.0 Design Requirements

Security

Security is a set of measures taken when designing, coding and deploying the application. We way we implement our code take these measures into account to prevent the system from being exposed to internal and external threats/ attacks such as cross site scripting (XSS) and cross site request forgery (CSRS). To achieve high security when designing the application, we used session and cookies for user authentication and only allow these users to Create, Get and delete items. With the aid of a third-party library both local and google authentication was achieved. For local authentication, “passport-local” and for Google authentication “passport-google-oauth2” library was used. The passwords are encrypted before being send to the database for storage. We also, implemented cookie using third party library called “cookie-parser”. Cookie simply is a way to know that user is authenticated we can store the information in the browser on the client side.

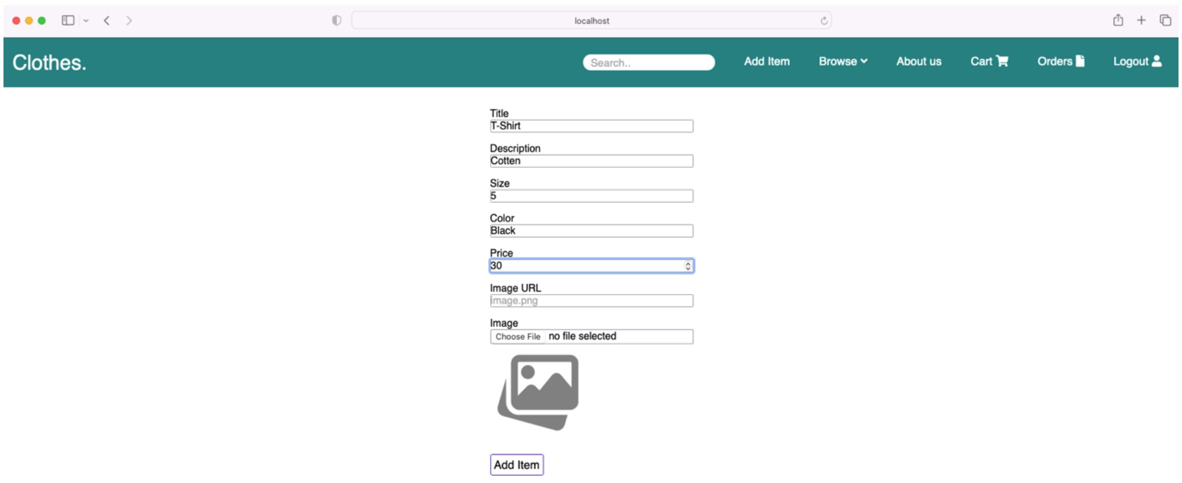
RESTfull

One of the main specifications of the application was ensuring that it supports RESTful as a distributed system approach. These involves 4 HTTP verbs. Three out of the four verbs were used whilst making the application, these included GET, POST, DELETE. The fundamental areas of POST were implemented when a new user sign up, login, add item, edit item by id, a POST verb was used to send information to the database and return a message (200) indicating that a successful request was stored. Secondly, the verb DELETE was utilized to have the ability to delete item from mongoDB by using the id of item to be removed. Finally, the verb GET was employed when a user successfully logged in, logout, browse, cart, order, API items, API users and error handling pages.

MongoDB

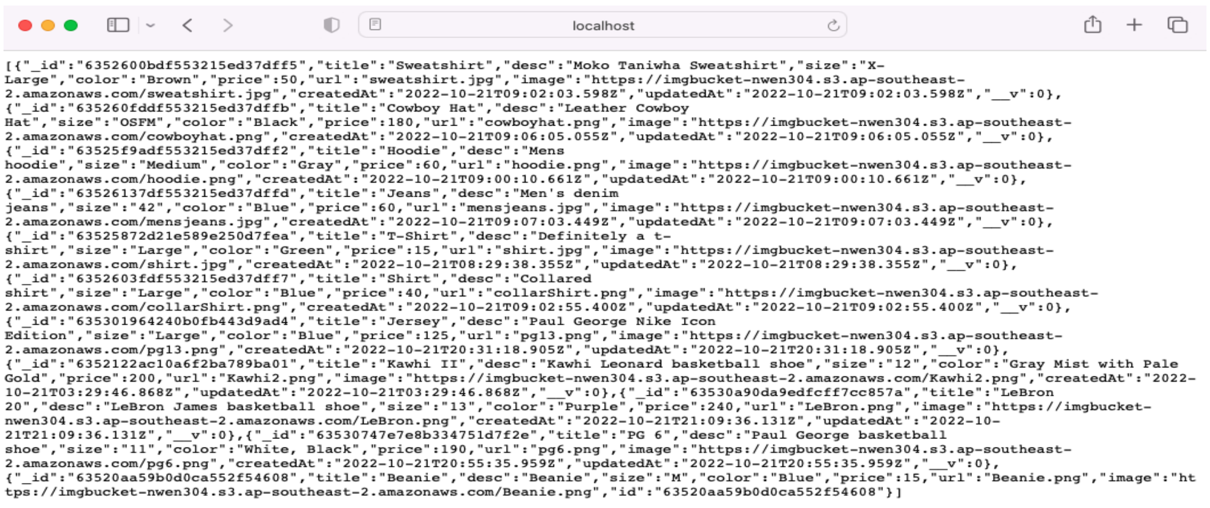
Another server side the application hosted is MongoDB, which ensures efficiency. The decision to use MongoDB database as using mongoose drive as mongoose is an object database modelling tool that allows users to skip some lower level operation and based on the fact that we can store items and user’s information. More importantly, mongoDB has greater suitability for NodeJS and advance schema-based feature.

Adding item to the database page have different field to fill such as Title, Description of the item, Size, Color, Price, Image URL and uploading image from a file.



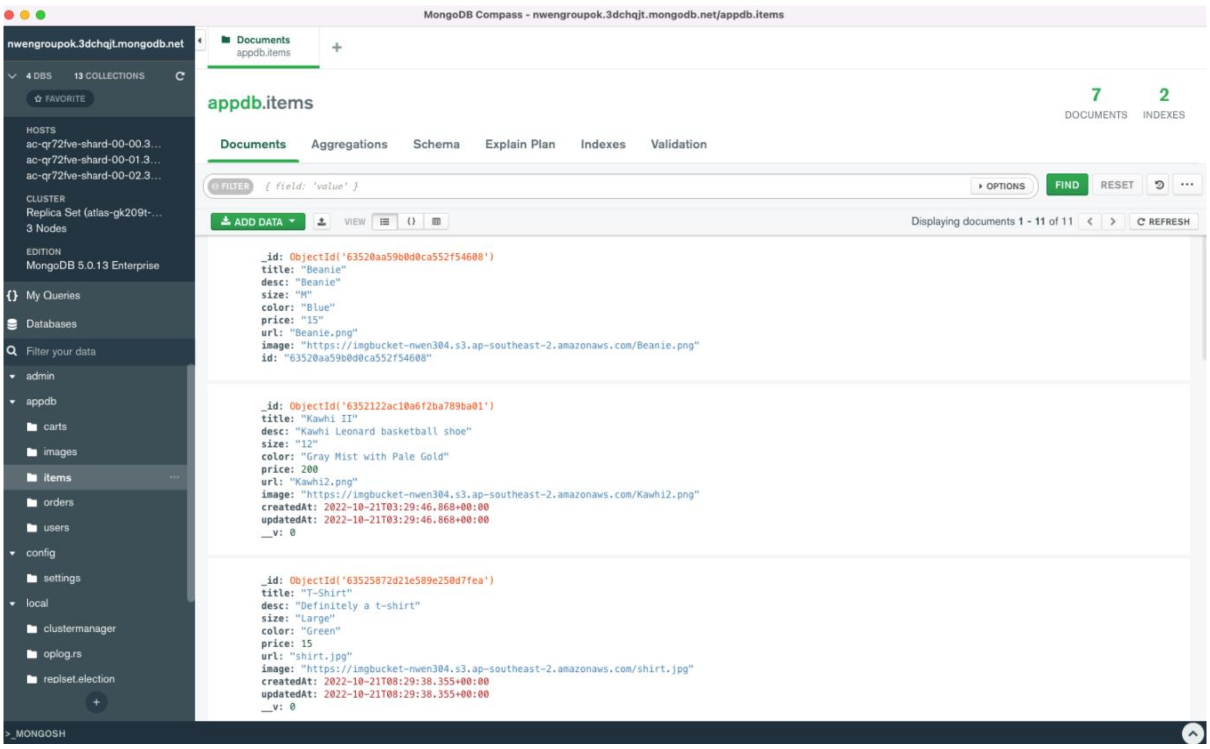
Title
T-Shirt
Description
Cotten
Size
5
Color
Black
Price
30
Image URL
image.png
Image
no file selected

After adding the items, we can then see them through API without a GUI
Using: http://localhost:3000/api/items



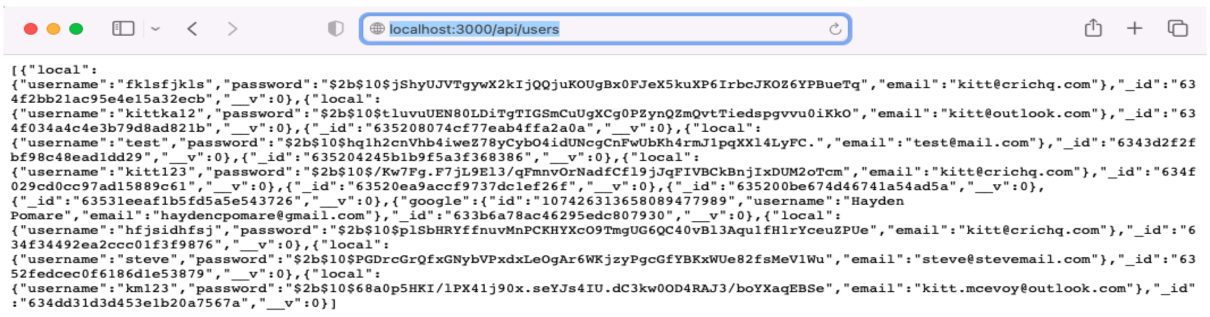
```
[{"_id":"6352600bdf553215ed37dff5","title":"Sweatshirt","desc":"Moko Taniwha Sweatshirt","size":"X-
Large","color":"Brown","price":50,"url":"sweatshirt.jpg","image":"https://imgbucket-nwen304.s3.ap-southeast-
2.amazonaws.com/sweatshirt.jpg","createdAt":"2022-10-21T09:02:03.598Z","updatedAt":"2022-10-21T09:02:03.598Z","__v":0},
{"_id":"635260fddf553215ed37dffb","title":"Cowboy Hat","desc":"Leather Cowboy
Hat","size":"OSFM","color":"Black","price":180,"url":"cowboyhat.png","image":"https://imgbucket-nwen304.s3.ap-southeast-
2.amazonaws.com/cowboyhat.png","createdAt":"2022-10-21T09:06:05.055Z","updatedAt":"2022-10-21T09:06:05.055Z","__v":0},
{"_id":"63525f9adf553215ed37dff2","title":"Hoodie","desc":"Mens
hoodie","size":"Medium","color":"Gray","price":60,"url":"hoodie.png","image":"https://imgbucket-nwen304.s3.ap-southeast-
2.amazonaws.com/hoodie.png","createdAt":"2022-10-21T09:00:10.661Z","updatedAt":"2022-10-21T09:00:10.661Z","__v":0},
{"_id":"63526137df553215ed37dffd","title":"Jeans","desc":"Men's denim
jeans","size":"42","color":"Blue","price":60,"url":"mensjeans.jpg","image":"https://imgbucket-nwen304.s3.ap-southeast-
2.amazonaws.com/mensjeans.jpg","createdAt":"2022-10-21T09:07:03.449Z","updatedAt":"2022-10-21T09:07:03.449Z","__v":0},
{"_id":"63525872d21e589e250d7fea","title":"T-Shirt","desc":"Definitely a t-
shirt","size":"Large","color":"Green","price":15,"url":"shirt.jpg","image":"https://imgbucket-nwen304.s3.ap-southeast-
2.amazonaws.com/shirt.jpg","createdAt":"2022-10-21T08:29:38.355Z","updatedAt":"2022-10-21T08:29:38.355Z","__v":0},
{"_id":"6352603fdf553215ed37dff7","title":"Shirt","desc":"Collared
shirt","size":"Large","color":"Blue","price":40,"url":"collarShirt.png","image":"https://imgbucket-nwen304.s3.ap-southeast-
2.amazonaws.com/collarShirt.png","createdAt":"2022-10-21T09:02:55.400Z","updatedAt":"2022-10-21T09:02:55.400Z","__v":0},
{"_id":"635301964240b0fb443d9ad4","title":"Jersey","desc":"Paul George Nike Icon
Edition","size":"Large","color":"Blue","price":125,"url":"pg13.png","image":"https://imgbucket-nwen304.s3.ap-southeast-
2.amazonaws.com/pg13.png","createdAt":"2022-10-21T20:31:18.905Z","updatedAt":"2022-10-21T20:31:18.905Z","__v":0},
{"_id":"6352122ac10a6f2ba789ba01","title":"Kawhi II","desc":"Kawhi Leonard basketball shoe","size":"12","color":"Gray Mist with Pale
Gold","price":200,"url":"Kawhi2.png","image":"https://imgbucket-nwen304.s3.ap-southeast-2.amazonaws.com/Kawhi2.png","createdAt":"2022-
10-21T03:29:46.868Z","updatedAt":"2022-10-21T03:29:46.868Z","__v":0},{"_id":"63530a90da9edfcff7cc857a","title":"LeBron
20","desc":"LeBron James basketball shoe","size":"13","color":"Purple","price":240,"url":"LeBron.png","image":"https://imgbucket-
nwen304.s3.ap-southeast-2.amazonaws.com/LeBron.png","createdAt":"2022-10-21T21:09:36.131Z","updatedAt":"2022-10-
21T21:09:36.131Z","__v":0},{"_id":"63530747e7e8b334751d7f2e","title":"PG 6","desc":"Paul George basketball
shoe","size":"11","color":"White, Black","price":190,"url":"pg6.png","image":"https://imgbucket-nwen304.s3.ap-southeast-
2.amazonaws.com/pg6.png","createdAt":"2022-10-21T20:55:35.959Z","updatedAt":"2022-10-21T20:55:35.959Z","__v":0},
{"_id":"63520aa59b0d0ca552f54608","title":"Beanie","desc":"Beanie","size":"M","color":"Blue","price":15,"url":"Beanie.png","image":"ht
tps://imgbucket-nwen304.s3.ap-southeast-2.amazonaws.com/Beanie.png","id":"63520aa59b0d0ca552f54608"}]
```

We then can see the items successfully added to MongoDB both in MongoDB compass Under the items folder or through a MongoDB cluster.



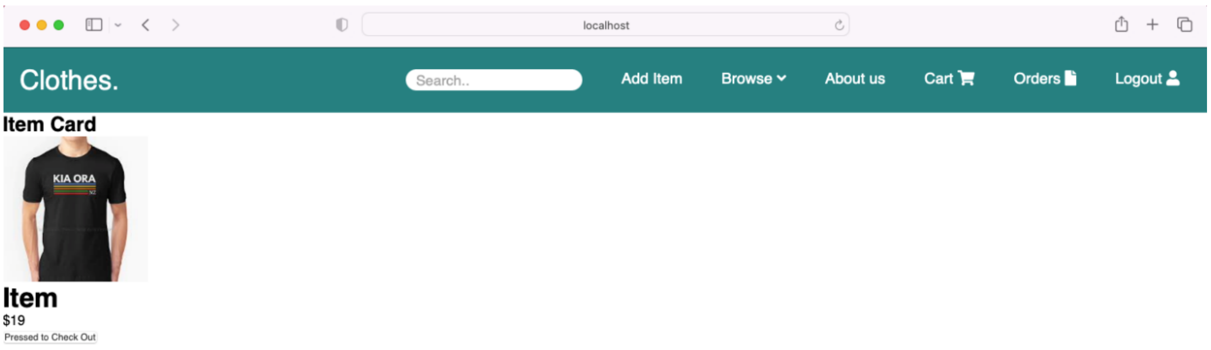
```
_id: ObjectId('63520aa59b0d0ca552f54608')
title: "Beanie"
desc: "Beanie"
size: "M"
color: "Blue"
price: "15"
url: "Beanie.png"
image: "https://imgbucket-nwen304.s3.ap-southeast-2.amazonaws.com/Beanie.png"
id: "63520aa59b0d0ca552f54608"

_id: ObjectId('6352122ac10a6f2ba789ba01')
title: "Kawhi II"
desc: "Kawhi Leonard basketball shoe"
size: "12"
color: "Gray Mist with Pale Gold"
price: 200
url: "Kawhi2.png"
image: "https://imgbucket-nwen304.s3.ap-southeast-2.amazonaws.com/Kawhi2.png"
createdAt: 2022-10-21T03:29:46.868+00:00
updatedAt: 2022-10-21T03:29:46.868+00:00
__v: 0

_id: ObjectId('63525872d21e589e250d7fea')
title: "T-Shirt"
desc: "Definitely a t-shirt"
size: "Large"
color: "Green"
price: 15
url: "shirt.jpg"
image: "https://imgbucket-nwen304.s3.ap-southeast-2.amazonaws.com/shirt.jpg"
createdAt: 2022-10-21T08:29:38.355+00:00
updatedAt: 2022-10-21T08:29:38.355+00:00
__v: 0
```

We can also read the users id without a GUI, by using: http://localhost:3000/api/users



```
[{"local":
{"username":"fklsfjkls","password":"$2b$10$jShyUJVTgywX2kIjQQjuKOUgBx0FJeX5kuXP6IrbcJKOZ6YPBueTq","email":"kitt@crichq.com"},"_id":"63
4f2bb21ac95e4e15a32ecb","__v":0},{"local":
{"username":"kittka12","password":"$2b$10$tluvuUEN80LDiTgTIGSmCuUgXCg0PZynQZmQvtTiedspgvvu0iKkO","email":"kitt@outlook.com"},"_id":"63
4f034a4c4e3b79d8ad821b","__v":0},{"_id":"635208074cf77eab4ffa2a0a","__v":0},{"local":
{"username":"test","password":"$2b$10$hq1h2cnVhb4iweZ78yCybO4idUNcgCnFwUbKh4rmJ1pqXXl4LyFC.","email":"test@mail.com"},"_id":"6343d2f2f
bf98c48ead1dd29","__v":0},{"_id":"635204245b1b9f5a3f368386","__v":0},{"local":
{"username":"kitt123","password":"$2b$10$/Kw7Fg.F7jL9El3/qFmnvOrNadfCfl9jJqFIVBCkBnjIxDUM2oTcm","email":"kitt@crichq.com"},"_id":"634f
029cd0cc97ad15889c61","__v":0},{"_id":"63520ea9accf9737dc1ef26f","__v":0},{"_id":"635200be674d46741a54ad5a","__v":0},
{"_id":"63531eeaf1b5fd5a5e543726","__v":0},{"google":{"id":"107426313658089477989","username":"Hayden
Pomare","email":"haydencpomare@gmail.com"},"_id":"633b6a78ac46295edc807930","__v":0},{"local":
{"username":"hfjsidhfsj","password":"$2b$10$plSbHRYffnuvMnPCKHYXcO9TmgUG6QC40vBl3Aqu1fH1rYceuZPUe","email":"kitt@crichq.com"},"_id":"6
34f34492ea2ccc01f3f9876","__v":0},{"local":
{"username":"steve","password":"$2b$10$PGDrcGrQfxGNybVPxdxLeOgAr6WKjzyPgcGfYBKxWUe82fsMeV1Wu","email":"steve@stevemail.com"},"_id":"63
52fedcec0f6186d1e53879","__v":0},{"local":
{"username":"km123","password":"$2b$10$68a0p5HKI/1PX41j90x.seYJs4IU.dC3kw0OD4RAJ3/boYXaqEBSe","email":"kitt.mcevoy@outlook.com"},"_id"
:"634dd31d3d453e1b20a7567a","__v":0}]
```

A logged in user can successfully visit a Cart page where they can see the item card with some details such as price and a button to pressed to check out



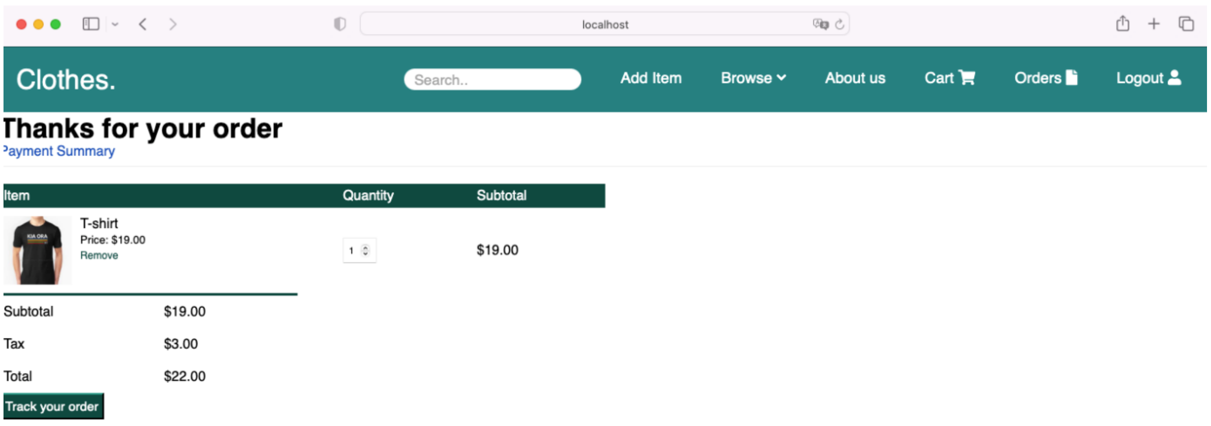
Item Card

Item

$19

Pressed to Check Out

A logged in user can successfully visit an Orders page where they can see the payments summary details display and a button to track their order.



Thanks for your order

Payment Summary

Item	Quantity	Subtotal
T-shirt Price: $19.00 Remove	1	$19.00

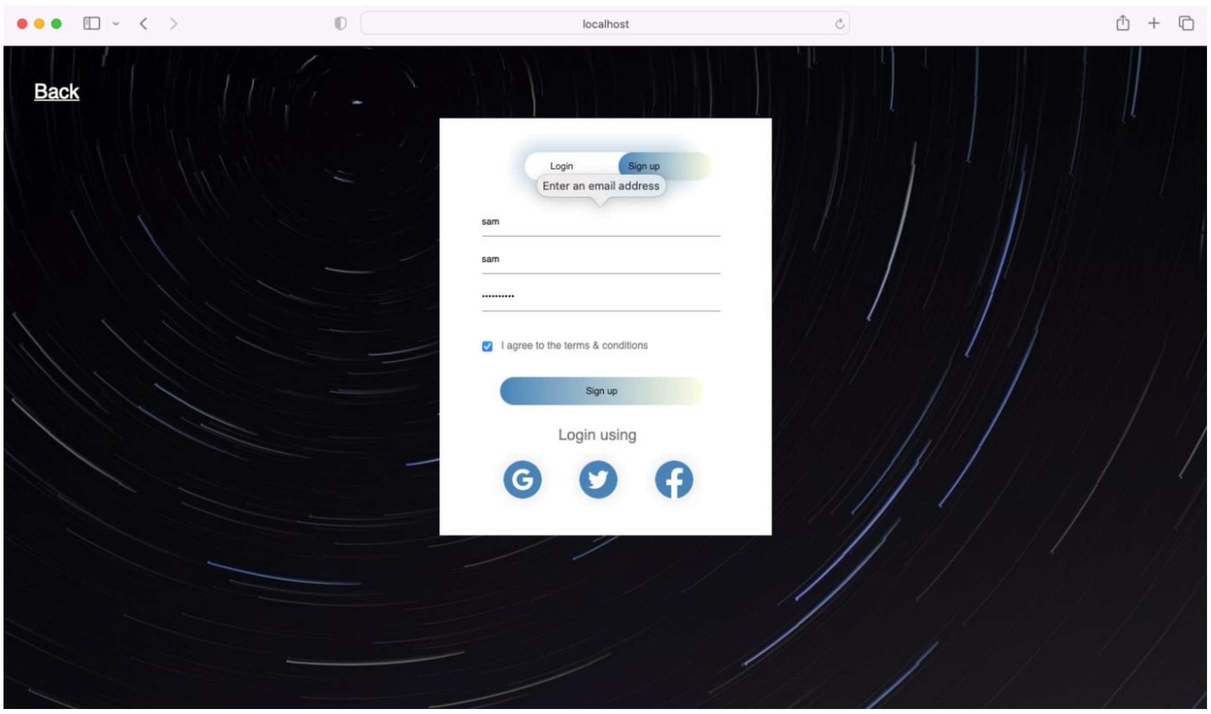
Subtotal	$19.00
Tax	$3.00
Total	$22.00

Track your order

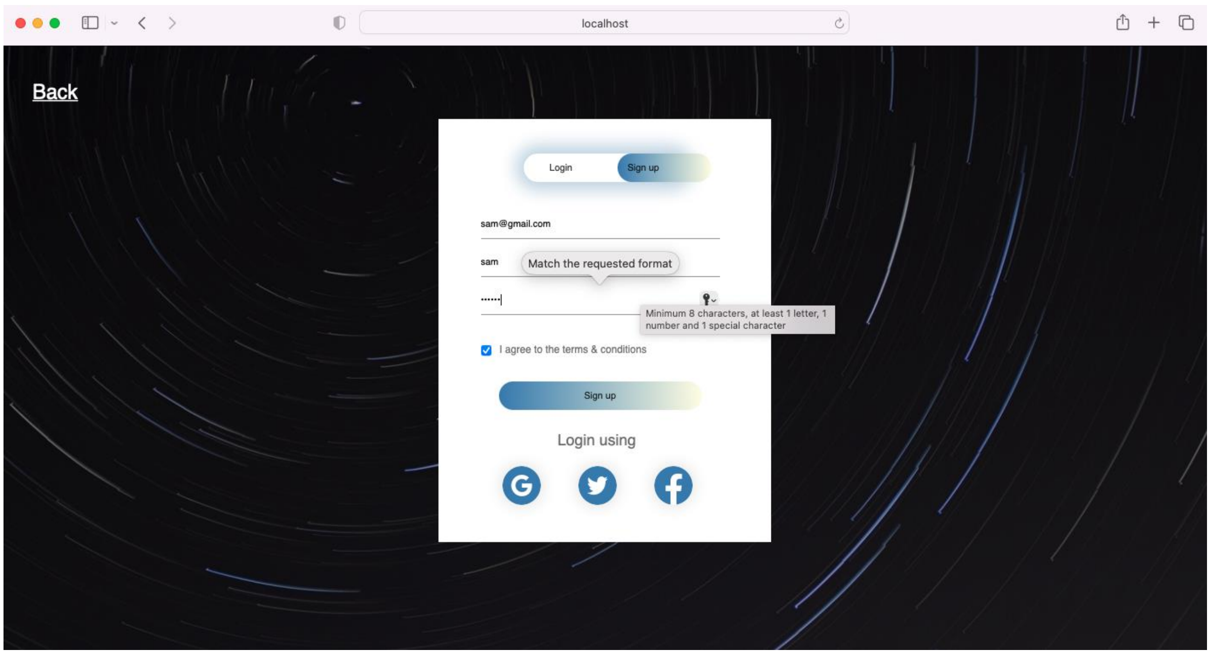
4.0 Error handling

Login/ signup

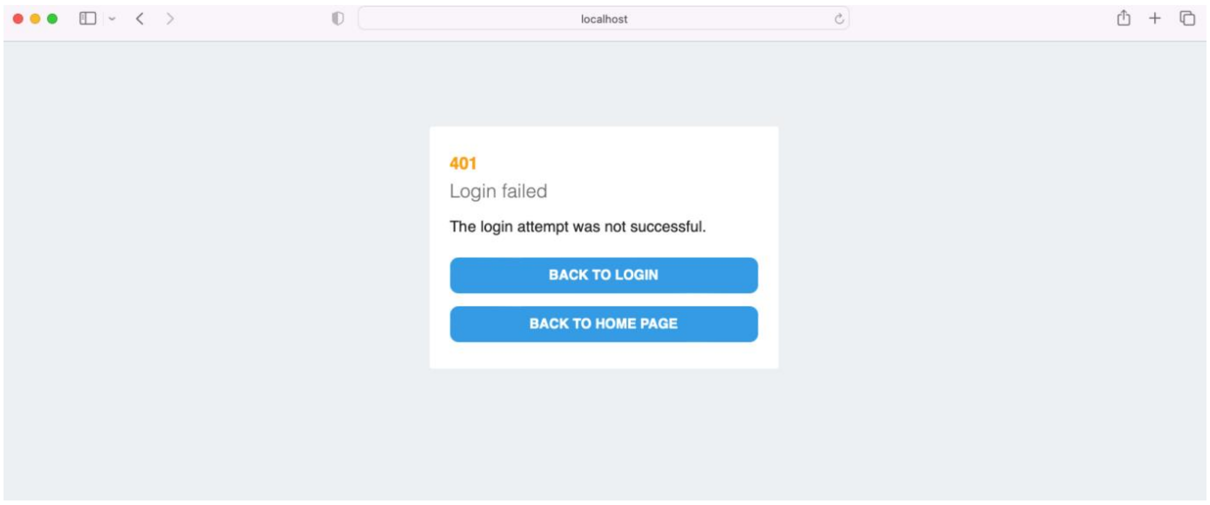
We have a number of different error states that can handle when it comes to both login and signup. First it checks for a valid email address. If a user tries to enter a name in state of email address a message tells the user to enter an email address.



A Second check to see if a password matches the request format with minimum 8 characters, at least 1 letter, 1 number and 1 special character.

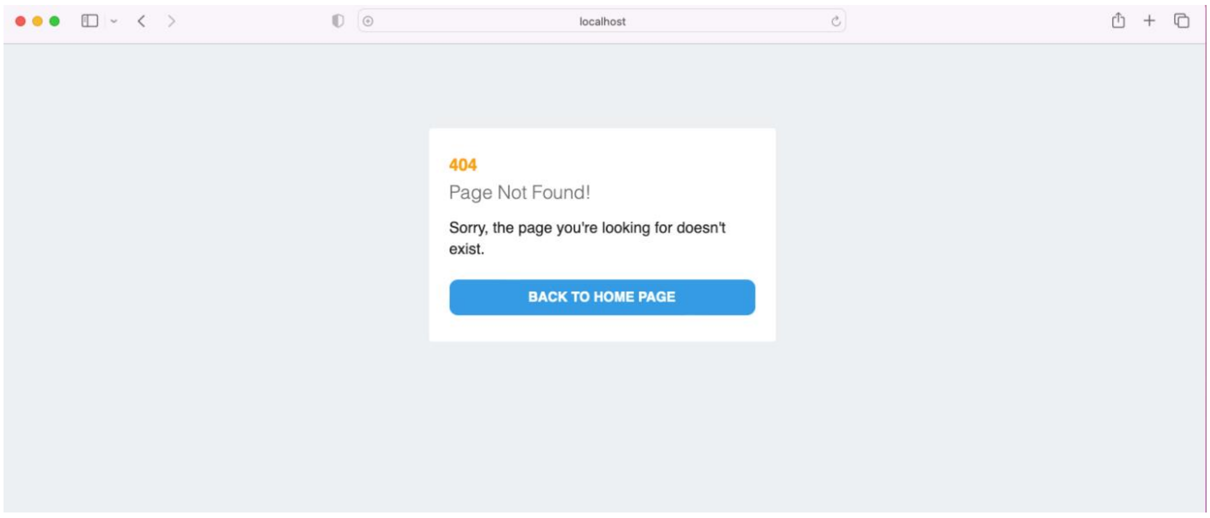

localhost
Back
Login
Sign up
sam@gmail.com
sam
Match the requested format
Minimum 8 characters, at least 1 letter, 1 number and 1 special character
I agree to the terms & conditions
Sign up
Login using

Third check, if a user login with wrong information of username or password it gets directed to a 401 page with a login failed message.

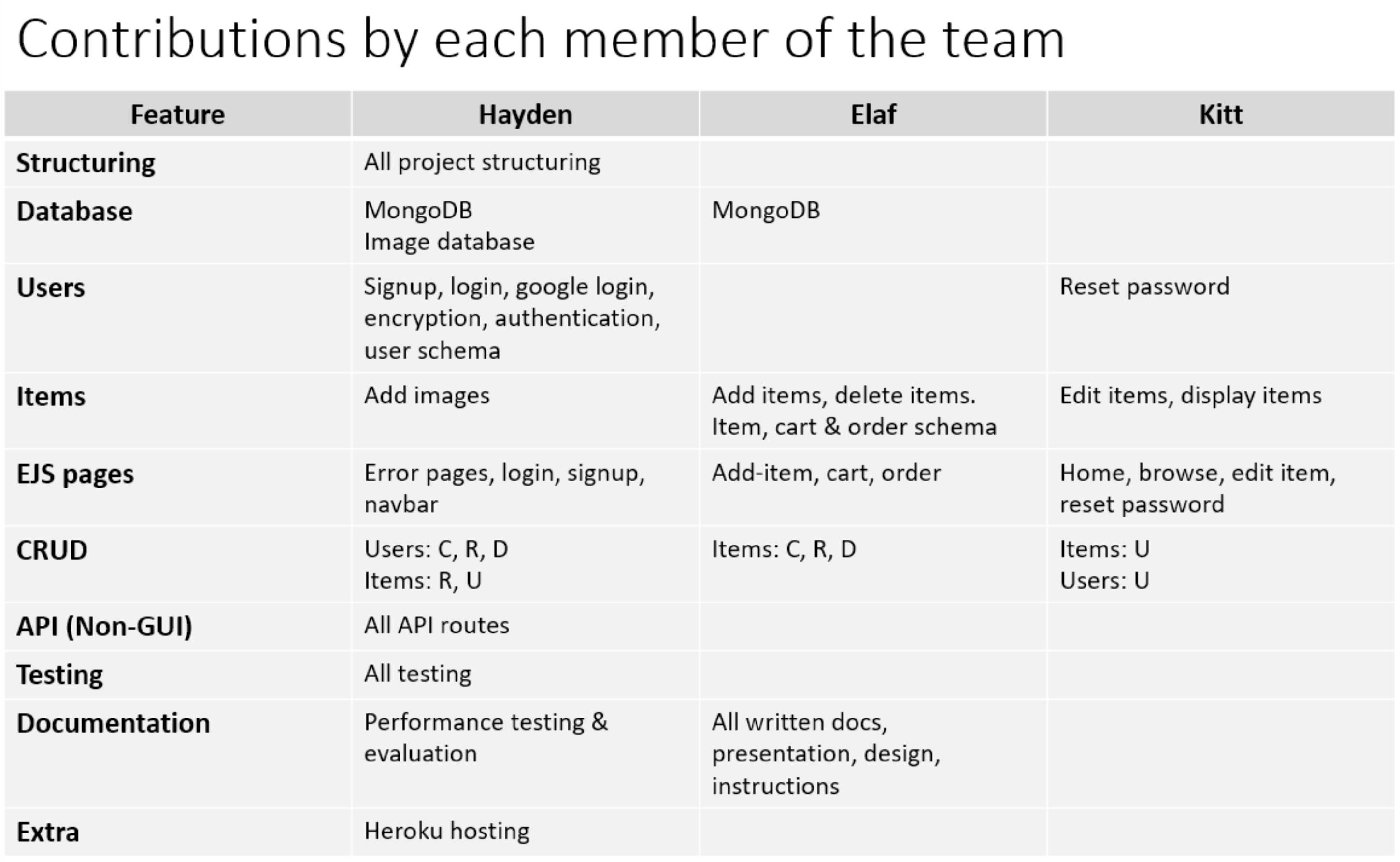


The login attempt was not successful.

Fourth check, if a user tries to access a page that is not there with wrong API address it gets directed to a 404 with Page Not Found message.



Sorry, the page you're looking for doesn't exist.



Contributions by each member of the team

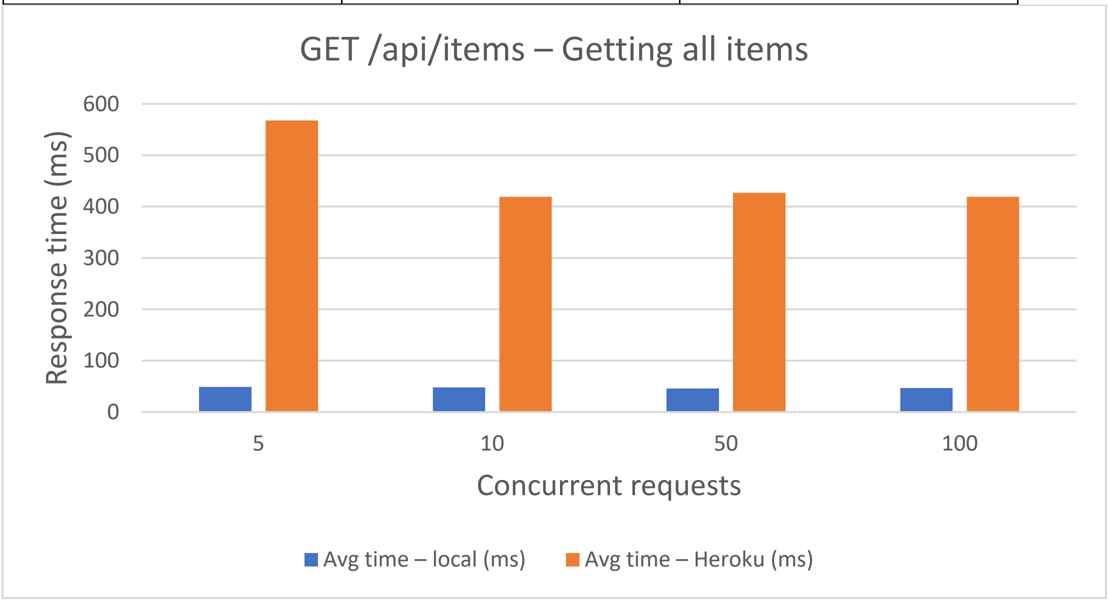
Feature	Hayden	Elaf	Kitt
Structuring	All project structuring		
Database	MongoDB Image database	MongoDB	
Users	Signup, login, google login, encryption, authentication, user schema		Reset password
Items	Add images	Add items, delete items. Item, cart & order schema	Edit items, display items
EJS pages	Error pages, login, signup, navbar	Add-item, cart, order	Home, browse, edit item, reset password
CRUD	Users: C, R, D Items: R, U	Items: C, R, D	Items: U Users: U
API (Non-GUI)	All API routes		
Testing	All testing		
Documentation	Performance testing & evaluation	All written docs, presentation, design, instructions	
Extra	Heroku hosting		

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

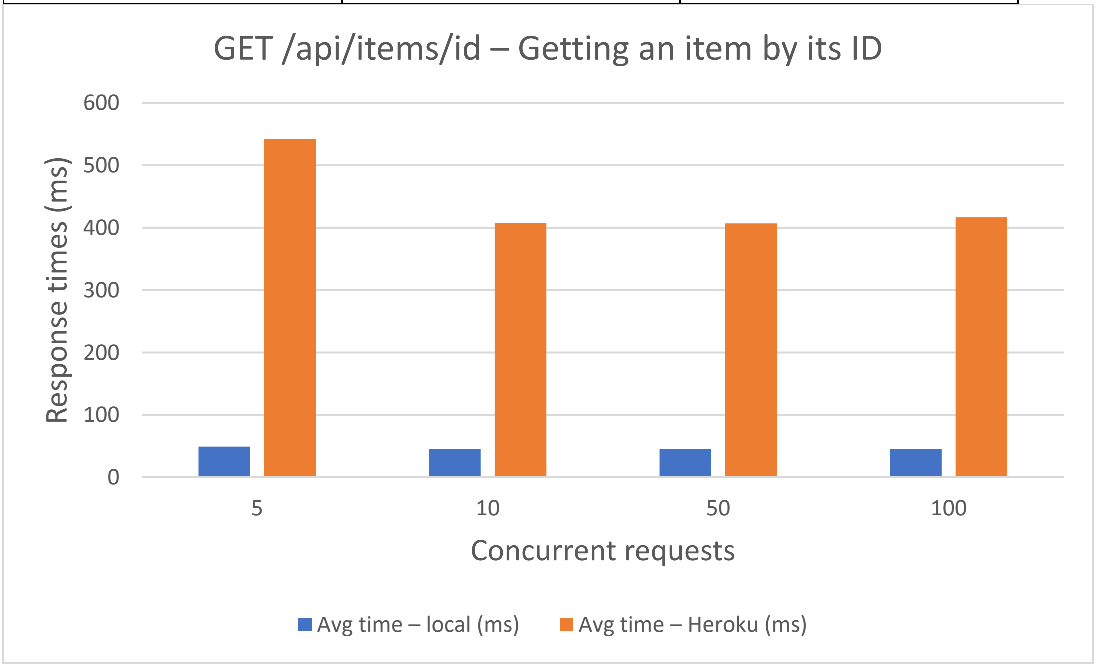
GET /api/items – Getting all items

Concurrent requests	Avg time – local (ms)	Avg time – Heroku (ms)
5	49	567.6
10	48	419.0
50	46	426.8
100	47	418.9



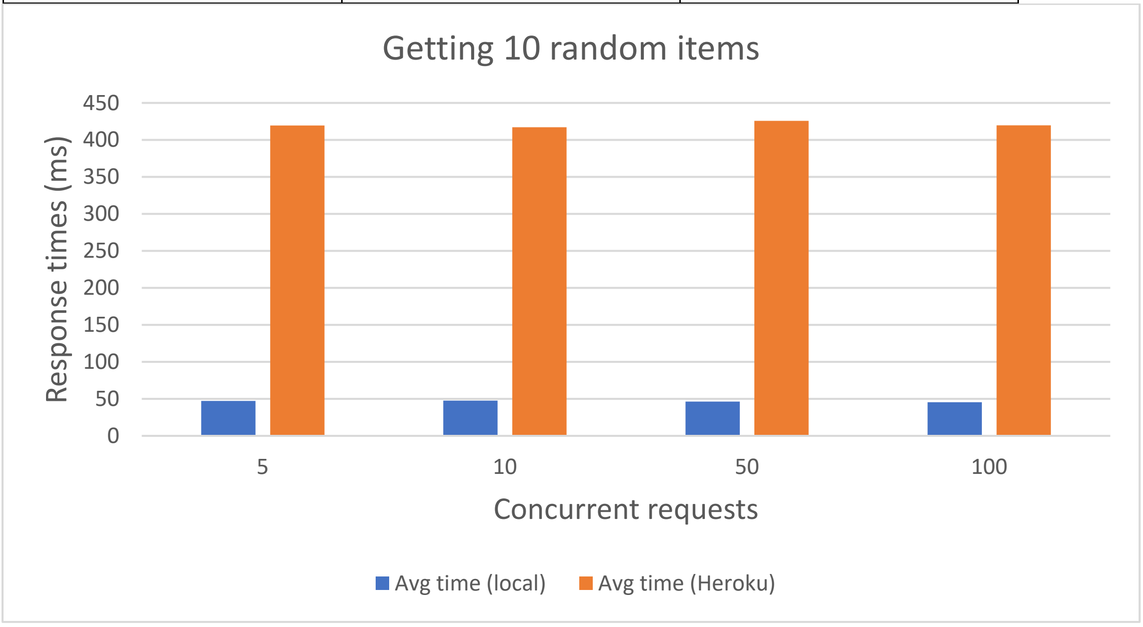
GET /api/items/id – Getting an item by its ID

Concurrent requests	Avg time – local (ms)	Avg time – Heroku (ms)
5	49.0	542.6
10	45.4	407.5
50	45.1	406.8
100	45.0	416.7
1000	44.8	



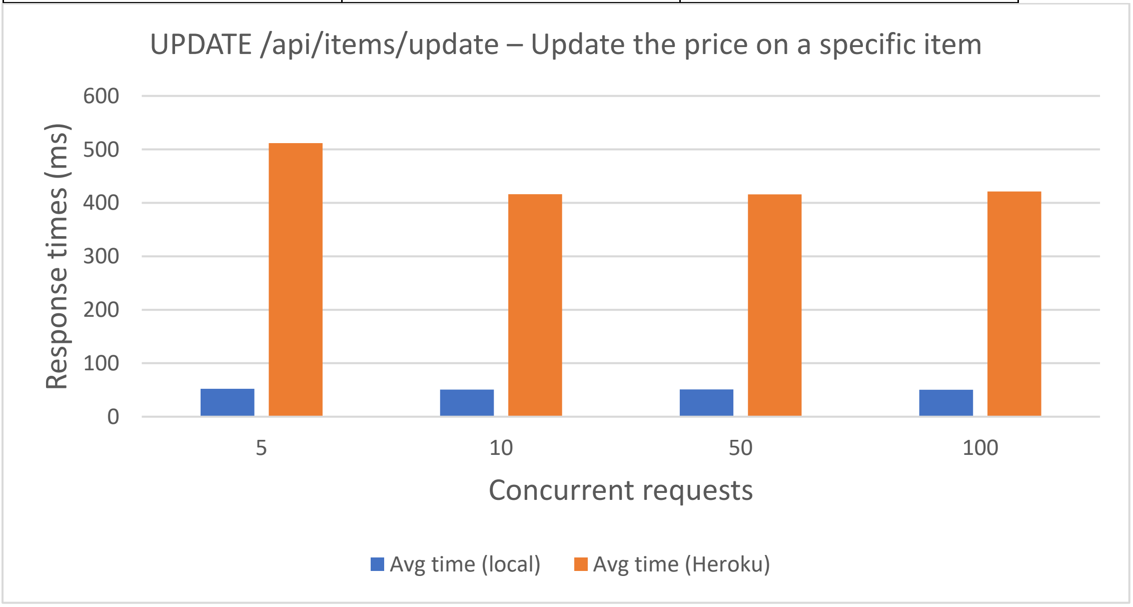
GET /api/items/10 – Getting 10 random items

Concurrent requests	Avg time (local)	Avg time (Heroku)
5	47.2	419.4
10	47.7	417.1
50	46.3	425.66
100	45.5	419.8



UPDATE /api/items/update – Update the price on a specific item

Concurrent requests	Avg time (local)	Avg time (Heroku)
5	52.2	511.8
10	50.8	416.2
50	51.0	416
100	50.5	421.12



CREATE /local-signup – Create a new user (Override matching username check)

Concurrent requests	Avg time (local)	Avg time (Heroku)
5	92.2	
10	89.9	
50	90.04	
100	89.45	

Section d - Performance Evaluation.

For this project we used MongoDB as our database. This is connected to and interacted with using our RESTful web application services. For the most part, as can be seen in the performance charts we developed using localhost with little to no wait time from the database. Items would already be loaded when the page loads.

When testing the performance with concurrent users connecting to the database, there was little to no bottlenecking. Each user had roughly the same response time. This was pushed the most when running over 100 requests, at which point it became so long in some cases there was no time to finish the testing. But this was rare and most experienced fine connections.

One clear drawback to performance is that due to us using a free version of MongoDB, the database uses a shared cluster. This means it scales horizontally over multiple servers with sharding. As opposed to a replica database which would ensure higher availability and better performance.

The database can be expected to cause minor bottlenecking. It was unnoticeable during our development and performed well during our testing. If the need were to ever arise that better performance was required, there are options within MongoDB and elsewhere which would allow that to be seamless.