

2019 Fall

Web Programming Lab

Team Project Report



Team 5

Team Member

2013313217 Cho Sang-Yeon

2014311268 Jeong Seon-Ho

2018311574 Lee Gyu-Won

I. OVERVIEW.....	3
1. PREREQUISITES	3
2. FUNCTION IMPLEMENTS.....	3
3. DESIGN	4
4. DB MODELING (CLASS DIAGRAM).....	6
II. IMPLEMENTATION.....	7
1. ADMIN) MEMBER LIST	7
2. ADMIN) MEMBER MODIFICATION, DELETE.....	8
3. ADMIN) PRODUCT MODIFICATION, DELETE	8
4. SELLER) PRODUCT REGISTRATION	9
5. SELLER) MAP API /W GEOLOCATION (CURRENT POSITION).....	10
6. SELLER) PRODUCT MODIFICATION / PRODUCT CANCELED.....	11
7. SELLER) SELLER'S PRODUCT LIST	12
8. BUYER) PRODUCT LIST	12
9. BUYER) PRODUCT SEARCH	14
10. BUYER) WISHLIST	14
11. BUYER) PRODUCT BUY & BID.....	15
12. BUYER) CALCULATION AND PRINT OF SHOPPING LIST.....	16
13. BUYER) DIRECTION FROM CURRENT POSITION TO SELLER.....	16
14. COMMON PAGE) LOGIN PAGE	17
15. COMMON) REGISTRATION PAGE.....	18
16. COMMON PAGE) PRODUCT PAGE	18
17. COMMON) SECURITY REGISTRATION	21

I. Overview

1. Prerequisites
 1. Python 3.7
 2. Django 2.1
 3. HTML, CSS, JS
 4. DB: Sqlite3
 5. OS: Window, Linux, Mac OS
2. Function implements

#	Function Page Name	Completion
Administrator Page		
1	Member List	<input checked="" type="checkbox"/>
2	Member Modification / Delete	<input checked="" type="checkbox"/>
3	Admin's Product Modification / Delete	<input checked="" type="checkbox"/>
Seller Page		
4	Product Registration	<input checked="" type="checkbox"/>
5	Product Modification / Product Canceled	<input checked="" type="checkbox"/>
6	Seller's Product list	<input checked="" type="checkbox"/>
7	Map API /w Geolocation (Current Position)	<input checked="" type="checkbox"/>
Buyer Page		
8	Product list	<input checked="" type="checkbox"/>
9	Product Search	<input checked="" type="checkbox"/>
10	Wish List	<input checked="" type="checkbox"/>
11	Product buy & Bid	<input checked="" type="checkbox"/>
12	Calculation and Print of Shopping list	<input checked="" type="checkbox"/>
13	Direction from current position	<input checked="" type="checkbox"/>
Common Page		
14	Login Page	<input checked="" type="checkbox"/>
15	Registration Page	<input checked="" type="checkbox"/>
16	Product Page	<input checked="" type="checkbox"/>
17	Security Registration	<input checked="" type="checkbox"/>

3. Design

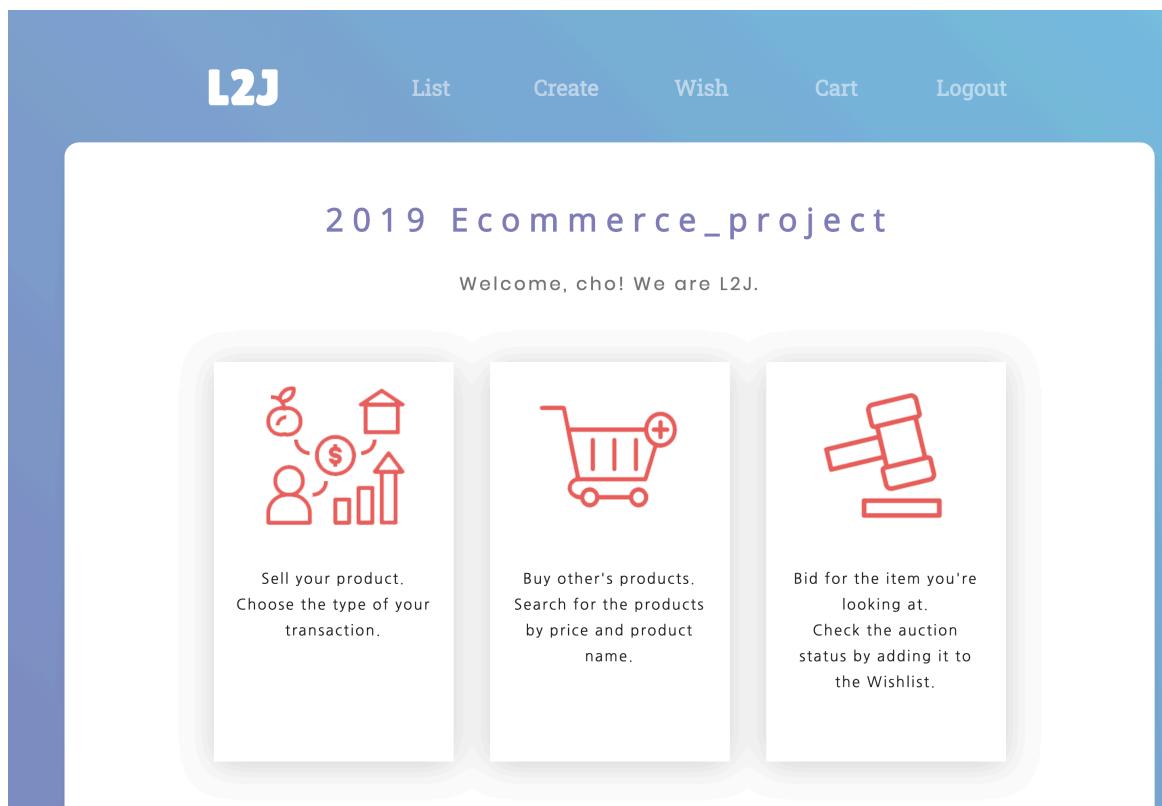


Figure 1 Main Page

A mobile interface showing a burger menu icon on the right. The main area displays a table of trading items:

Trading				
#	Name	Price	Place	Status
2	MacBook2	10000 won	seoul	purchasing compl
7	9pm	23112313123	dwqdqd	auction won

Figure 2 Responsive & Mobile Burger Menu

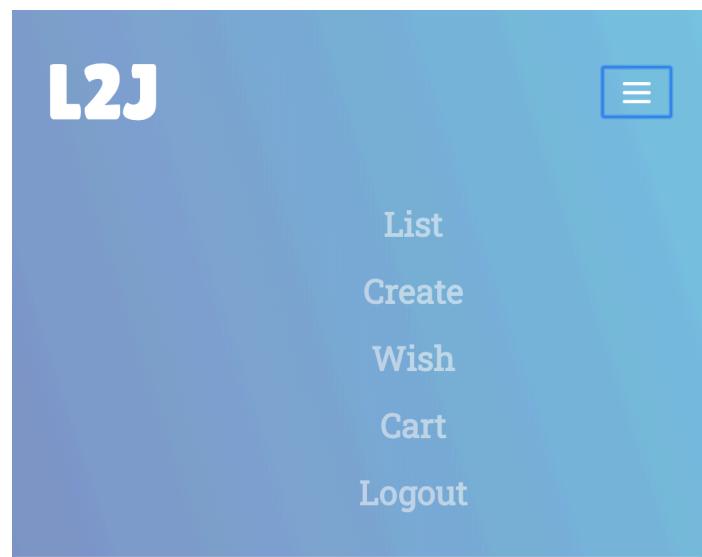


Figure 3 Open Burger Menu bar

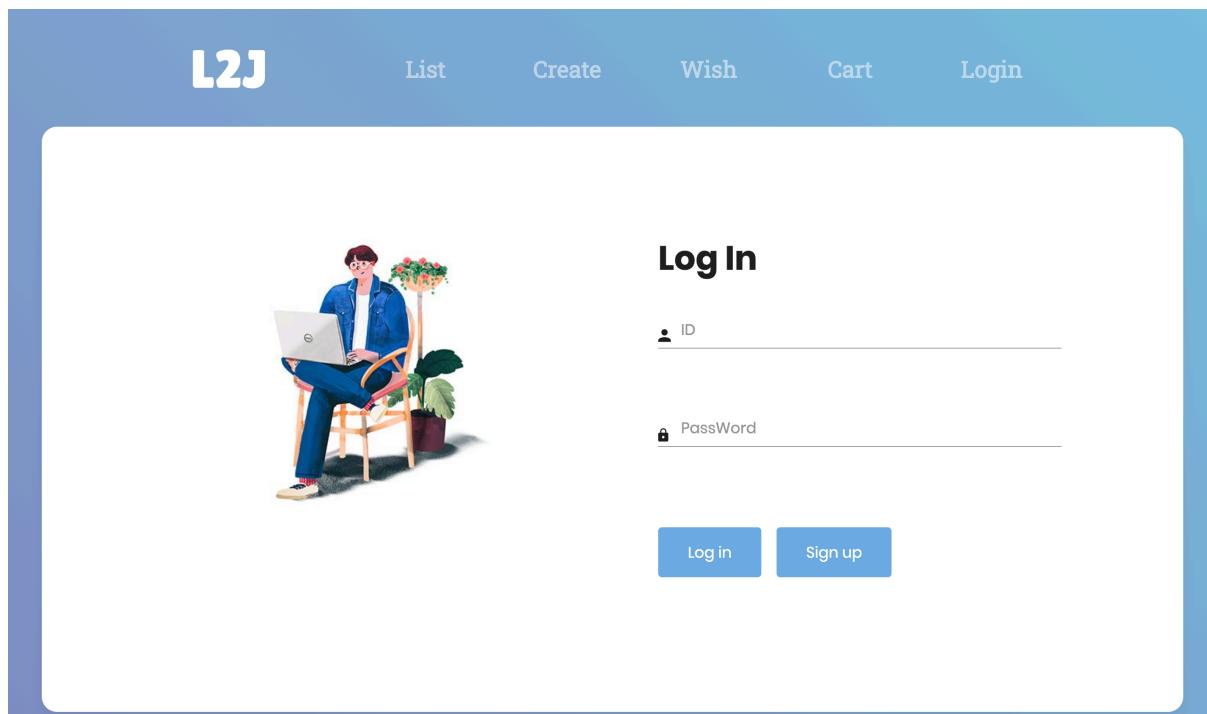
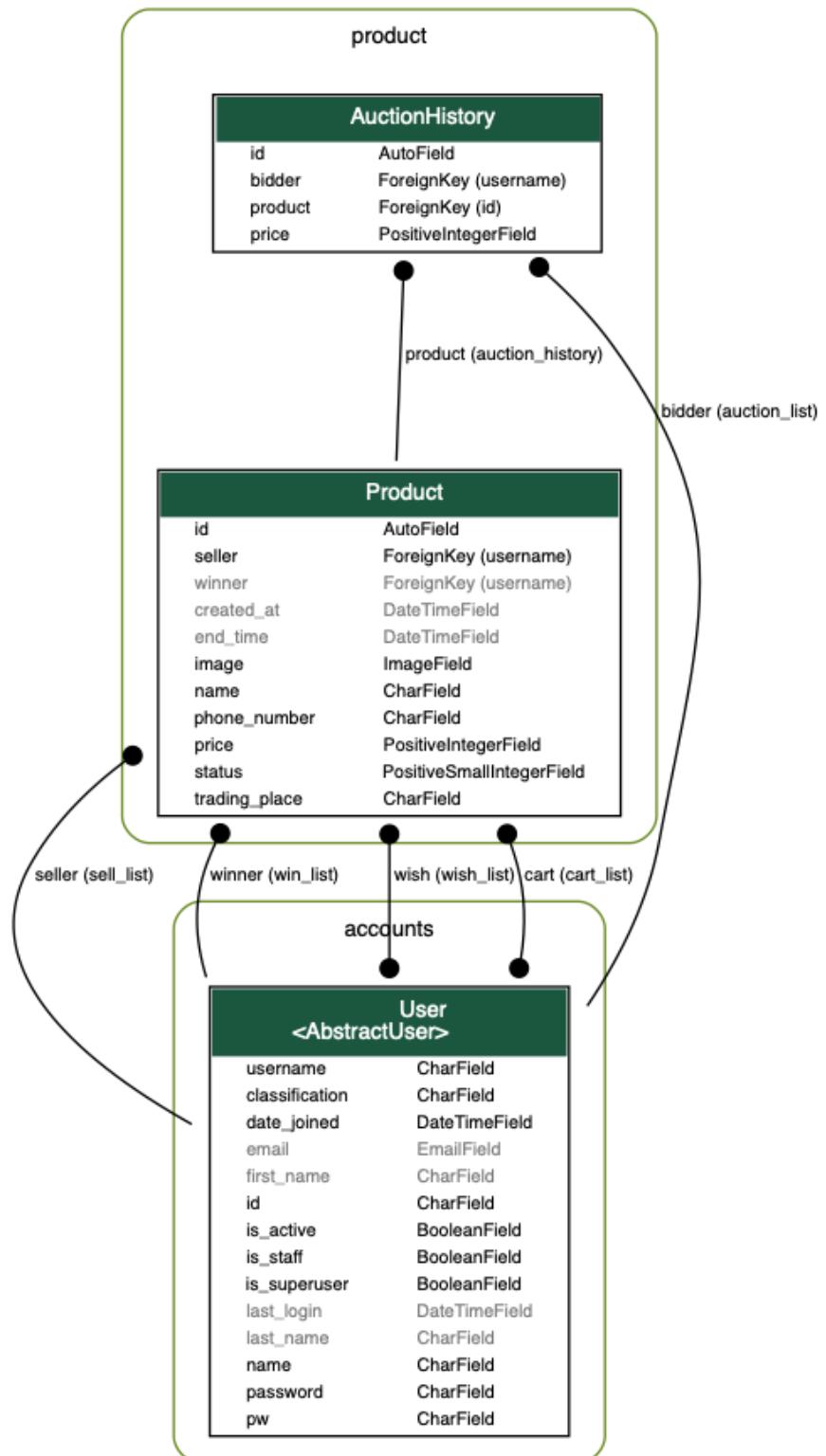


Figure 4 Login Page

4. DB Modeling (Class Diagram)



II. Implementation

1. Admin) Member List

*Extra Function : Another Admin Login Page

For secure, we do not make any button to go admin page. To access the admin page, we must enter directly from url to admin login page, where you enter the admin ID password that you created in the database in advance. Then we can see the list of database models, and in the list we can access to User model which show a member list of our project. At the member list page, admin can see user's password, name, classification. And also admin can search user by id and view each user list according to classification (buyer, seller).

The screenshots illustrate the Django admin interface:

- Login Page:** Shows the "Django administration" login screen with fields for "Username" and "Password", and a "Log in" button.
- Main Dashboard:** Shows the "Django administration" dashboard with three main sections:
 - ACCOUNTS:** Sub-sections include "Users" (+ Add, Change), "Groups" (+ Add, Change), and "Products" (+ Add, Change).
 - AUTHENTICATION AND AUTHORIZATION:** Sub-sections include "Groups" (+ Add, Change).
 - PRODUCT:** Sub-sections include "Products" (+ Add, Change).It also displays "Recent actions" and "My actions" (None available).
- Users List Page:** Shows the "Django administration" users list with the URL "Home - Accounts - Users". It includes a search bar, a "Select user to change" dropdown, and a table of users with columns: ID, PW, NAME, CLASSIFICATION, and STAFF STATUS. The table contains four rows:

ID	PW	NAME	CLASSIFICATION	STAFF STATUS
seono123123	*95827253	aad	Buyer	○
seono	a	a	Buyer	○
seon	a	a	Seller	○
True	True	-	-	○

A "FILTER" sidebar on the right allows filtering by classification: All, Buyer, Seller.

Last name:

Email address:

Staff status
Designates whether the user can log into this admin site.

Active
Designates whether this user should be treated as active. Unselect this instead of deleting accounts.

Date joined: Date: 2019-12-15 Today

Time: 00:24:13 Now

Username: seono123123

Name: asd

Pw: *5827253

Id: seono123123

Classification: Buyer

2. Admin) Member Modification, Delete

If admin click the id of user at the user list table, admin will go to the user detail information page. In this page, admin can modify user's information and delete user. According to the above, admin can delete user in user detail page and at user list page, admin can delete a large of user data at once.

3. Admin) Product Modification, Delete

***Extra Function**

The screenshot shows the Django Admin interface for modifying a product. The URL is 127.0.0.1:8000/admin/product/product/5/change/. The form includes fields for Trading place (123123), Status (auction), Price (45), Image (currently: img/no_image.png, change: 파일 선택 | 선택된 파일 없음), Seller (seono), Winner (seono), Wish (seono), Cart (seono), and End time (Date: 2019-12-14 Today, Time: 15:24:00 Now). There are also dropdown menus for Seller, Winner, Wish, and Cart, each showing 'seono' as an option. At the bottom, there are buttons for Delete, Save and add another, Save and continue editing, and SAVE.

Django administration

Select product to change

Search: a | 1 result (5 total)

Action: Go | 0 of 1 selected

NAME	STATUS	PRICE	SELLER
a	purchased completed	120	seono

1 product

FILTER

By status

- All
- auction
- purchased in progress
- purchased completed

When admin access to Product list page, same as user list page, admin can see detail of all products, delete and modify products.

4. Seller) Product Registration

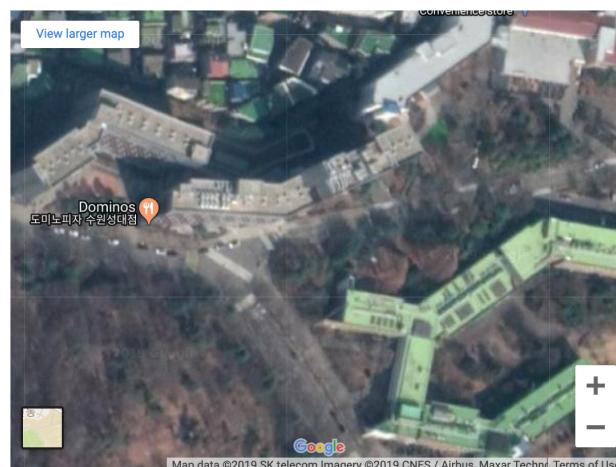
We made product registration page by using django's model form. Product's name, seller's phone number, trading place, price and photo are the common factors of both 'auction' and 'purchased in progress' option. A difference is that 'auction' option need a deadline factor. So, we use jQuery by showing the deadline-select only when the seller chooses 'auction'.

Product Registration

Name

Phone Number

Write Your Trading Place



Select Type of Transaction ↓

Write the price.

0

파일 선택 선택된 파일 없음

Submit

There are three status for product, but 'purchase complete' should not appear to the option that

buyer can choose. So we use 'display:none' option by disappear it.

```
class Product(models.Model):
    """Model definition for Product."""
    STATUS = (
        (1, 'auction'),
        (2, 'purchased in progress'),
        (3, 'purchased completed'),
    )
    #id_status option:nth-child(3){
    | display:none;
    }
```

*Extra Function: If seller set price to 0 and the status is not auction, error message shows up.



Error! 가격을 0원으로 설정하실 수 없습니다. *

Product Registration

5. Seller) Map API /w Geolocation (Current Position)

*Extra Function

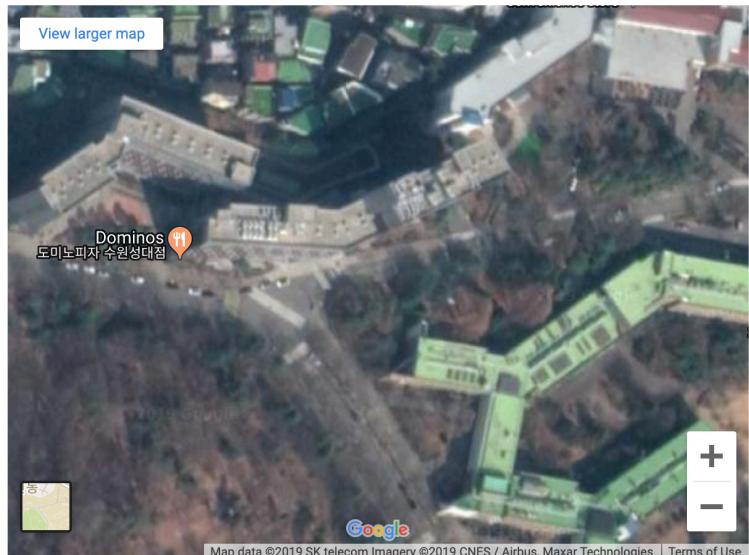
In Product Registration, Seller can insert his/her current location which collected by GPS or Wi-Fi Network. We use Google Maps API so that we can embed map iframe in our sites and seller can check their current location. For security, seller can also unauthorize this function.

Product Registration

전공 서적

010-7343-5356

기술사 신관



Select Type of Transaction purchased in progress 

6. Seller) Product Modification / Product Canceled

Seller can modify and cancel their own product registrations. Seller can update selling place, price, phone number and status of their products.

cho bids **30000** won. [edit](#)

dd bids **336** won. [delete](#)

dd bids **335** won.

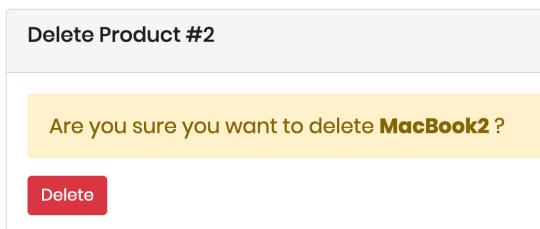
dd bids **23112313123** [edit](#)

won. [delete](#)

cho bids **31231313**

won.

*Extra Function: There is an one more page to check that seller really close their products.



7. Seller) Seller's Product List

If the user is seller, the user can see what they register in the product's list page. Seller can edit or delete the product's information they already registered before.

A screenshot of a web application showing a list of products. The header includes "L2J", "List", "Create", "Wish", "Cart", and "Logout". The main content shows a table with two rows of product data. The table has columns for #, Name, Price, Trading Place, Status, Seller, Created, History, and Manage. The "Seller" column for both rows is highlighted with an orange border. The "Manage" column contains "edit" and "delete" links. The footer includes "Move to Top" and "2019 E-commerce_Project L2J".

#	Name	Price	Trading Place	Status	Seller	Created	History	Manage
6	Macbook air	1000000 won	Sungkyunkwan univ. station	purchased in progress	sellerididid	Dec. 16, 2019, 1:36 a.m.		edit delete
7	Computer System 3e	0 won	Sungkyunkwan univ. station	aution	sellerididid	Dec. 16, 2019, 1:38 a.m.		edit delete

8. Buyer) Product list

Unlike seller, buyer can see all products that other sellers registered. In this page, we provide search function. Buyer's can search products by price, product name, seller's name. By using minimum price and maximum price option, they can set the range of price.

The screenshot shows a web browser window with the URL `127.0.0.1:8000`. The page title is "L2J". The main content area is titled "Product List" and contains a search bar with placeholder text "Search what you want!". Below the search bar is a table with columns: #, Name, Price, Trading Place, Status, Seller, Created, and History. The table has 7 rows of data. A blue "검색" (Search) button is located at the bottom right of the search form.

#	Name	Price	Trading Place	Status	Seller	Created	History
1	Airpods 2nd generation	200000 won	성대역 1번출구	purchased in progress	Gyuwon	Dec. 14, 2019, 9:30 p.m.	
2	macbook pro	2000000 won	명륜동	auction	Gyuwon	Dec. 14, 2019, 9:34 p.m.	
6	Macbook air	1000000 won	Sungkyunkwan univ. station	purchased in progress	sellerididid	Dec. 16, 2019, 1:36 a.m.	
7	Computer System 3e	0 won	Sungkyunkwan univ. station	auction	sellerididid	Dec. 16, 2019, 1:38 a.m.	

If we make option like it,

macbook	(product name)	판매자 이름	(seller's name)
1000000	(minimum price)	2000000	(maximum price)
<button>검색</button>			

Product list shows products which meet all options that buyer made.

The screenshot shows a web browser window with the URL `127.0.0.1:8000`. The page title is "L2J". The main content area is titled "Product List" and contains a search bar with placeholder text "Search what you want!". Below the search bar is a table with columns: #, Name, Price, Trading Place, Status, Seller, Created, and History. The table has 2 rows of data. A blue "검색" (Search) button is located at the bottom right of the search form.

#	Name	Price	Trading Place	Status	Seller	Created	History
2	macbook pro	2000000 won	명륜동	auction	Gyuwon	Dec. 14, 2019, 9:34 p.m.	
6	Macbook air	1000000 won	Sungkyunkwan univ. station	purchased in progress	sellerididid	Dec. 16, 2019, 1:36 a.m.	

9. Buyer) Product Search

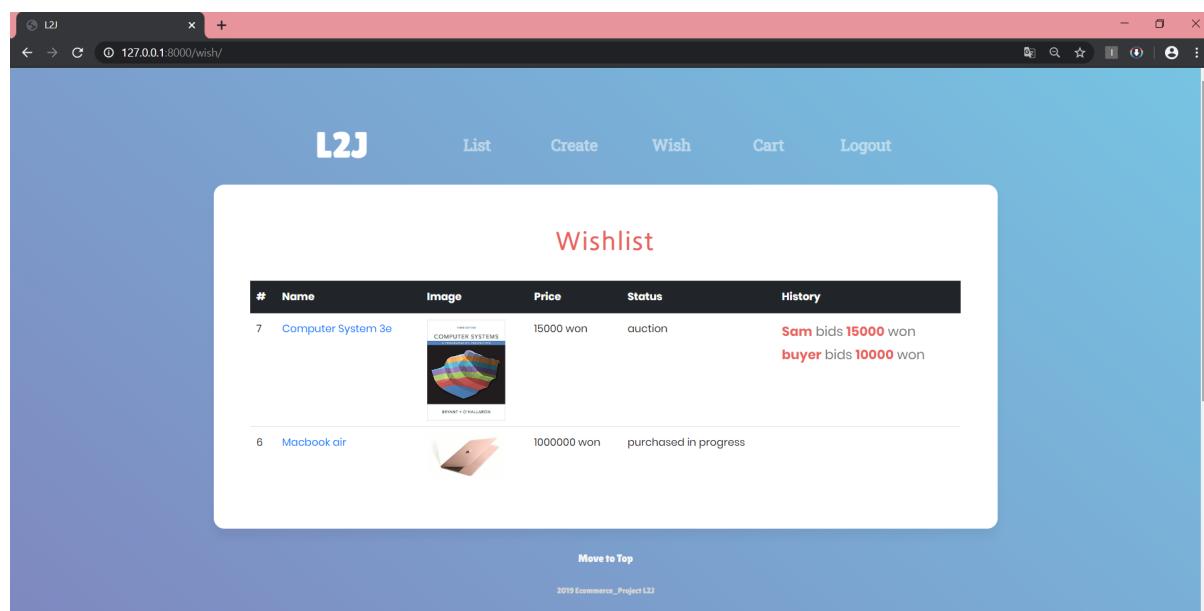
User can search their own purpose with many kinds of search parameters like Seller name, Product name, and hope price. We use Django ORM to implement this multiple search function. We customize form input tag's name attributes which has its own purpose, and this name go directly into DB filter. In Python, using ** in front of variable means that these variables using for key arguments.

```
@login_required
def ProductList(request):
    if request.method == "GET":
        product_list= Product.objects.all() if request.user.is_buyer() else
Product.objects.filter(seller=request.user)
        # 현재 접속자가 Seller인 경우 본인 판매 물건만
    else:
        # Seller name, Product name, hope price,
        filter_params = {key: value for key, value in request.POST.items() if key != 'csrfmiddlewaretoken' and value}
        product_list= Product.objects.filter(**filter_params)
        print(filter_params, product_list)

    return render(request, 'product_list.html', {"product_list": product_list} )
```

10. Buyer) Wishlist

Buyers can add products in their wish list. In wish list, they can see the status of the products. In case of auction, buyers can check the history of bid.



The screenshot shows a web browser window for the L2J project. The URL in the address bar is 127.0.0.1:8000/wish/. The page has a blue header with the L2J logo and navigation links for List, Create, Wish, Cart, and Logout. Below the header is a white content area titled 'Wishlist'. It contains a table with two rows of data. The table columns are #, Name, Image, Price, Status, and History. Row 7: Computer System 3e, Image of a computer system, 15000 won, auction, History: Sam bids 15000 won buyer bids 10000 won. Row 6: Macbook air, Image of a laptop, 1000000 won, purchased in progress.

#	Name	Image	Price	Status	History
7	Computer System 3e		15000 won	auction	Sam bids 15000 won buyer bids 10000 won
6	Macbook air		1000000 won	purchased in progress	

Move to Top

2019 Ecommerce_Project L2J

11. Buyer) Product Buy & Bid

Buyer can purchase a product or bid an auction. Buyer also can find auction's winner and who buy the product. If Product's state is "purchase completed", Buyer cannot purchase or bid the product. The status of the product change 'completed' when you click Buy. In Auction, User should bid and submit more than current price. Otherwise, Bidding is not acceptable.

상품 등록 시간 : Dec. 15, 2019, 10:34 p.m.

판매가: 2000000

상품 최종 구매자 :



Figure 5 Product Purchase

상품 등록 시간 : Dec. 16, 2019, 5:01 p.m.

옵션 마감 시간 : Dec. 16, 2019, 5:02 p.m.

판매가: 0

최고 입찰자 :



Figure 6 Auction

상품 등록 시간 : Dec. 9, 2019, 7:39 p.m.

판매가: 1111

상품 최종 구매자 : cho

판매 완료

Figure 7 Purchase Completed

12. Buyer) Calculation and Print of Shopping list

In case of 'auction' products, the product is added to the cart when the auction ends and the user wins. User can also add 'purchased in progress' products in the detail page. Users can see total number and price of the products they add to cart.

The screenshot shows a 'Shopping Cart' page from the L2J application. At the top, there are navigation links: List, Create, Wish, Cart, and Logout. The main content area is titled 'Shopping Cart'. It displays a table with two items:

#	Name	Image	Price
6	Macbook air		1000000 won
8	Iphone 11 case		40000 won

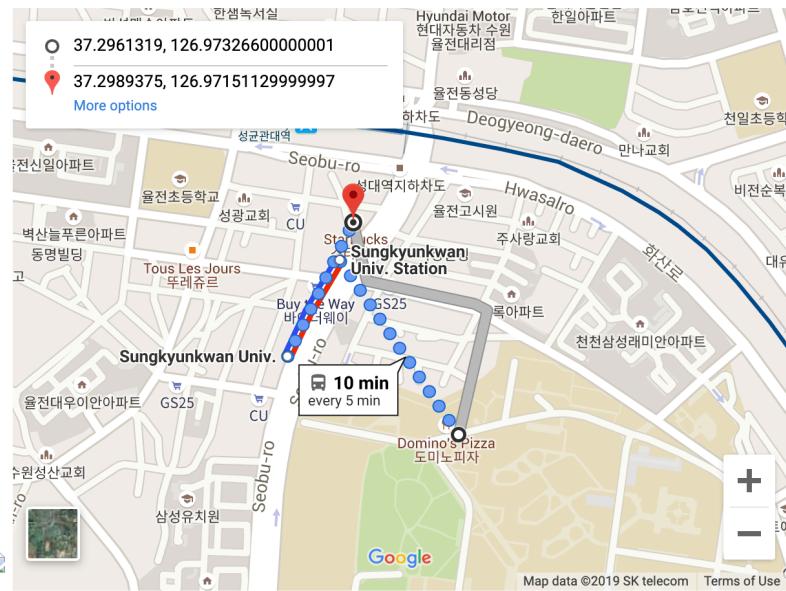
At the bottom left, it says 'Total 2 items'. On the right, it shows 'Total Price: 1040000 won'. A green 'Buy All' button is located at the bottom right.

If they click 'Buy All' button at the bottom of the page, shopping cart becomes empty and the status of products in shopping carts change to 'purchase complete'.

13. Buyer) Direction from current position to Seller

*Extra Function

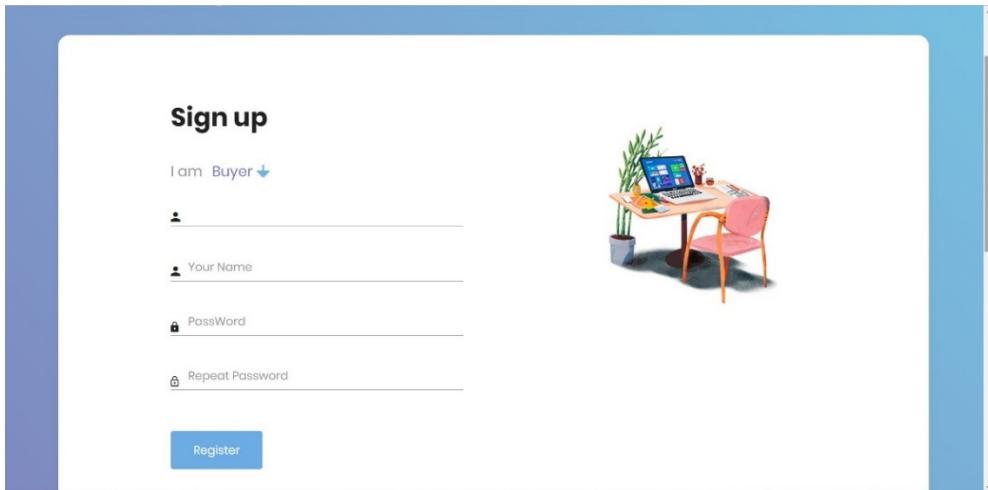
Buyer can get direction from buyer's current position to sellers. We use Google Direction API which can search direction (mode: transportation). Buyer can easily check their direction and time to go, and this information is very helpful to determine buy a product.



14. Common Page) Login Page

In django's account app, we have login function and registration function. If the user is not authenticated, the first page users will meet is the login page. We send user's login and password data by using POST.

15. Common) Registration Page



The registration page features a "Sign up" header and a "I am: Buyer" dropdown. It includes four input fields: "User ID", "Your Name", "PassWord", and "Repeat Password". A "Register" button is at the bottom. In the background, there's a cartoon illustration of a person sitting at a desk with a laptop, a pink chair, and some plants.

This is our Registration page. User should select buyer or seller, write their id, name and password twice.

We check the password twice to make the correct user input. By using checkbox, user can select if they want to buy products, or sell products.

```
class User(AbstractUser):
    username = models.CharField(max_length=45, blank=False, primary_key=True)
    name = models.CharField(max_length=45, blank = False)
    pw = models.CharField(max_length=45, default=True)
    id = models.CharField(max_length=45, default=True)
    classification = models.CharField(max_length=10, blank=False, choices=CHOISES)
```

This is our user model code. User's register information is saved by this model.

16. Common Page) Product Page

User can see product's detail page by click product's name in the product list page.

#	Name	Price	Trading Place	Status	Seller	Created	History
1	Airpods 2nd generation	200000 won	성대역 1번출구	purchased completed	Gyuwon	Dec. 14, 2019, 9:30 p.m.	
2	macbook pro	2000000 won	명륜동	auction	Gyuwon	Dec. 14, 2019, 9:34 p.m.	
6	Macbook air	1000000 won	Sungkyunkwan univ. station	purchased completed	sellerididid	Dec. 16, 2019, 1:36 a.m.	

In Product Page, photo and map iframe appear on the top.

Macbook 16인치



상품 등록 시간 : Dec. 16, 2019, 5:31 p.m.

판매가: 2000000

상품 최종 구매자 :



Figure 8 Product Detail Page

Some information appears different depend on the type of the transaction

상품 등록 시간 : Dec. 16, 2019, 4:11 p.m.

판매가: 150000

상품 최종 구매자 :



(Purchased in progress)

If user click purchase button, the status of transaction change to 'purchase complete'.

상품 등록 시간 : Dec. 16, 2019, 4:15 p.m.

옵션 마감 시간 : Dec. 16, 2019, 5 p.m.

판매가: 0

상품 최종 구매자 :



(Auction)

Until the deadline of the auction, users can bid. However, if user enter the price lower than the current maximum price, it is ignored and history does not show that one.

If the transaction ends, these buttons (wish, cart, purchase, bid) disappear and the winner of the auction is added in the page.

상품 등록 시간 : Dec. 16, 2019, 4:33 p.m.

옵션 마감 시간 : Dec. 16, 2019, 4:35 p.m.

판매가: 123456

상품 최종 구매자 : buyer

옵션 종료

Figure 9 When the auction is end

```
{% if product.status == 1 %}
<p>옵션 마감 시간 : {{product.end_time}}<br></p>
{% endif %}

<p>판매가: {{product.price}}</p>
<p>상품 최종 구매자 : {{product.winner.username}}</p><br>
{% if product.status == 1 and product.is_auction_end %}
<p>옵션 종료</p>
```

(Use django's IF condition by showing the html differently)

17. Common) Security Registration

```
from .models import User
from django import forms
import re
CHOISES=(('Buy','Buyer'),('Sell','Seller'))
)
def min_length_8_validator(value):
    if len(value) < 8:
        raise forms.ValidationError('8글자 이상 입력해주세요')

def need_alpha_and_special(value):
    pattern = re.compile('(?=.*\d{1,50})(?=.*[~!@#$%^&()-=]{1,50})(?=.*[a-zA-Z]{1,50}).{8,50}$')
    if pattern.match(value)== None:
        raise forms.ValidationError('비밀번호는 영문 및 특수기호를 각각 1개 이상 포함해야합니다.')
```

I am Buyer ↓

 aaa

8글자 이상 입력해주세요

 asd

 PassWord

비밀번호는 영문 및 특수기호를 각각 1개 이상 포함해야합니다.

 Repeat Password

이 입력란을 작성하세요.

 Register

Using python's regular expression module, we check user's input data. Id should be longer than 8 and password require at least one alphabet and one special symbol each. If form is not valid raise the error message.