

DB Project Final Report

DB Web Project

Department of Software

2016312568

Jung Hee Yoon

목차

1. Overview
2. ER diagram
3. Relation Data Model
4. Implementation and Results

1. Overview

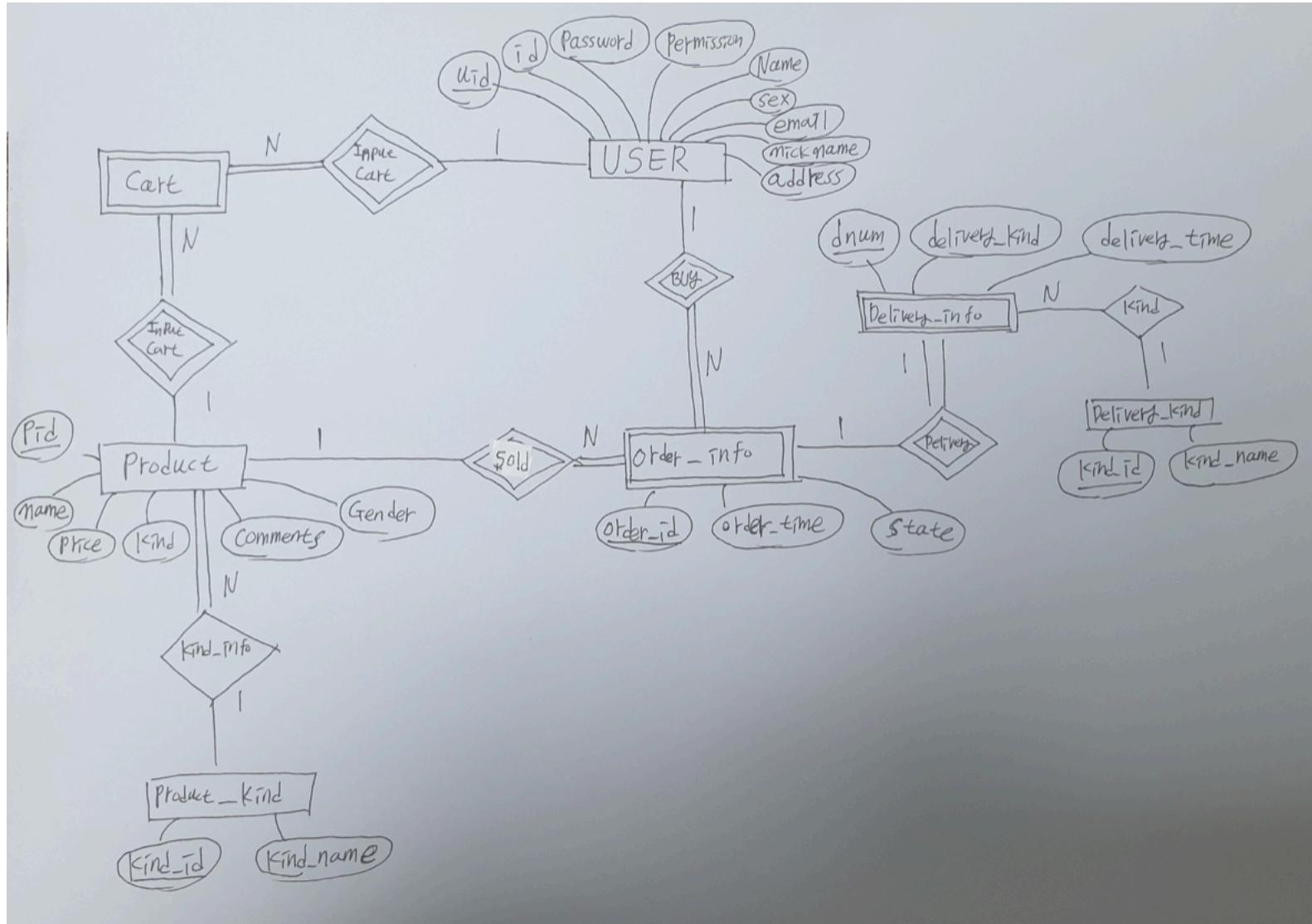
제가 만든 웹은 의류 쇼핑몰입니다. 의류 쇼핑몰 웹에는 크게 Customer Page와 Admin Page로 분류됩니다. 맨처음 웹을 들어가게 된다면 Customer Page가 보이며 관리자 페이지로는 접근이 불가 합니다. 또한, 로그인을 하였을 때 관리자 권한이 없는 회원은 그대로 Customer Page가 유지가 되며, 관리자 권한이 있는 회원으로 로그인을 하였을 경우 Admin Page로 이동하게 됩니다. Customer Page는 Homepage, Product Page, Cart page, User Profile Page로 구성되어 있으며 Admin Page는 Dash Board, Product List, User List, Order List, Delivery List로 구성되어 있습니다. 각 상세한 사항은 ‘4.Implementation and Results’에서 소개하도록 하겠습니다.

제 쇼핑몰의 컨셉은 회원에게만 판매(로그인을 하지 않으면 구매 불가)를 하고 있으며 각 상품에 대해서 사이즈는 모두 FREE사이즈이며 남, 여, 남여공용으로만 분류 되어있습니다. Customer가 상품을 카트에 넣고 구매를 하면 Order List에 추가가 됩니다. 이를 관리자가 보고 주문확인을 하고, 배송을 시킵니다. 배송을 시킬 경우, 배송사에 어떤 배송사를 이용했는지 선택하게 합니다. 자세한 사항은 역시 ‘4.Implementation and Results’에서 말씀드리도록 하겠습니다.

이 웹을 구현하기 위해서 Express(Node.js), Mysql, Vue.js를 사용하였습니다. 웹을 실행시키기 위해서는 아래 처럼 각 폴더에 들어가서 백엔드를 실행시킬거면 npm start, 프론트엔드를 실행시킬거면 npm run dev를 하면 실행이 됩니다.

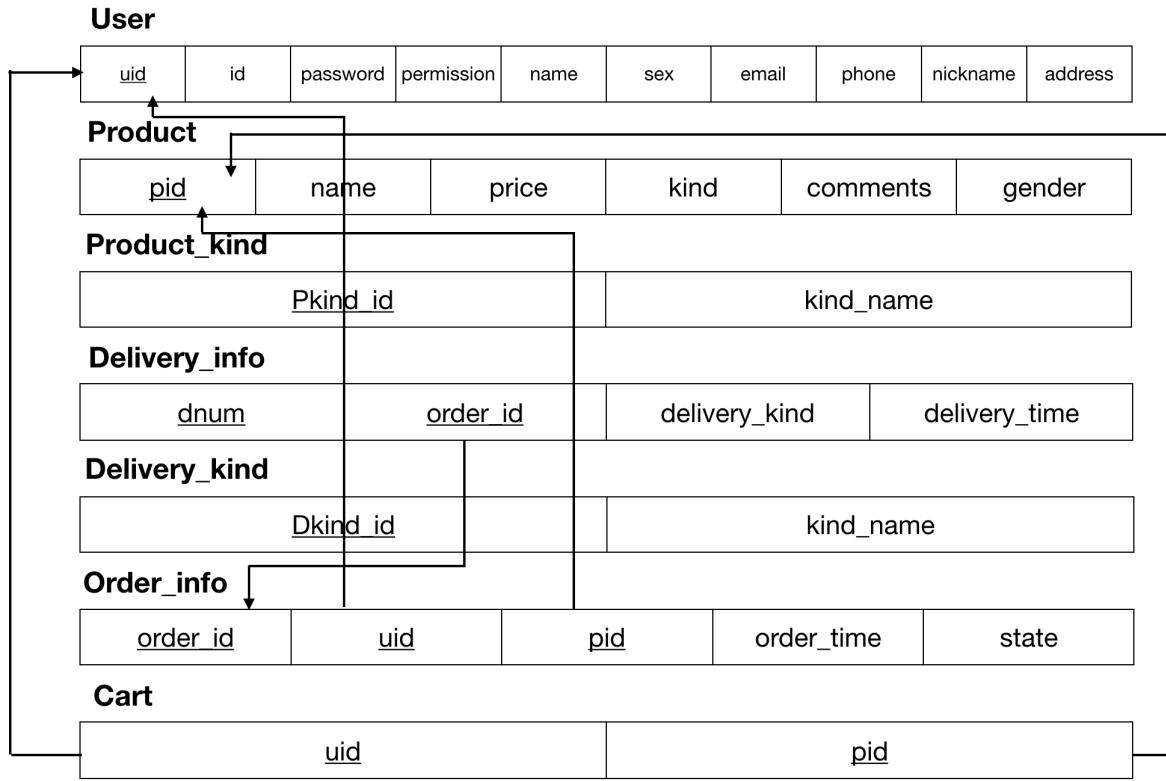
```
(base) jeonghuiyun-ui-MacBook-Pro:backend JungHeeYoon$ pwd  
/Users/JungHeeYoon/Desktop/Shop/backend  
(base) jeonghuiyun-ui-MacBook-Pro:backend JungHeeYoon$ npm start  
> backend@0.0.0 start /Users/JungHeeYoon/Desktop/Shop/backend  
> node ./bin/www  
  
connected db!  
POST /api/users/login 200 60.081 ms - 230  
GET /api/products 304 3.838 ms - -  
GET /api/orders/ 200 4.647 ms - 1518  
GET /api/orders/find_best 304 5.290 ms - -  
  
(base) jeonghuiyun-ui-MacBook-Pro:frontend JungHeeYoon$ pwd  
/Users/JungHeeYoon/Desktop/Shop/frontend  
(base) jeonghuiyun-ui-MacBook-Pro:frontend JungHeeYoon$ npm run dev  
> DB_WEB_Project@2.1.0 dev /Users/JungHeeYoon/Desktop/Shop/frontend  
> vue-cli-service serve --open  
[INFO] Starting development server...  
98% after emitting CopyPlugin  
[DONE] Compiled successfully in 38652ms  
  
App running at:  
- Local: http://localhost:8080/  
- Network: http://192.168.0.7:8080/  
  
Note that the development build is not optimized.  
To create a production build, run npm run build.  
15:26:37
```

2. ER diagram



위 ER다이어그램에 있는 각 attribute는 다음 Relation data model에서 설명을 적어놓겠습니다. 위의 ER다이어그램을 통해 USER가 Product를 Buy 하면 (Product 가 Sold 가 되면) Order_info에 추가됩니다. 또한 Order_info에서 Delivery를 하게되면 Delivery info에 들어가게 됩니다. User가 Product를 Cart에 Input Cart를 하게 되면 Cart에 추가가 됩니다. Product, Delivery_info는 각 엔티티마다 종류가 있습니다. 예를 들어, Product같은 경우 신발, 아우터, 바지 등으로 종류가 나뉘며 Delivery는 어떤 배송사에 의해 배송되는지가 다릅니다. 따라서 Product_kind, Delivery_kind를 만들어 정보를 넣어놓았습니다. Product_kind, Delivery_kind가 weak하지 않은 이유는 종류를 만들어놓고 (배송사가 여러개인데) 어떤 종류의 상품은 아직 준비되지 않았거나 어떠한 배송사를 이용하지 않을 수 있기에 weak entity가 아닙니다.

3. Relation Data Model



위의 ER Diagram 을 Relation Data Model로 나타냈을 때의 모습입니다. Order info, Cart같은 경우 Weak Entity이기에 User, Product의 PK를 가져와 자신의 PK로 삼습니다. Delivery_info는 Weak Entity이며 Order_info의 PK를 가져와 자신의 PK로 삼습니다. 아래의 정보는 각 attribute 의 정보입니다.

3.1. User

uid : 유저를 식별 할 id

id: 로그인 할 때 사용 할 id

password: 로그인 할 때 사용할 password

permission: 사용자 권한. (Customer, Admin)

name: 사용자 실명

sex: 사용자 성별

email: 사용자 이메일

phone: 사용자 전화번호

nickname: 사용자 닉네임

address: 사용자 주소. 주문 및 배송에 이 주소로 저장됨.

3.2. Product

pid : 상품을 식별할 id
name: 상품 이름.
price: 상품 가격
kind: 상품 종류
comments: 상품 설명
gender: 상품 성별 (남,녀, 남녀공용)

3.3. Product_kind

Pkind_id: 상품종류 id
kind_name: 상품 종류 이름 (ex) outer, pants, shoes....)

3.4. Delivery_info

dnum : 배송 id
order_id: 배송할 주문번호
delivery_kind: 배송사 종류
delivery_time: 배송보낸 시간

3.5. Delivery_kind

Dkind_id : 배송사 id
kind_name : 배송사 이름

3.6.Order_info

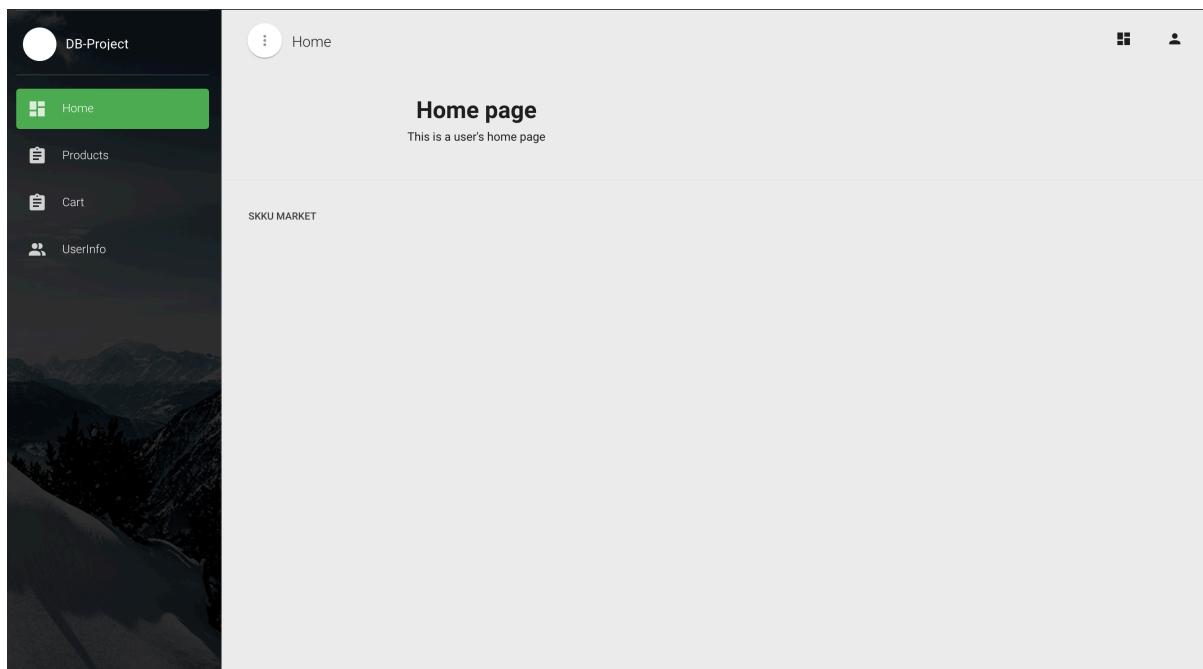
order_id : 주문번호
uid : 상품을 주문한 user uid
pid : user가 주문한 상품 pid
order_time : user가 주문한 시간
state : 주문상태 (주문, 주문 확인, 배송, 끝)

3.7.Cart

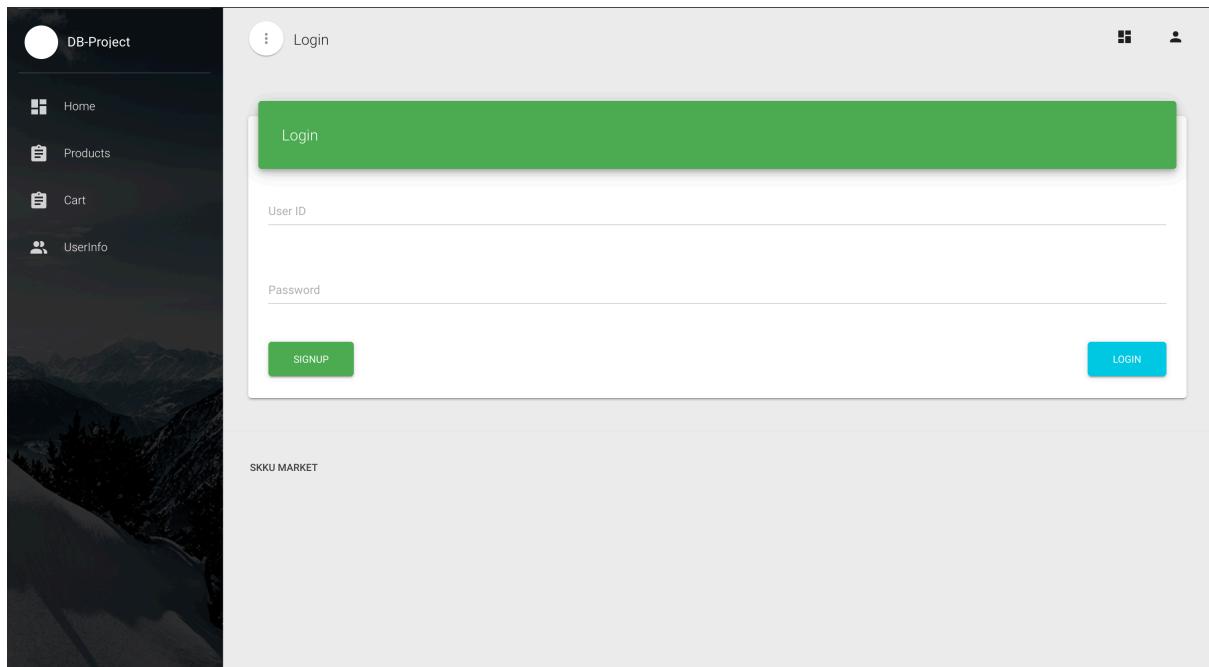
uid : Cart에 저장한 User id
pid: User가 Cart에 저장한 상품 pid

4. Implementation and Results

4.1. Customer Homepage

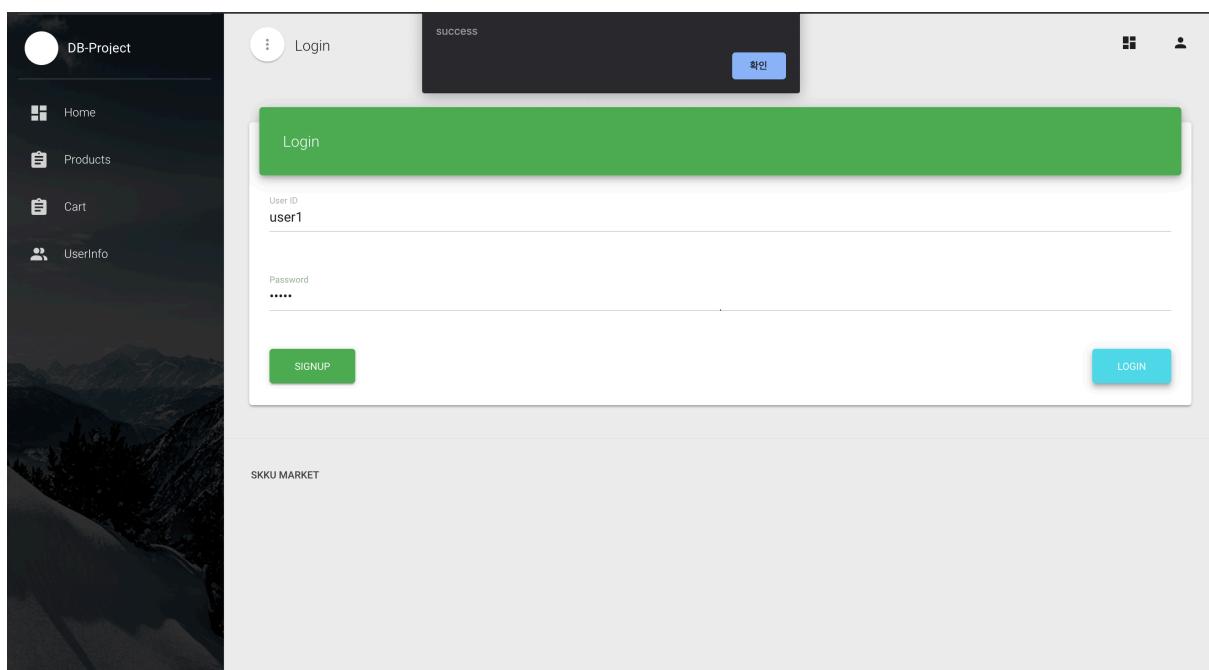


4.2. Customer Login Page

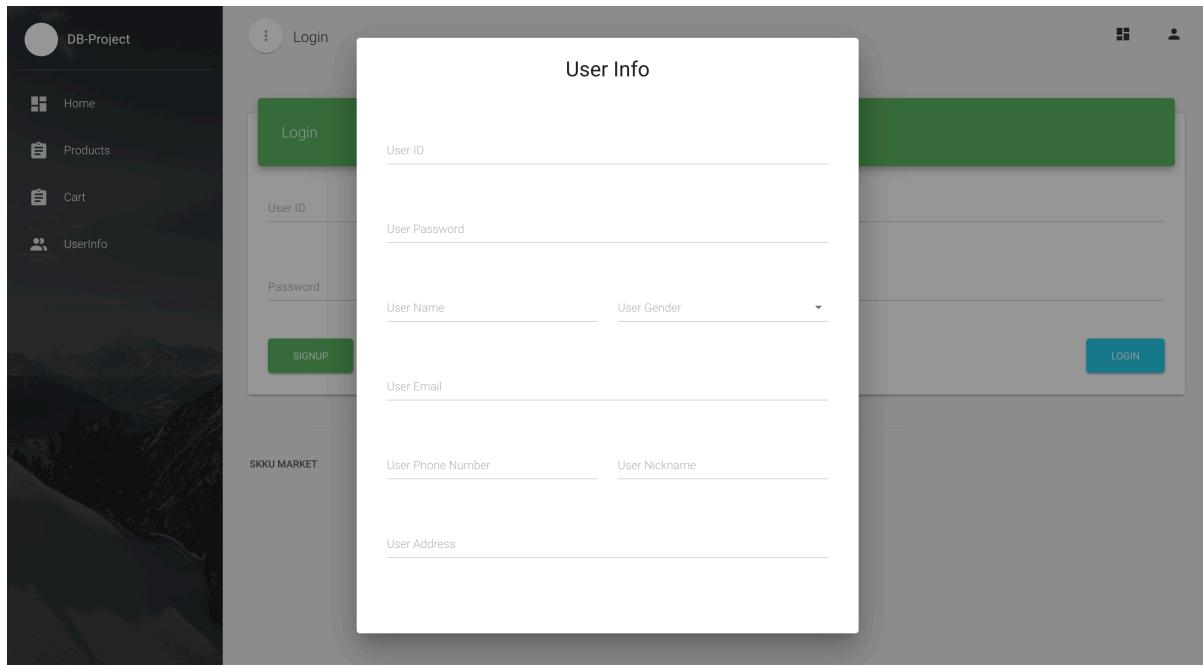


Function 1 : Login

Id와 Password를 입력후 데이터베이스에 있는 값과 비교하여 같으면 SUCCESS Alert를 띄워주며 permission에 맞는 페이지로 이동합니다. (Customer : Homepage, Admin : DashBoard)



Function 2. Sign Up

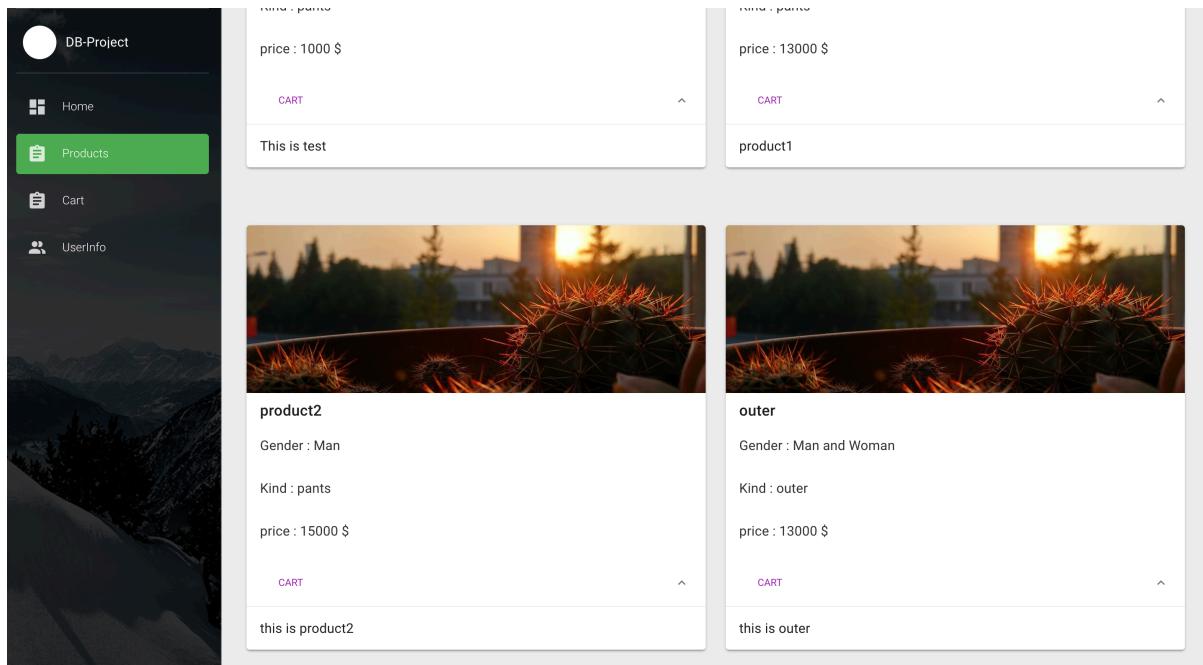


왼쪽아래 SIGNUP버튼을 클릭하게 되면 위와같은 Modal이 뜨게됩니다. 위의 내용을 적은뒤 스크롤하여 SAVE버튼을 누르게되면 회원가입이 되며 user데이터 베이스에 추가됩니다. 이때, UserID가 이미 테이블에 존재한다면 SAVE버튼을 눌렀을때 “Sorry use another ID”라는 Alert가 나오며 다시 입력하게 합니다. 데이터베이스 구조상 uid라는 유니크한 키가 있지만 login할때에 있어 UserID, password가 같게 사용할 수 있는 사람이 있을 수 있기에 한 처리입니다.

4.3. Customer Products Page

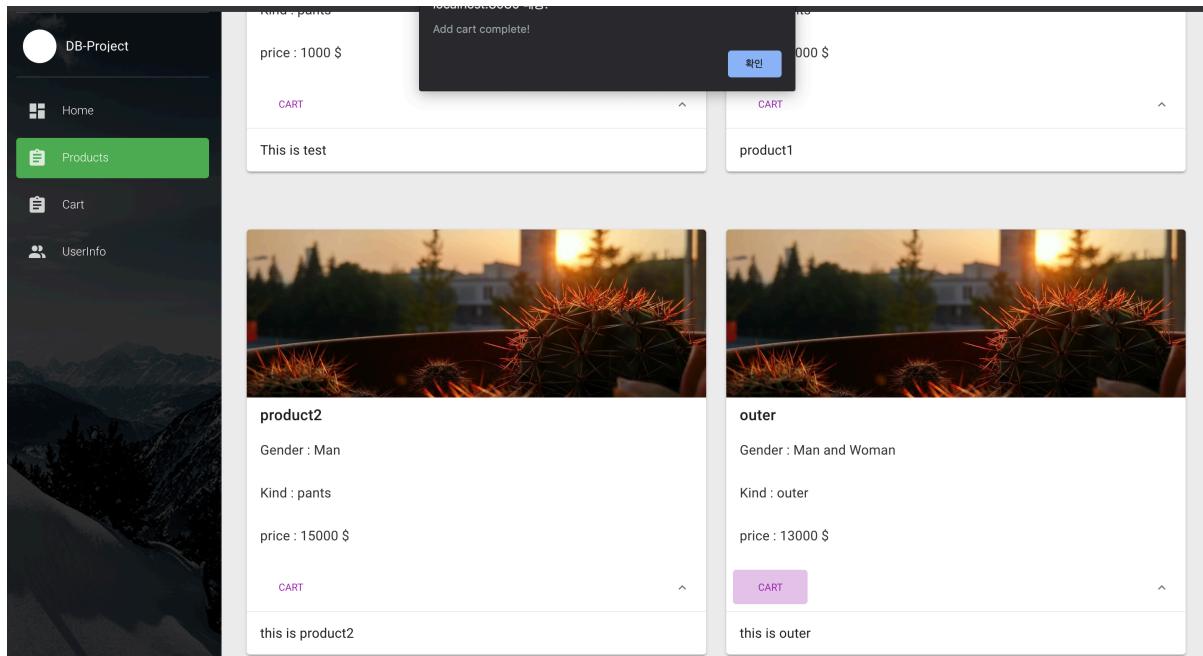
Function 3 : Get Products data

좌측 Drawer에서 Products를 누르면 Products 정보가 아래 사진처럼 나열됩니다.



Function 4 : Add cart In Products

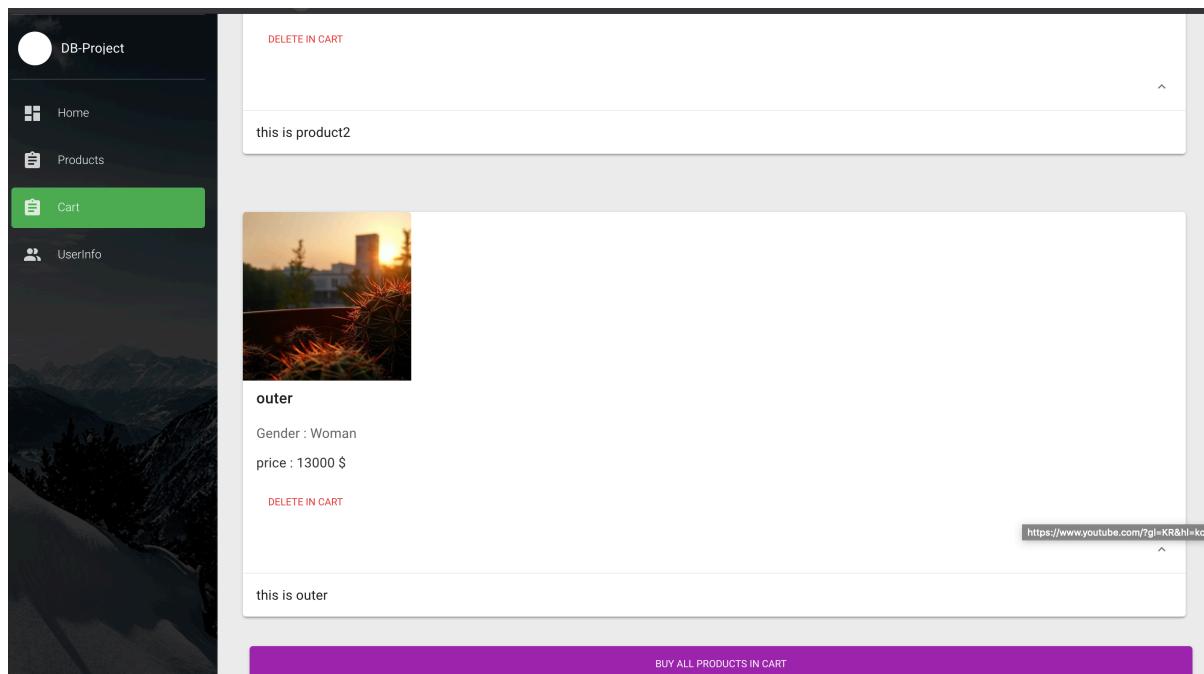
각 상품 좌측하단에 있는 CART버튼을 누르면 Cart 테이블에 상품을 추가합니다. 이미 들어가 있는 상품일 경우 Cart에 넣지 않습니다.



4.4. Customer Cart Page

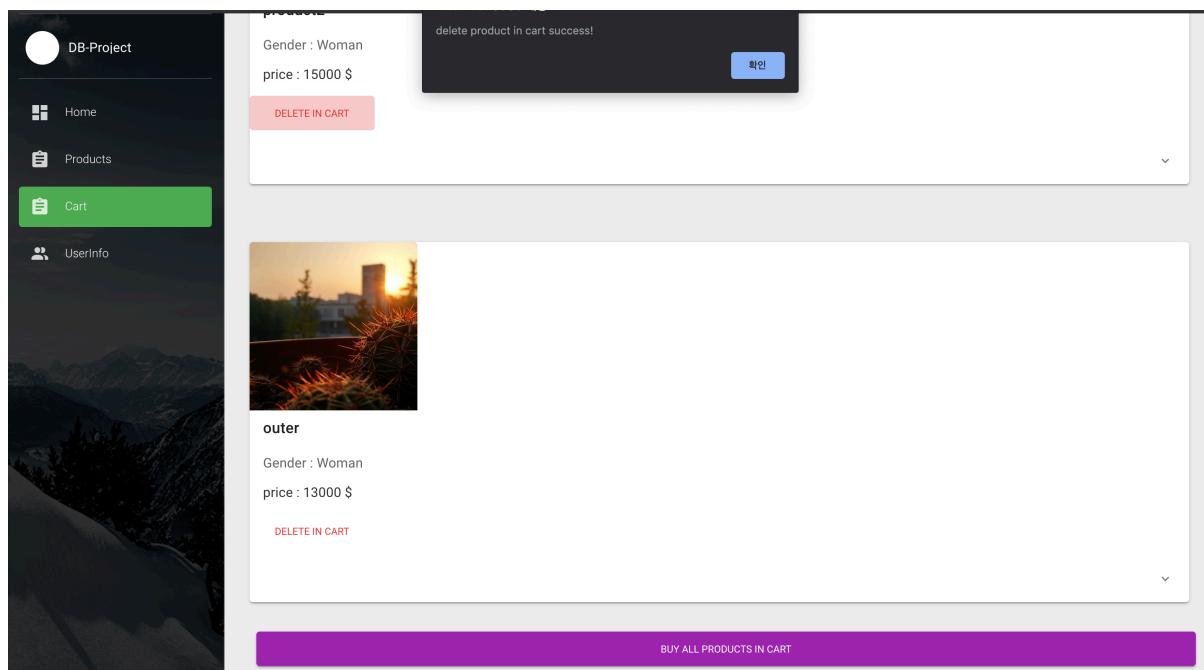
Function 5 : Get Cart list

Cart 테이블에 있는 uid, pid에 맞게 상품을 가져옵니다.



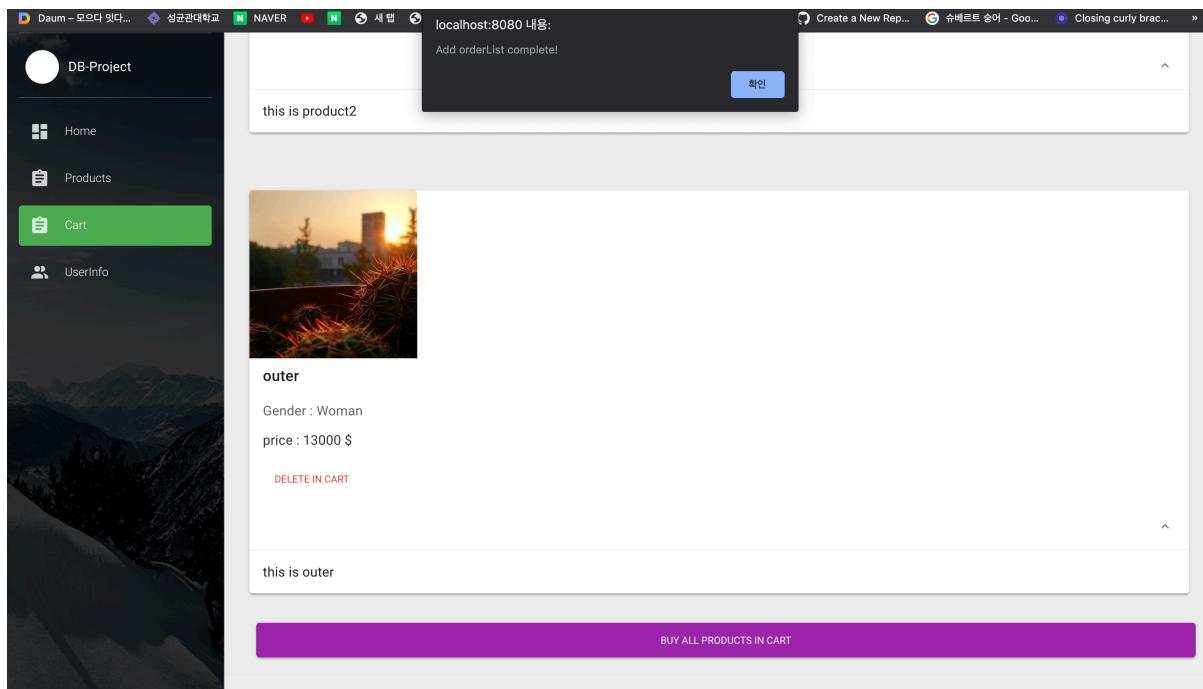
Function 6 : Delete Product in cart

가지고온 각 상품 리스트에서 좌측 하단에 Delete In Cart버튼을 클릭하면 Cart 테이블에서 상품을 삭제합니다.



Function 7 : Buy products in cart (Add products in cart to order_list)

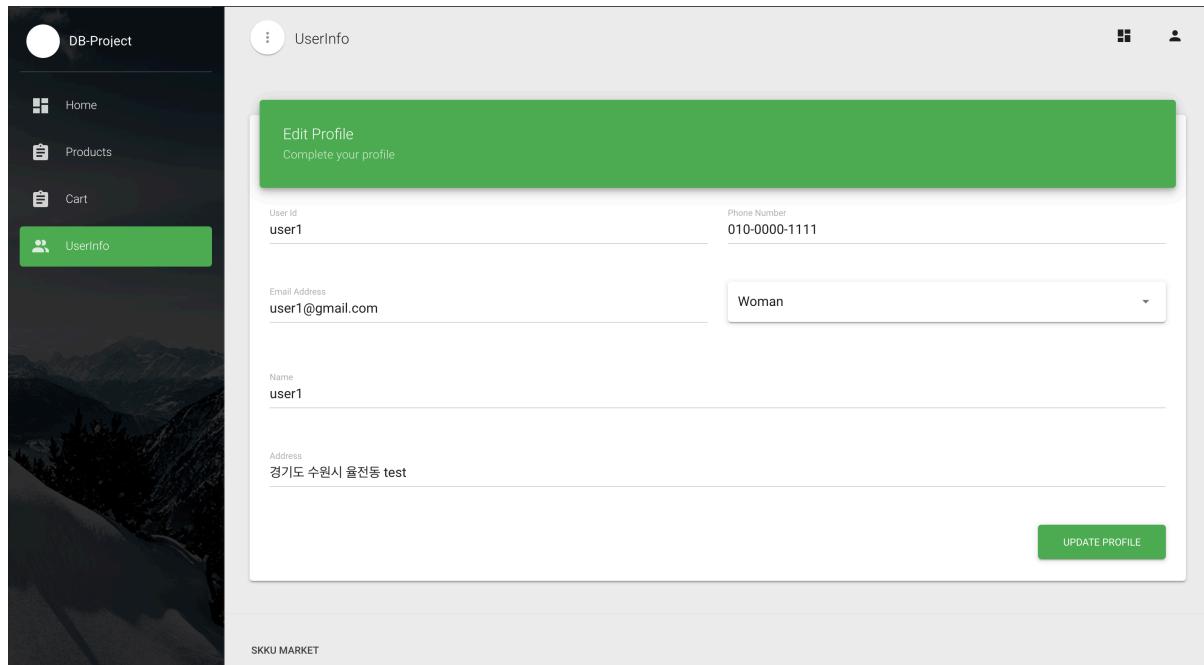
만약 카트에 담긴 상품들이 존재한다면 아래 보라색 Buy All Products In Cart라는 버튼이 생깁니다. 이버튼을 클릭하면 위의 상품들과 유저정보가 Order_list에 추가됩니다. 또한 상품을 구매한 시간이 Order_list에 들어갑니다. 또한 자동으로 Cart에 담긴상품들을 삭제합니다.



4.5. Customer User profile Page

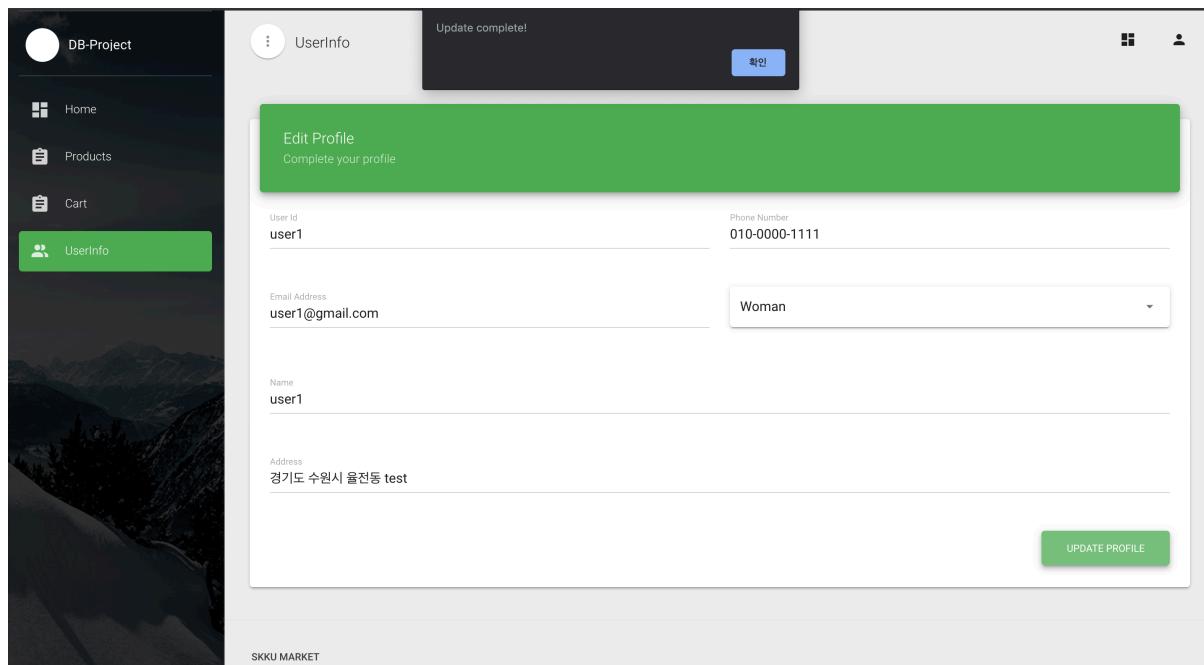
Function 8 : Get User Profile

User 데이터를 가져와 띄워줍니다.



Function 9 : Update Profile

위에 사진에서 내용을 바꾼뒤 우측아래 UPDATE Profile버튼을 누르면 유저정보가 갱신됩니다.

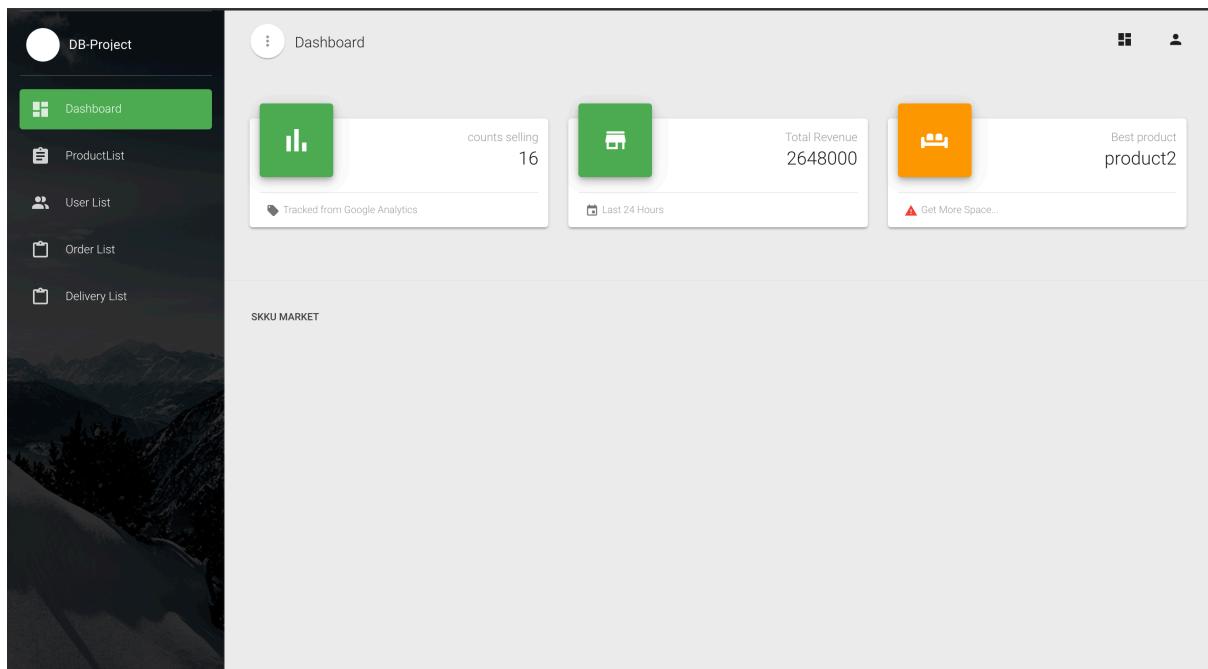


4.6. Admin DashBoard Page (After login admin page)

Function 10 : Get Total selling count, Total Revenue. Also get Product that is most sold.

Admin 권한이 있는 계정으로 들어왔을때 처음 보이는 페이지입니다.

Order_list에서 총 팔린 상품 개수와 총 판매액, 가장 많이 팔린 상품을 가져와 띄워줍니다.



4.7. Admin Product List page

Function 11 : Get Products List
상품들 정보를 가져와 띄워줍니다.

The screenshot shows the 'ProductList' page within the 'DB-Project' application. The left sidebar has 'ProductList' selected. The main area displays a table titled 'Products' with the following data:

No.	PID	Name	Price
0	0	test	100000
1	1	test2	1000000
2	2	test3	1000
3	3	product1	13000
4	4	product2	15000
5	5	outer	13000

An 'ADD PRODUCT' button is located in the top right corner of the main content area.

Function 12 : Add Product in Product list
우측상단에 있는 ADD Product버튼을 누르면 아래와 같이 Modal이 띄워집니다.
아래의 정보를 채우고 SAVE를 누르면 Product가 추가됩니다.

The screenshot shows the 'ProductList' page with a modal window titled 'Product Info' overlaid. The modal contains fields for 'Product Name', 'Product Price', 'Select Product Kind', 'Comment (Information of Products)', and 'Gender'. At the bottom of the modal are 'CLOSE' and 'SAVE' buttons. The background shows the list of products with the 'ADD PRODUCT' button visible.

4.8. Admin User List Page

Function 13 : Get User List

User List를 가져와 띄워줍니다.

The screenshot shows a web-based application interface for managing users. On the left, there is a sidebar with a dark background featuring a mountain landscape image. The sidebar contains the following navigation items:

- DB-Project
- Dashboard
- ProductList
- User List** (highlighted in green)
- Order List
- Delivery List

The main content area has a light gray background. At the top, it says "User List". Below that is a table titled "User List". The table has columns: No., UID, ID, Nick Name, and Permission. The data in the table is as follows:

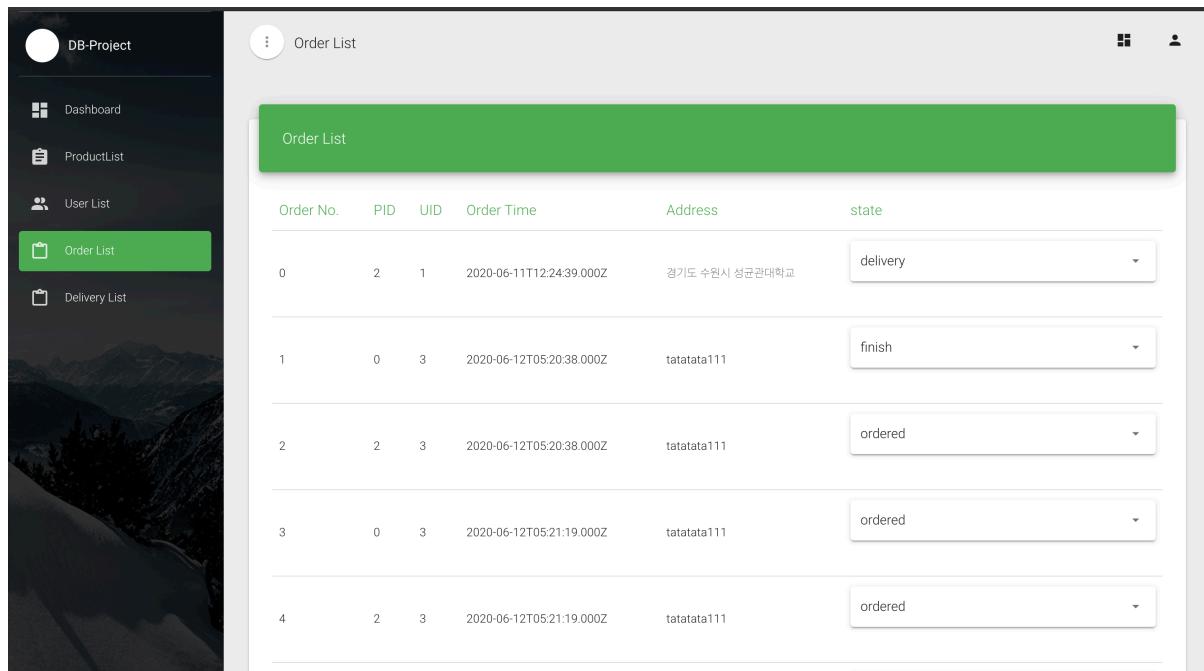
No.	UID	ID	Nick Name	Permission
0	0	test3	test3	admin
1	1	test3	test3	admin
2	2	test2	test2	admin
3	3	test31	test3	customer
4	4	test1	test1	customer
5	5	admin	admin	admin
6	6	user1	user1	customer

At the bottom of the main content area, it says "SKKU MARKET".

4.9. Admin Order List Page

Function 14 : Get Order Info

주문 내역을 가져와 띄워줍니다.



The screenshot shows the 'Order List' page within a web application. On the left, there is a sidebar with a dark background featuring a mountain landscape image. The sidebar contains the following menu items:

- DB-Project
- Dashboard
- ProductList
- User List
- Order List** (highlighted in green)
- Delivery List

The main content area has a header 'Order List' and displays a table with the following data:

Order No.	PID	UID	Order Time	Address	state
0	2	1	2020-06-11T12:24:39.000Z	경기도 수원시 성군관대학교	delivery
1	0	3	2020-06-12T05:20:38.000Z	tatata111	finish
2	2	3	2020-06-12T05:20:38.000Z	tatata111	ordered
3	0	3	2020-06-12T05:21:19.000Z	tatata111	ordered
4	2	3	2020-06-12T05:21:19.000Z	tatata111	ordered

Function15 : Change State

아래와 같이 주문 상태를 바꿀 수 있게 해줍니다. 값이 바뀌면 바뀔때마다 테이블을 갱신해줍니다.

The screenshot shows the 'Order List' page within a web application. On the left is a sidebar with 'DB-Project' at the top, followed by 'Dashboard', 'ProductList', 'User List', 'Order List' (which is highlighted in green), and 'Delivery List'. The main area has a header 'Order List' and a table with columns: Order No., PID, UID, Order Time, Address, and state. The 'state' column for each row contains a dropdown menu. The dropdown for the first row (Order No. 0) shows 'delivery'. The dropdown for the second row (Order No. 1) shows 'finish', 'ordered', 'confirm', and 'delivery'. The dropdown for the third row (Order No. 3) shows 'finish'. The dropdown for the fourth row (Order No. 4) shows 'ordered'.

이때 delivery를 선택하게 되면 아래와 같이 배송사를 선택하게 하며 배송사를 선택한뒤 SAVE버튼을 클릭하면 Delivery List에 해당 주문정보를 추가합니다. 이때 state를 Delivery로 바꾼 시간이 Delivery_time에 들어갑니다.

The screenshot shows the same 'Order List' page as before, but with a modal dialog titled 'Delivery Info' overlaid. The dialog has a dropdown labeled 'Select Delivery company' with a note '*indicates required field'. At the bottom of the dialog are 'CLOSE' and 'SAVE' buttons. The background table rows are partially visible, showing the state dropdowns again.

4.10. Admin Delivery List Page

Function 16 : Get Delivery info

주문 목록에서 배송상태를 추가시키면 배송 리스트에 추가됩니다. 배송리스트를 가져와서 띄워줍니다.

The screenshot shows a user interface for managing deliveries. On the left is a dark sidebar with a navigation menu:

- Dashboard
- ProductList
- User List
- Order List
- Delivery List** (highlighted in green)

The main content area has a header "DeliveryInfo" and a sub-header "Vuetify Typography". It includes a note: "Please checkout the [full documentation](#)". Below this is a table titled "Delivery List" with the following data:

Delivery No.	Order No.	Delivery Kind	Order Time
0	0	0	2020-06-12T04:10:00.000Z
1	0	4	2020-06-12T19:50:55.000Z
2	10	1	2020-06-12T21:34:42.000Z
3	12	3	2020-06-13T02:38:02.000Z

At the bottom of the main content area, it says "SKKU MARKET".