

Ecommerce - Phase 1



Program:

Course Code: CSE411

Course Name: Distributed Computer Systems

Examination Committee

Prof. Gamal Abdel Shafy Ebrahim

**Ain Shams University
Faculty of Engineering
Fall Semester – 2021**



Student Personal Information

1700923	عمرو ايهاب عبدالعزيز محمد عبدالعزيز
1701040	لؤي انور عبدالرازق حجازي
1701048	ليلي وائل سيد عبدالمنعم
1701051	لينة ايمن محمود



Contents

Introduction	4
Target beneficiaries of the project	4
Adopted programming languages	4
System Architecture	5
Application-Level Protocol	6
Distributed Database Design	9
Testing	12
End-user guide	13
Resources Needed	21
Role of Each Member	22
References	23



Introduction

Our system is a distributed online marketplace that sellers can use to offer their products and get their profits. Sellers can browse all users' products, view details of each product, and view all other users and stores. Sellers can view their stores, manage their products, and add other sellers' products to his store to offer it. Sellers also can order and purchase other sellers' products and provide their info to ship these products to him. Sellers can search for products for sale by other sellers and search for other sellers using username to browse their stores.

Each seller has to sign up and log into our system to benefit from our provided services and view his account to check his personal info, charge balance to purchase new products from other sellers, view his purchase history with the purchased items, and view his sold history with the sold items. In the users' store, User can edit their products info to add more details or change the product image or price, add new product to his store to sell it later, and delete a specific product if he wants. Also, each seller can remove the other sellers' products from his store or purchase it to edit, delete and sell into his store.

When the user purchases product from another user's store, the money is transferred to the seller's balance and the purchased product is transferred to the buyer store.

Target beneficiaries of the project

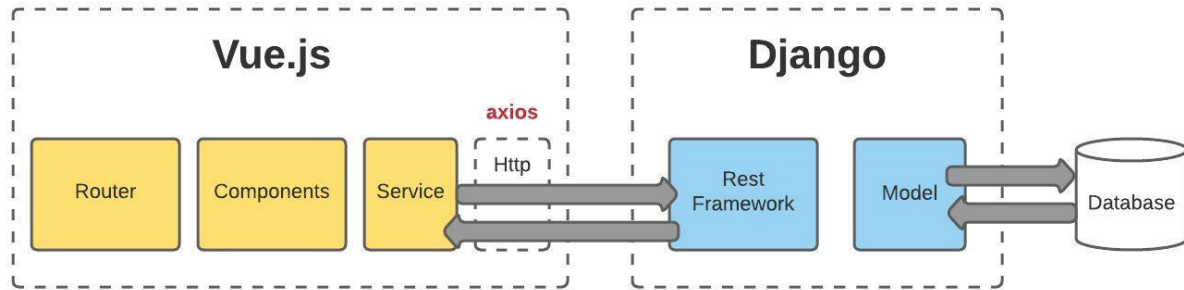
One of our marketplace system goals is to make the marketing of products much easier which helps small sellers to establish their business and get their first customers. We also aim to facilitate the communication among users to let them gain more experience dealing with other experienced sellers.

Customers can order products easily and support other junior sellers to pay attention or sell their own products and become our new client.

Adopted programming languages

We used Django REST framework to build the API and Vue JS to build the front end of the project.

System Architecture



We choose Django for our system which uses the Client-Server and MVC Architectures since it supports

1. Rapid Development

Actually, this Django architecture that separates in different components makes it easy for multiple developers to work on different aspects of the same application simultaneously. That is also one of the features of Django.

2. Loosely Coupled Components

This architecture of Django has different components which require each other at certain parts of the application, at every instant, that increases the security of the overall website. As the model file will now only save on our server rather than saving on the webpage.

3. Ease of Modification

This is an important aspect of development as there are different components in Django architecture. If there is a change in different components, we don't have to change it in other components.

This is actually one of the special features of Django, as here it provides us with much more adaptability of our website than other frameworks.

Application-Level Protocol

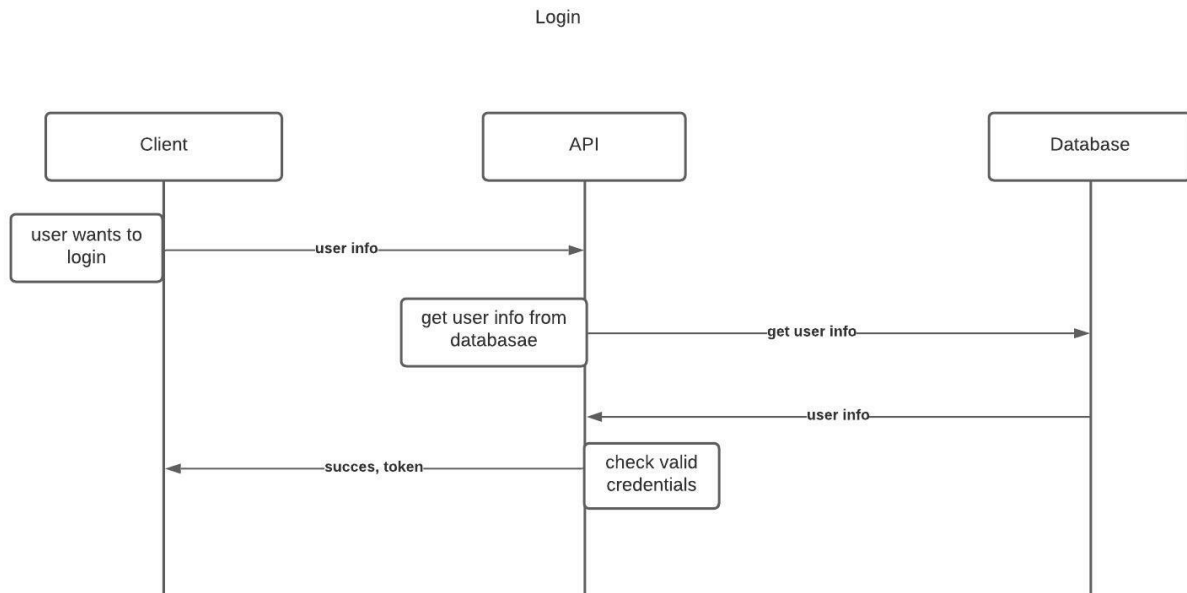


Figure 1 Login Scenario

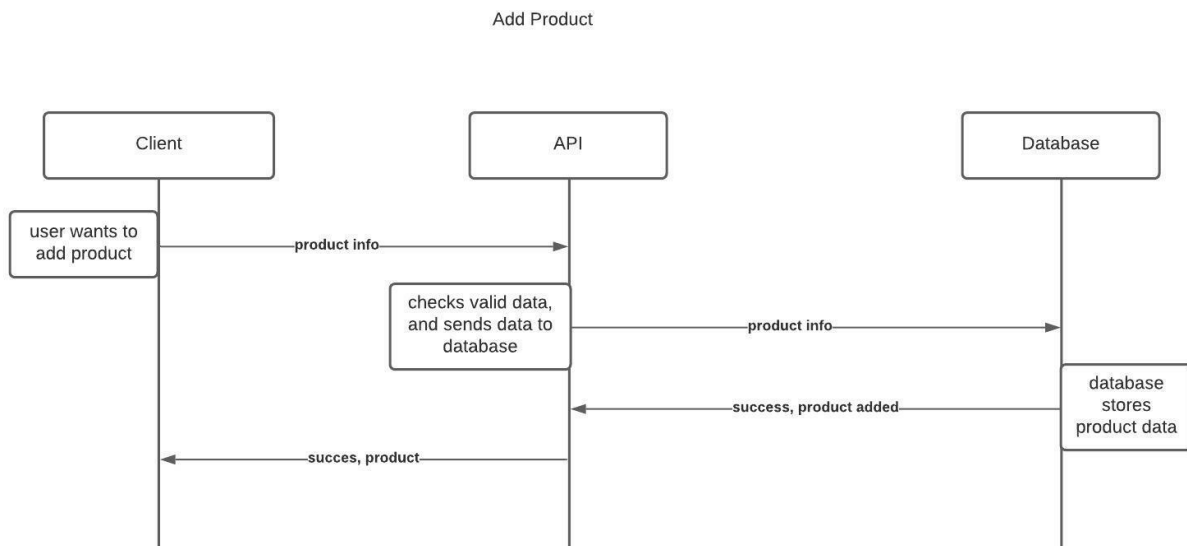


Figure 2 Add product scenario

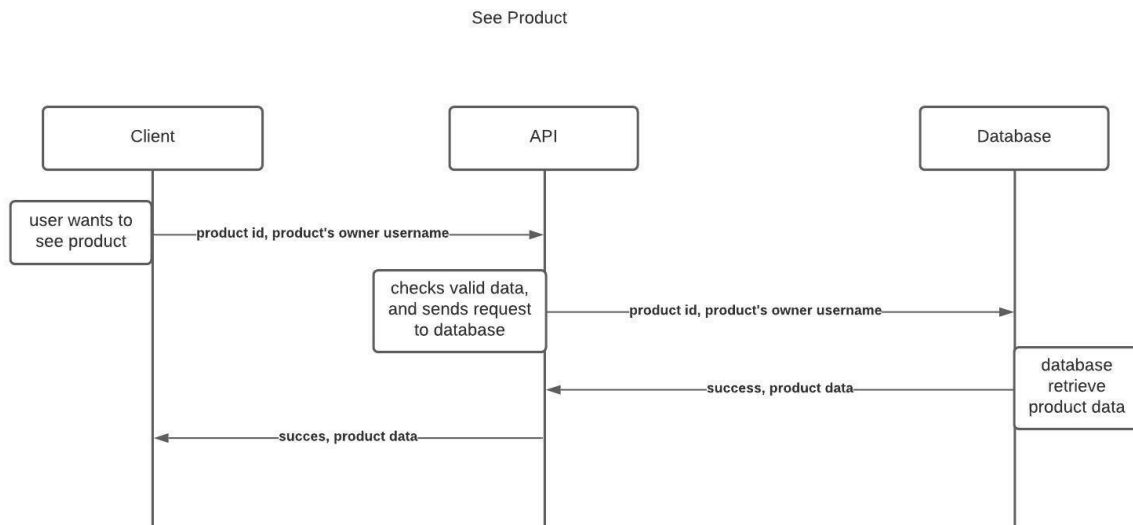


Figure 3 Get product details scenario

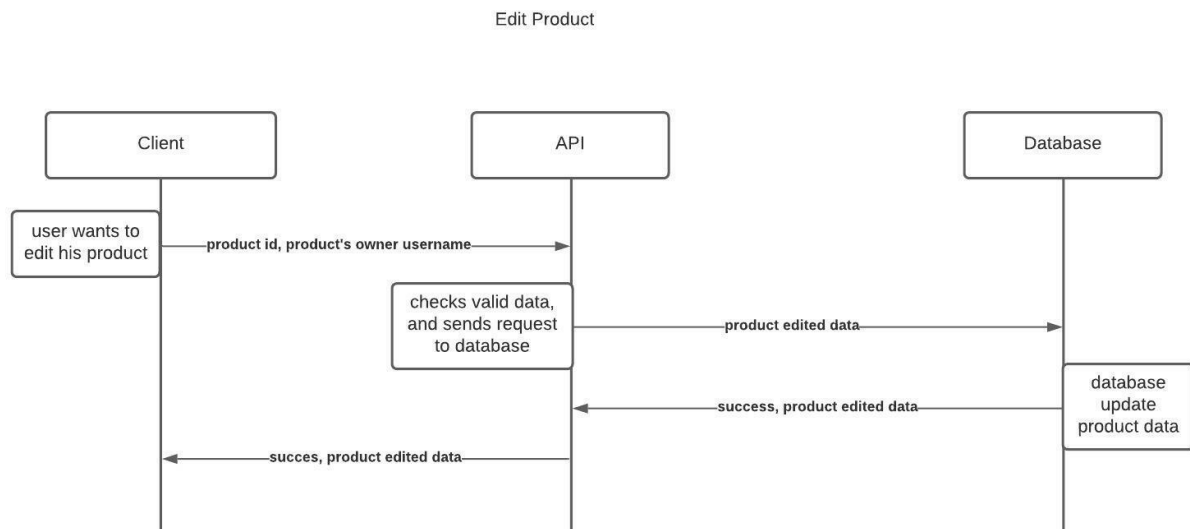


Figure 4 User editing their own product scenario

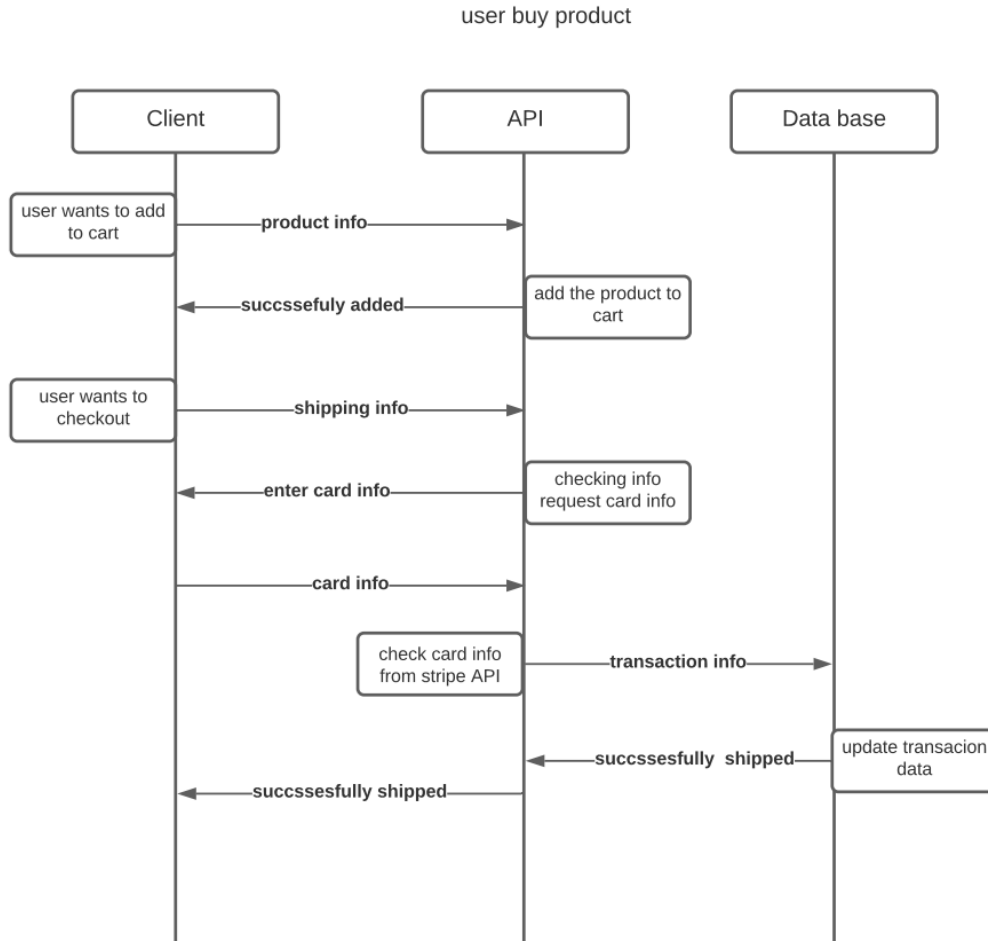


Figure 5 User buying a product scenario

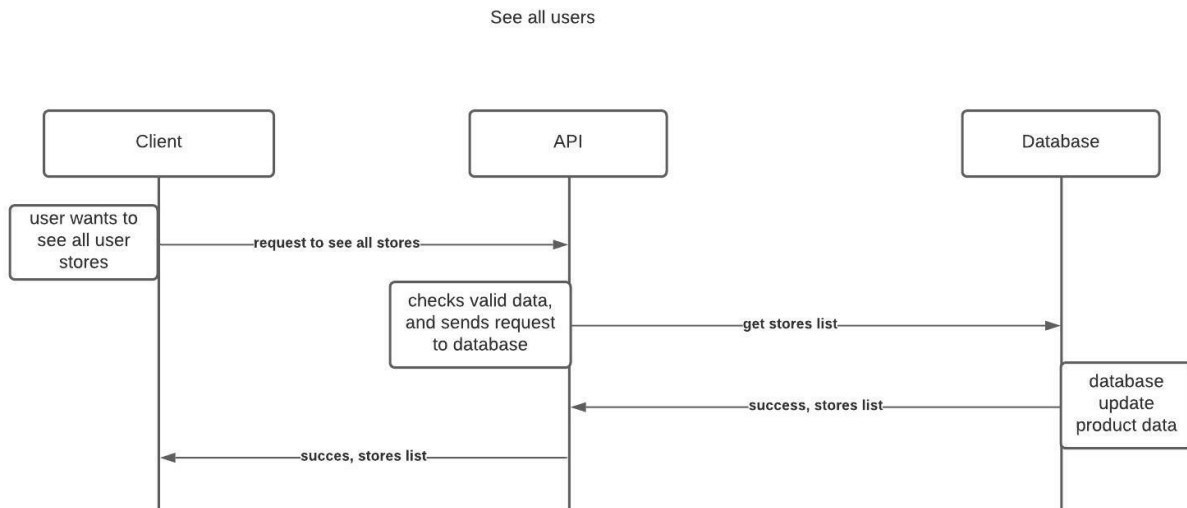


Figure 6 User browsing all stores list

Distributed Database Design

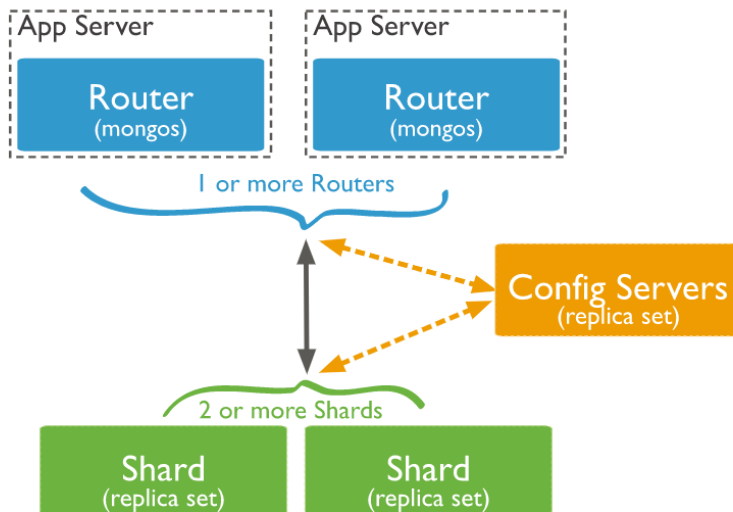
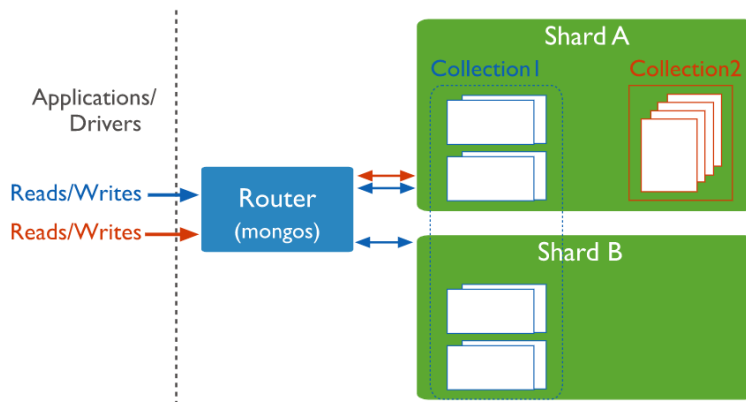
We are using MongoDB which doesn't enforce a schema on the database.

The database consists of collections which are stored in different "shards". The database decides on which shard to place the record based on a chosen shard key for every collection.

In MongoDB when sharding is enabled, the collections are fragmented horizontally

The application connects to the different shards through a router which uses the info in the config server to find the records.

We used 2 shards and 2 replicas for every shard.





COLLECTION ATTRIBUTES

MYUSER	first_name last_name email username not_owned_products many to many relation with a Product balance date_joined last_login is_admin is_staff is_superuser
PRODUCT	owner -> foreign key to a MyUser category name slug description price image thumbnail date_added
SOLDPRODUCT	seller -> foreign key to a MyUser buyer -> foreign key to a MyUser category name slug description price image thumbnail date_added



ORDER

user -> **foreign key** to a MyUser
first_name
last_name
email
address
place
phone
created_at
paid_amount

ORDERITEM

Order -> foreign key to an Order
Price

Testing

API Testing:

We used automated tests for views, models and urls in Django tests. We used postman to test all of the API endpoints.

```
(env) PS C:\Users\Amr\Documents\Github\E-Commerce-Website\ecommerce> python manage.py test
Creating test database for alias 'default'...
This version of django does not support "NULL, NOT NULL column validation check" fully. Visit https://nesdis.github.io/django/support/
This version of django does not support "schema validation using CONSTRAINT" fully. Visit https://nesdis.github.io/django/support/
This version of django does not support "schema validation using KEY" fully. Visit https://nesdis.github.io/django/support/
This version of django does not support "schema validation using REFERENCES" fully. Visit https://nesdis.github.io/django/support/
This version of django does not support "COLUMN DROP NOT NULL " fully. Visit https://nesdis.github.io/django/support/
This version of django does not support "DROP CASCADE" fully. Visit https://nesdis.github.io/django/support/
System check identified no issues (0 silenced).
.....
Ran 7 tests in 8.388s

OK
Destroying test database for alias 'default'...
(env) PS C:\Users\Amr\Documents\Github\E-Commerce-Website\ecommerce>
```

Figure 7 API Automated Testing

Sample of postman API end points testing

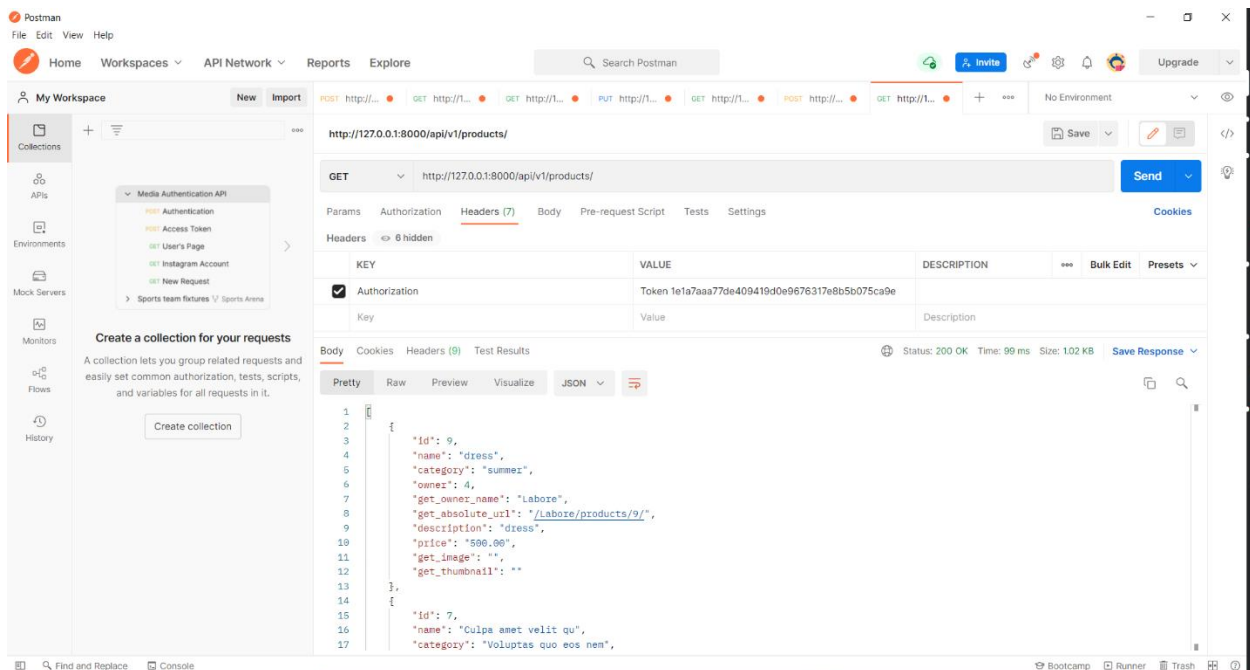


Figure 8 postman testing



End-user guide

1. Signup:

First user needs to create an account with valid data.

Sign up

First name

Last name

Username

Email

Password

Confirm password

2. Login:

Login into your account with the registered data.

Log in

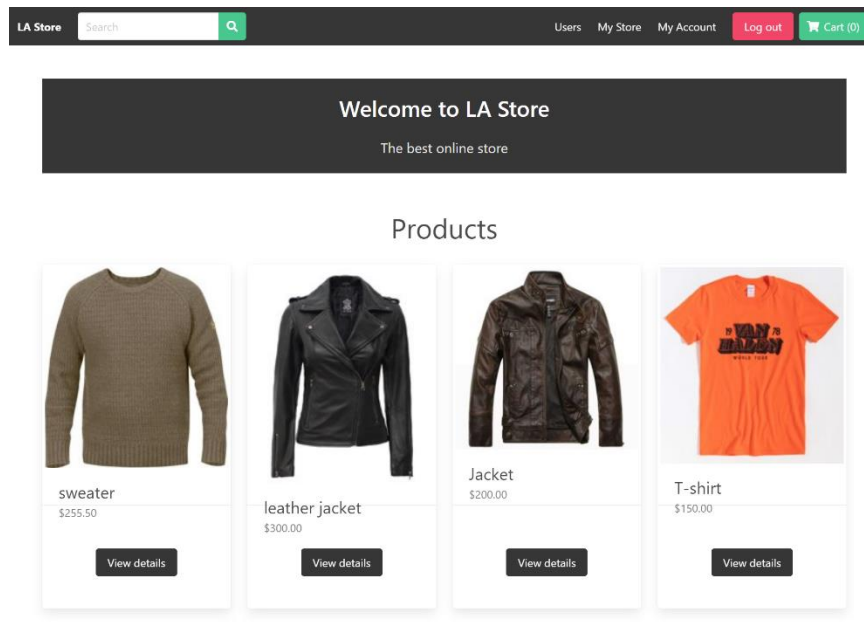
Email

Password

Or [click here](#) to sign up!

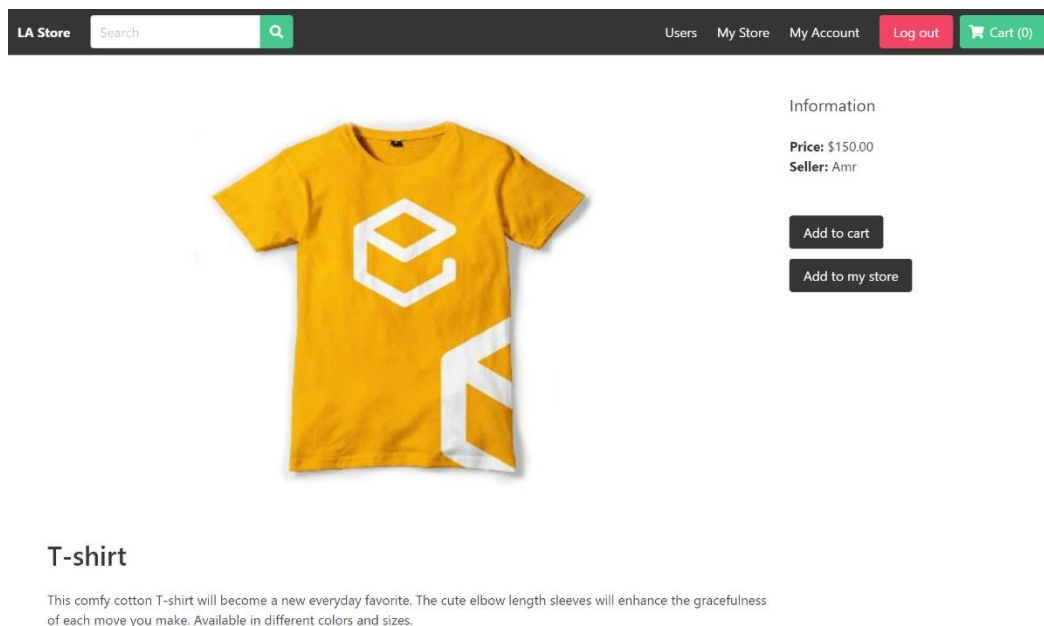
3. Home Page:

Your home page that contains all the products that are sold by other users, where you can view any products to see more details.



4. Product Page:

In the view details page for each product, you will be able to choose whether to add the product to your card to buy it or add it to your store to display it inside your store page.





Cart Page:

Your cart will contain all the products you want to buy and when you are done adding products to your card you can go to checkout.

LA Store

Users

My Store

My Account

Log out

Cart (3)

Cart

Product	Price	Quantity	Total	
leather jacket	\$300.00	1	\$300.00	
T-shirt	\$150.00	1	\$150.00	
TV	\$8000.00	1	\$8000.00	

Summary

\$8450.00, 3 items

Proceed to checkout

5. Checkout Page:

You need to enter a valid shipping info and click submit to successfully buy the products and it will appear in your store and will be delivered to the shipping address so you can sell it in your store.

LA Store

Users

My Store

My Account

Log out

Cart (3)

Checkout

Product	Price	Quantity	Total
leather jacket	\$300.00	1	\$300.00
T-shirt	\$150.00	1	\$150.00
TV	\$8000.00	1	\$8000.00
Total		3	\$8450.00

Shipping details

* All fields are required

First name*

loay

Address*

new cairo

Last name*

anwar

Country*

egypt

E-mail*

loay@gmail.com

Phone*

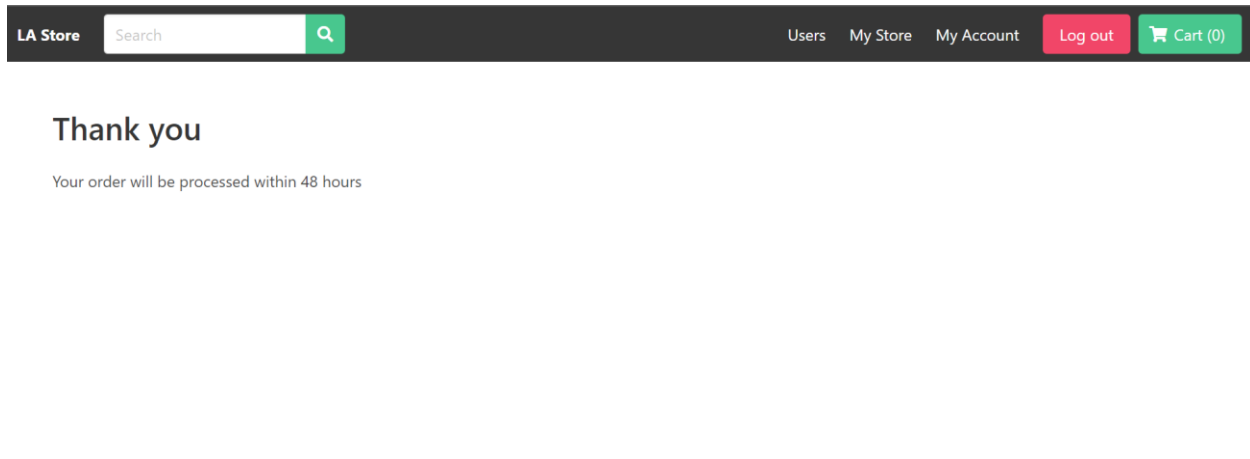
01927829136

Submit



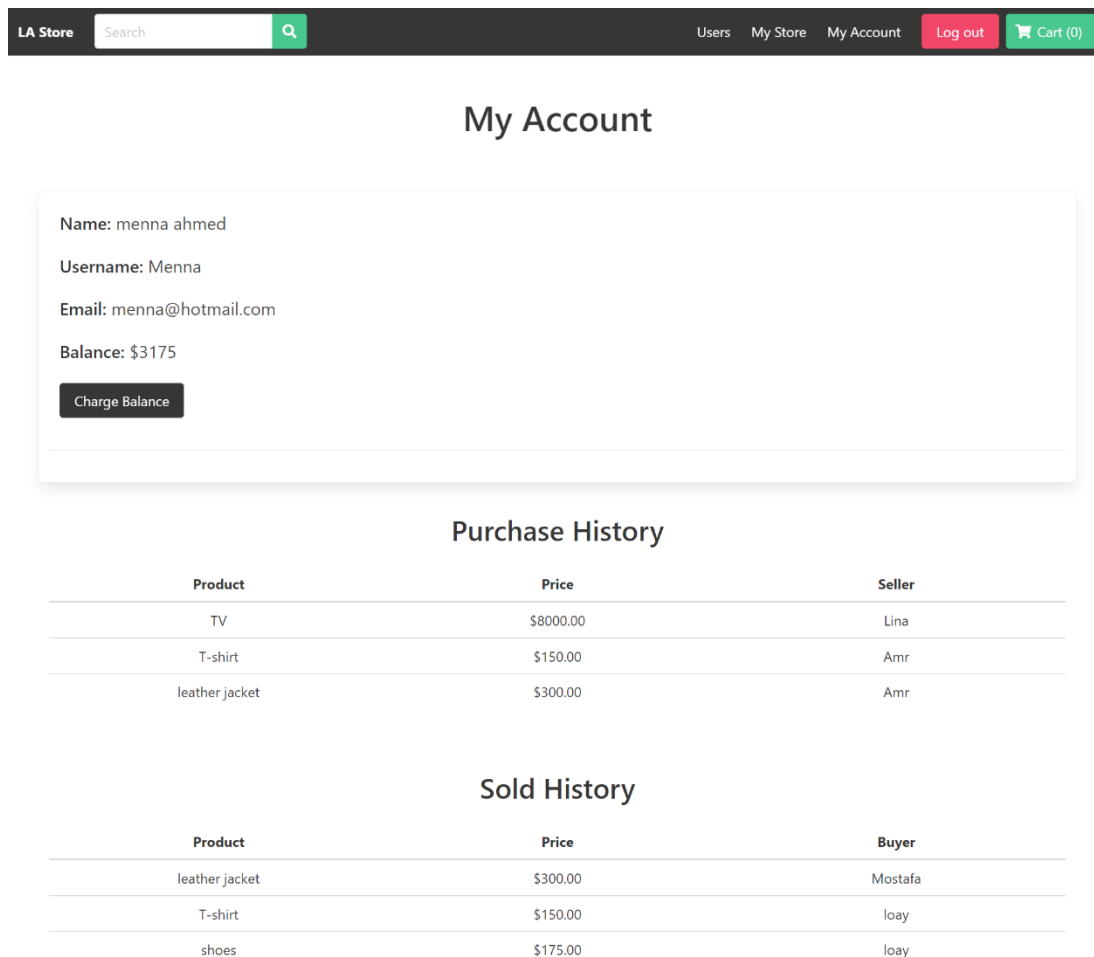
6. Success Page:

This page will appear after you buy a product to your store.



7. My Account Page:

My account page contains your personal info. and a button to charge your balance as well as your purchased history and sold history.





8. Charge Balance:
You can charge your balance using a valid card info.

LA Store

Users

My Store

My Account

Log out

Cart (2)

Charge Balance

Enter Amount

4242 4242 4242 4242

08 / 22 1234

Pay with stripe

9. Users Page:
Users page contains all other users stores where you can view any store to see their products and buy it or add it to your store to be sold.

LA Store

Users

My Store

My Account

Log out

Cart (0)

Users

Lina

lina ayman
lina.am99@hotmail.com

View Store

Laila

laila wael
laila@hotmail.com

View Store

loay

loay anwar
loay@gmail.com

View Store

Amr

amr ehab
amr@yahoo.com

View Store

Mostafa

Mostafa mohamed
mostafa@yahoo.com

View Store

Menna

menna ahmed
menna@hotmail.com

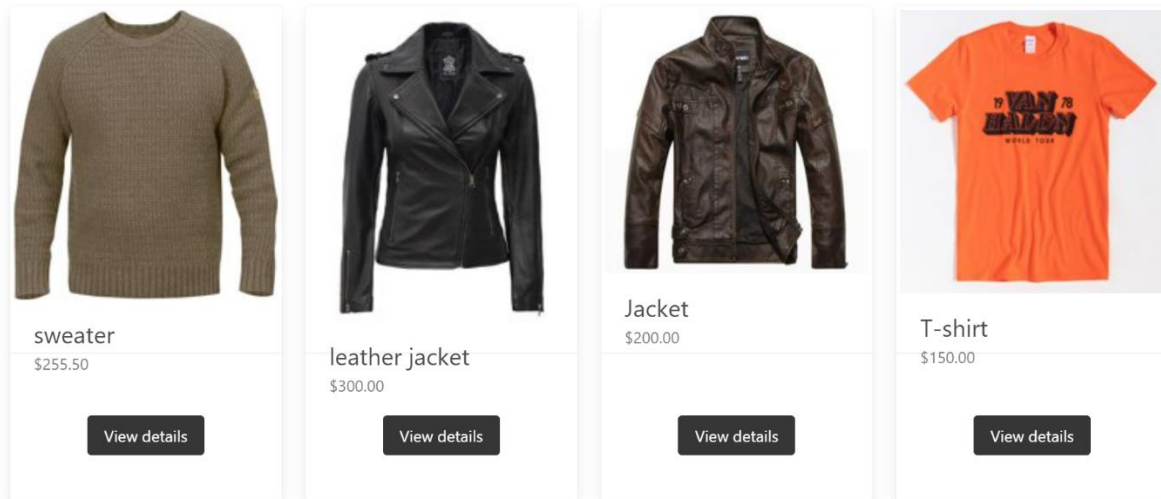
View Store

10. User Store Page:

In view store page you can see all the products sold by this user.



Amr Store

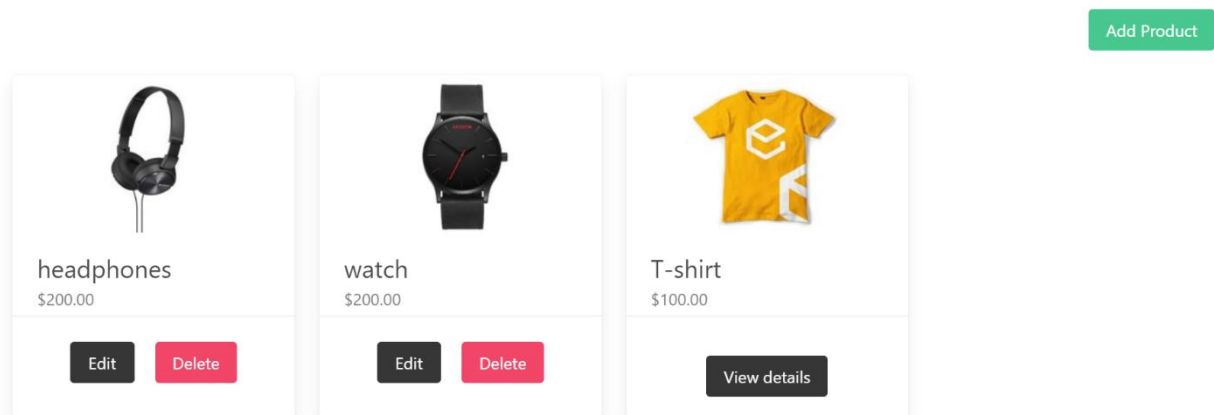


11. My Store:

My store page contains all the products that you sell where you can edit or delete them, and all products you sell for other users where you can buy them or remove them from your store. You can also add a new product.



My Store





12. Add Product:

Add product page where you can add a new product to your store.

LA Store

Search

Q

Users

My Store

My Account

Log out

Cart (0)

Add Product

Product image

Choose File

 headphones.jpg

Product name

headphones

Category

electronics

Description

Bluetooth Headphones Wireless Over Ear Stereo I

Price

1

Save

13. Search:

You can search for any product or user.

LA Store

Search

Q

Users

My Store

My Account

Log out

Cart (0)

Search

Search term: "t-shirt"

T-shirt

\$99.00

View details



14. Admin Page:

Admin panel page contains reports about the transactions performed on the systems.



Admin Panel

Sold History

Product	Price	Seller	Buyer	Purchased Data
shoes	\$175.00	loay	Lina	December 10th 2021, 9:43:46 pm
sweater	\$255.50	Amr	Lina	December 10th 2021, 9:43:46 pm
leather jacket	\$300.00	Menna	Mostafa	December 10th 2021, 9:21:34 pm
T-shirt	\$150.00	Menna	loay	December 10th 2021, 9:17:29 pm
shoes	\$175.00	Menna	loay	December 10th 2021, 9:17:28 pm
TV	\$8000.00	Lina	Menna	December 10th 2021, 6:46:39 pm
T-shirt	\$150.00	Amr	Menna	December 10th 2021, 6:43:35 pm
leather jacket	\$300.00	Amr	Menna	December 10th 2021, 6:43:35 pm

Total Money: \$9505.5

Number of sold products: 8



Resources Needed

To run the server these requirements need to be installed.

```
asgiref==3.4.1
certifi==2021.10.8
cffi==1.15.0
charset-normalizer==2.0.9
coreapi==2.3.3
coreschema==0.0.4
cryptography==36.0.0
defusedxml==0.7.1
Django==3.2.9
django-cors-headers==3.10.1
django-rest-framework==0.1.0
django-templated-mail==1.1.1
djangorestframework==3.12.4
djangorestframework-simplejwt==4.8.0
djongo==1.3.6
djoser==2.1.0
idna==3.3
itypes==1.2.0
Jinja2==3.0.3
MarkupSafe==2.0.1
oauthlib==3.1.1
Pillow==8.4.0
pycparser==2.21
PyJWT==2.3.0
pymongo==3.12.1
python3-openid==3.2.0
pytz==2021.3
requests==2.26.0
requests-oauthlib==1.3.0
six==1.16.0
social-auth-app-django==4.0.0
social-auth-core==4.1.0
sqlparse==0.2.4
stripe==2.63.0
uritemplate==4.1.1
urllib3==1.26.7
```

To run the frontend, you need to install

Vue, axios, bulma, bulma-toast, moment using npm package manager



Role of Each Member

All work was done together through online meetings and pair programming.

Name	Contribution
عمرو ايهاب عبدالعزيز	API endpoints (backend) Database Report Testing
لؤى انور عبدالرازق	API endpoints (backend) Database Report Testing
لينة ايمن محمود	Database Frontend Report Testing
ليلى وائل سيد	Database Frontend Report testing



References

- 1- Django REST framework documentation
- 2- Bulma CSS documentation
- 3- Vue.js documentation