

# Intro

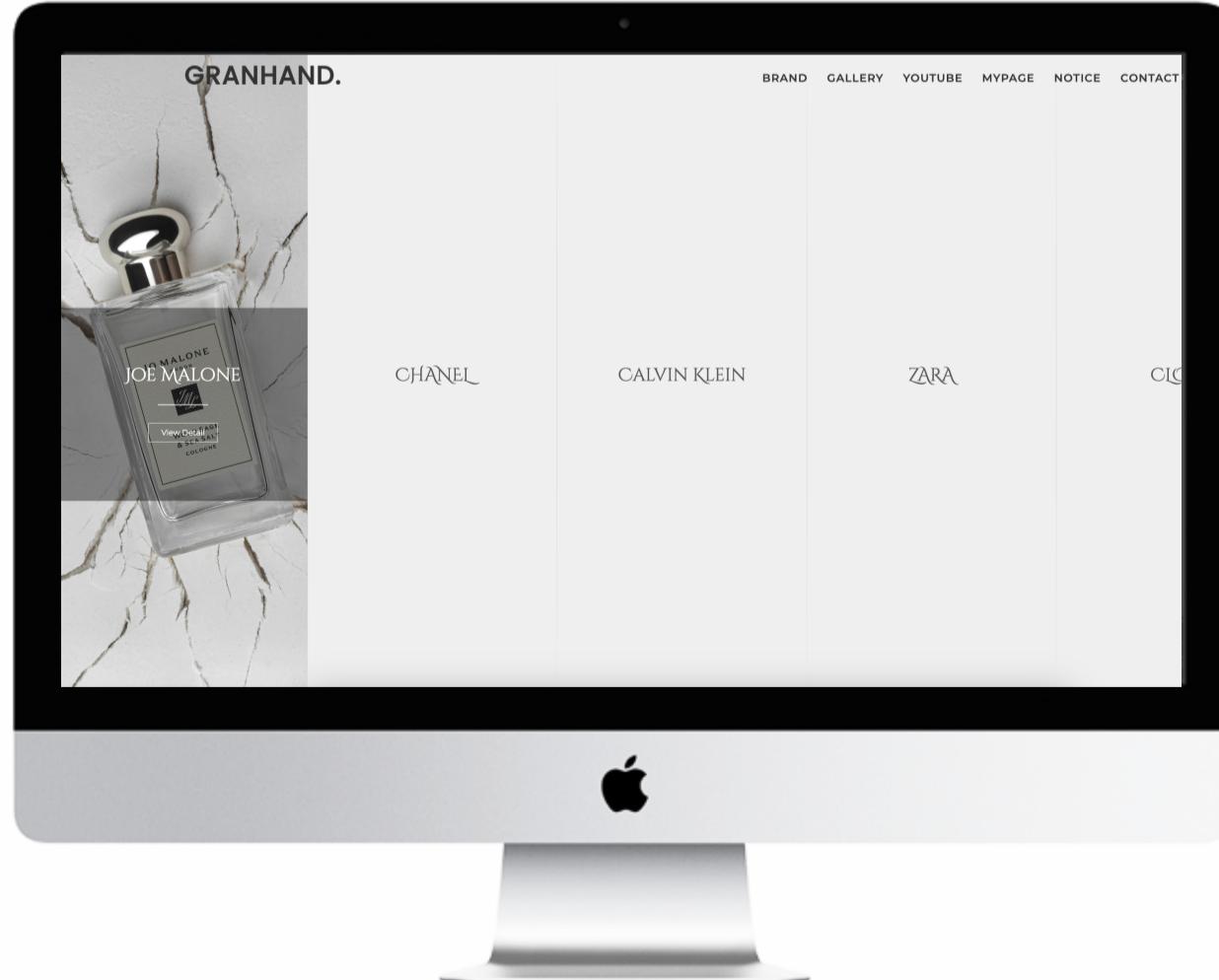
- 리액트 기업형 포트폴리오 소개

**name** : GRANHAND.  
**responsive** : PC, TABLET, MOBILE  
**Skills** : REACT, REDUX-SAGA, SCSS

- 폰트 : 구글 웹 폰트

**Title :** Poppins      **Sub Title :** Montserrat  
**Hello**      **Hello**      **Strong Font :** Cinzel decorative      **Youtube Font :** El Messiri  
**Hello**      **Hello**

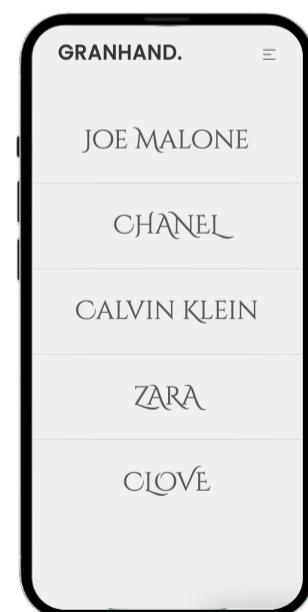
Pc



Tablet



Mobile



- 색상

	#fa8072		#ff5722		#c71010
	#fefbf4		#1d66be		#445870
	#6f3d14		#333333		

# Folder Structure

---

 node_modules		
 public		
 DB	• DB	axios로 member.js / swiper.js 파일 데이터 받기
 member.js		
 swiper.js		
 img	• img	public 폴더에 접근하기 위한 \${process.env.PUBLIC_URL} / 절대경로를 활용하여 이미지 불러오기
 index.html	• index.html	렌더링된 결과 브라우저에 출력
  src		
 components	• components	common / main / sub로 세분화하여 관리
 common		
 main		
 sub		
 redux	• redux	외부 data를 컴포넌트 외부에서 관리 / 비동기를 동기처리하기 위한 것
 scss	• scss	css 전처리기 : sass --watch 후 스타일 작업
 App.js		
 Index.js	• index.js	컴포넌트를 조합하여 렌더링하는 파일

# Common Structure 01

## components

common

JS Header.js

JS Footer.js

JS Layout.js

JS Menu.js

JS Modal.js

- **Menu** Tablet / Mobile 영역에서 나타날 헤더 메뉴
- **Modal** Gallery / Youtube 페이지 팝업 창 모달

- **Layout** 서브 공통 페이지

```
<section className={`content ${props.name}`} ref={frame}>
  <div className='sub_con'>
    <div className='inner'>
      <h1>{props.name}</h1>
      {props.children}
      /* 공통 레이아웃 이외의 것들 */
    </div>
  </div>
</section>
```

## SCSS

\_reset.scss  
\_color.scss  
\_font.scss  
\_mixin.scss  
\_modal.scss  
\_layout.scss  
\_header.scss  
\_footer.scss  
style.scss

style 초기화

공통 스타일 변수로 저장 / 관리

각 컴포넌트별 스타일 관리

## 반응형 사이즈 변수 설정

```
$tablet: 1179px;
$mobile: 539px;
```

## 구글웹폰트 연결

```
@import url('https://fonts.googleapis.com/css2?family=Cinzel+Decorative:wght@400;700&family=Cinzel:wght@400;500;600&family=Montserrat:wght@400;500;600&family=Philosopher&family=Poppins:wght@300;500&display=swap');
```

## 공통 컴포넌트

```
@import 'common/reset';
@import 'common/mixin';
@import 'common/font';
@import 'common/color';
@import 'common/header';
@import 'common/footer';
@import 'common/modal';
@import 'common/layout';
@import 'common/menu';
```

## 메인 컴포넌트

```
@import 'main/main';
@import 'main/visual';
@import 'main/news';
@import 'main/pics';
@import 'main/vids';
@import 'main/btns';
```

## 서브 컴포넌트

```
@import 'sub/brand';
@import 'sub/gallery';
@import 'sub/youtube';
@import 'sub/notice';
@import 'sub/mypage';
@import 'sub/contact';
```

# Common Structure 02

- **Header**

공통 header

- **Sub**

페이지마다 다른  
content들이 담길  
Sub 페이지

- **Footer**

공통 footer

The screenshot shows a mobile view of the GRANHAND website. At the top is a header bar with the brand name "GRANHAND." and navigation links: BRAND, GALLERY, YOUTUBE, MYPAGE, NOTICE, and CONTACT. Below the header is a section titled "BRAND" containing a quote from NUBE about a notebook. The main content area features three circular profile pictures of staff members: Kim Nam-Gil (COMPANY CEO), Lee Da-Hee (MANAGER), and Cha Eun-Woo (EXHIBITION MANAGER). Each profile includes a small bio and an email address. At the bottom is a footer section with links to Store, Newsletter, and Support, along with social media icons and a copyright notice.

## Tablet/Mobile

GRANHAND.



Brand  
Gallery  
Youtube  
Mypage  
Notice  
Contact

해당 버튼 클릭시  
Tablet/Mobile  
메뉴 디자인

# Main Components

Main 컴포넌트에 section별 컴포넌트 분리

## components

### main

- JS** Main.js
- JS** Visual.js
- JS** News.js
- JS** Pics.js
- JS** Vids.js
- JS** Btns.js
- JS** Cookie.js

### Visual

- prev / next 버튼 클릭 시 이미지 scale 효과

### News

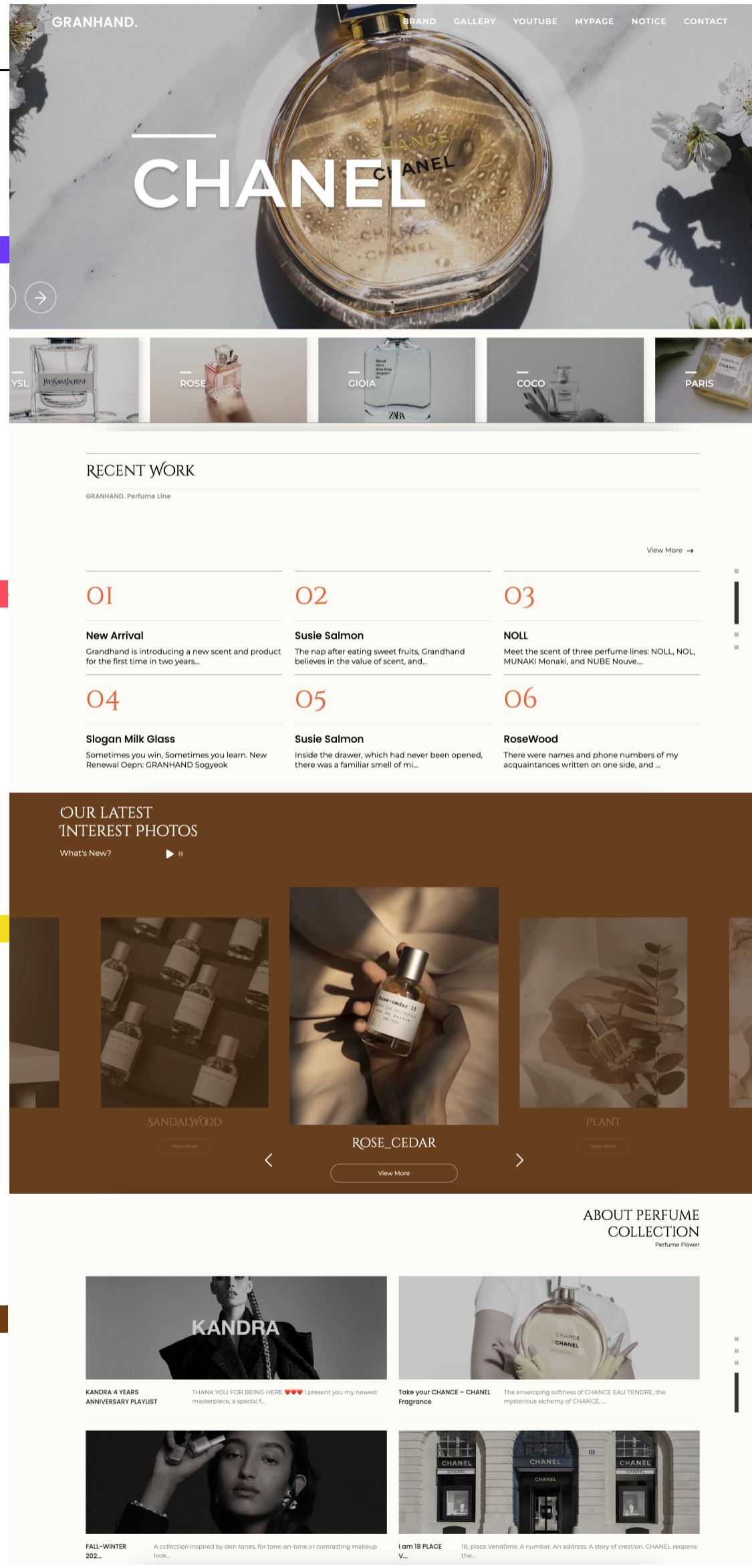
- LocalStorage에 데이터 저장 후 서브 News와 연동

### Pics

- Youtube api 데이터 연동 서브 Youtube 페이지 데이터 동일

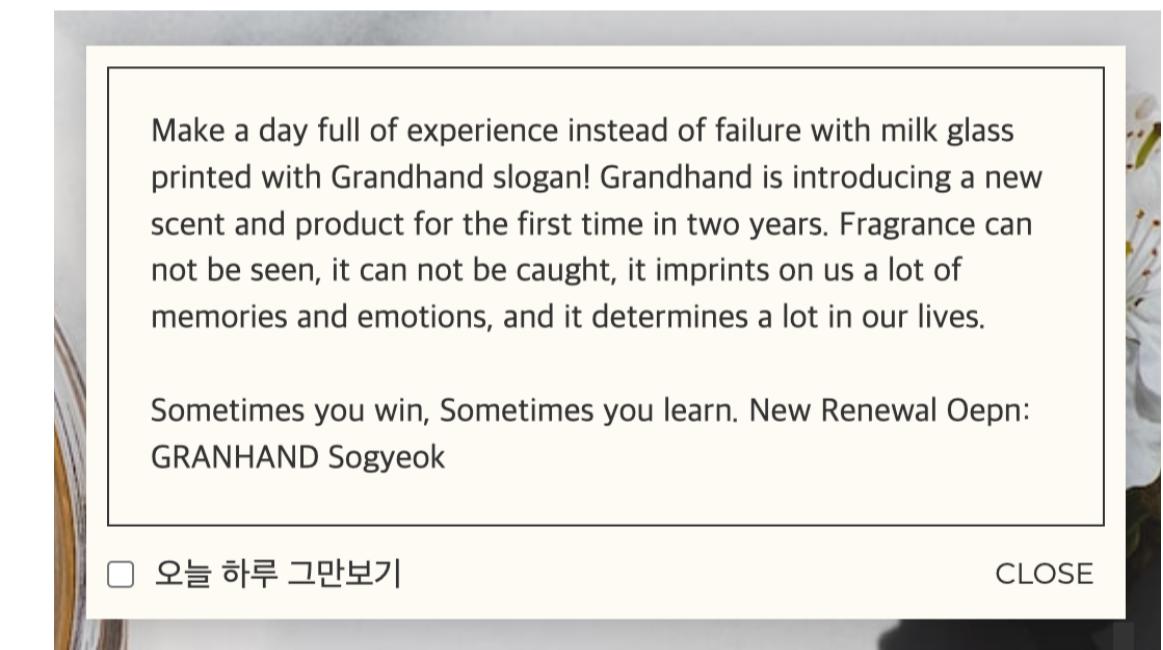
### Vids

- Flickr api 데이터 연동 서브 Gallery 페이지 데이터 동일



## Cookie 팝업

- 쿠키 : 브라우저에 물리적으로 저장되는 작은 크기의 문자열
- 쿠키 접근 : document.cookie
- 쿠키 형식 : name=value; path=/; expires=만료일



```
const isCookie = document.cookie.indexOf('cookie=done');
```

```
useEffect(() => {
  isCookie === -1 쿠키가 존재하는지 아닌지를 -1로 구분
  ? (popup.current.style.display = 'block')
    (쿠키가 없다면 보이기)
  : (popup.current.style.display = 'none');
    (쿠키가 있다면 보이지 않기)
}, [isCookie]);
```

# Sub Components : Brand

메뉴명 = 컴포넌트

components

sub

JS Brand.js

- 처음 마운트 될때 한번만 axios로 외부 데이터를 호출해서 DB/member.json파일의 데이터 받기

```
useEffect(() => {
  const [Members, setMembers] = useState([]);
  axios.get(`${process.env.PUBLIC_URL}/DB/member.json`)
    .then((json) => {
      setMembers(json.data.members);
    });
}, []);
```

- public/DB/member.json

```
{
  "members": [
    {
      "name": "Kim Nam-Gil",
      "position": "COMPANY CEO |",
      "pic": "member1.jpg",
      "mail": "namgildearo@naver.com",
      "txt": "Sometimes You Win Sometimes You Learn : When I\n          opened an old notebook with a coated four-leaf clover,\n          looked just like me in high heels in a sheep's clothing\n          smiling broadly."
    }
  ]
}
```

sub

{ } member.json

GRANHAND.

BRAND GALLERY YOUTUBE MYPAGE NOTICE CONTACT

BRAND

'Inside the drawer, which had never been opened, there was a familiar smell of minutes between scarves, accessories, and various miscellaneous things. When I opened an old notebook with a coated four-leaf clover, there were names and phone numbers of my acquaintances written on one side, and a faded photo next to it showed a young woman who looked just like me in high heels in a sheep's clothing smiling broadly.'

- NUBE, Perfume, GRANHAND.



Kim  
Nam-Gil  
COMPANY CEO |

namgildearo@naver.com → mail

Sometimes You Win Sometimes You Learn : When I opened an old  
notebook with a coated four-leaf clover, looked just like me in high  
heels in a sheep's clothing smiling broadly. → txt

Lee  
Da-Hee  
MANAGER |

dahee@gmail.com

phone numbers of my acquaintances written on one side, and a  
faded photo next to it showed a young woman who looked just like  
me in high heels.



Cha  
Eun-Woo  
EXHIBITION MANAGER |

eunwoo@gmail.com



Inside the drawer, which had never been opened, there was a familiar

# Redux

## JavaScript 상태관리 라이브러리

- useSelector로 상태값을 가져오고
- useDispatch()로 업데이트 후
- app에서 data 관리

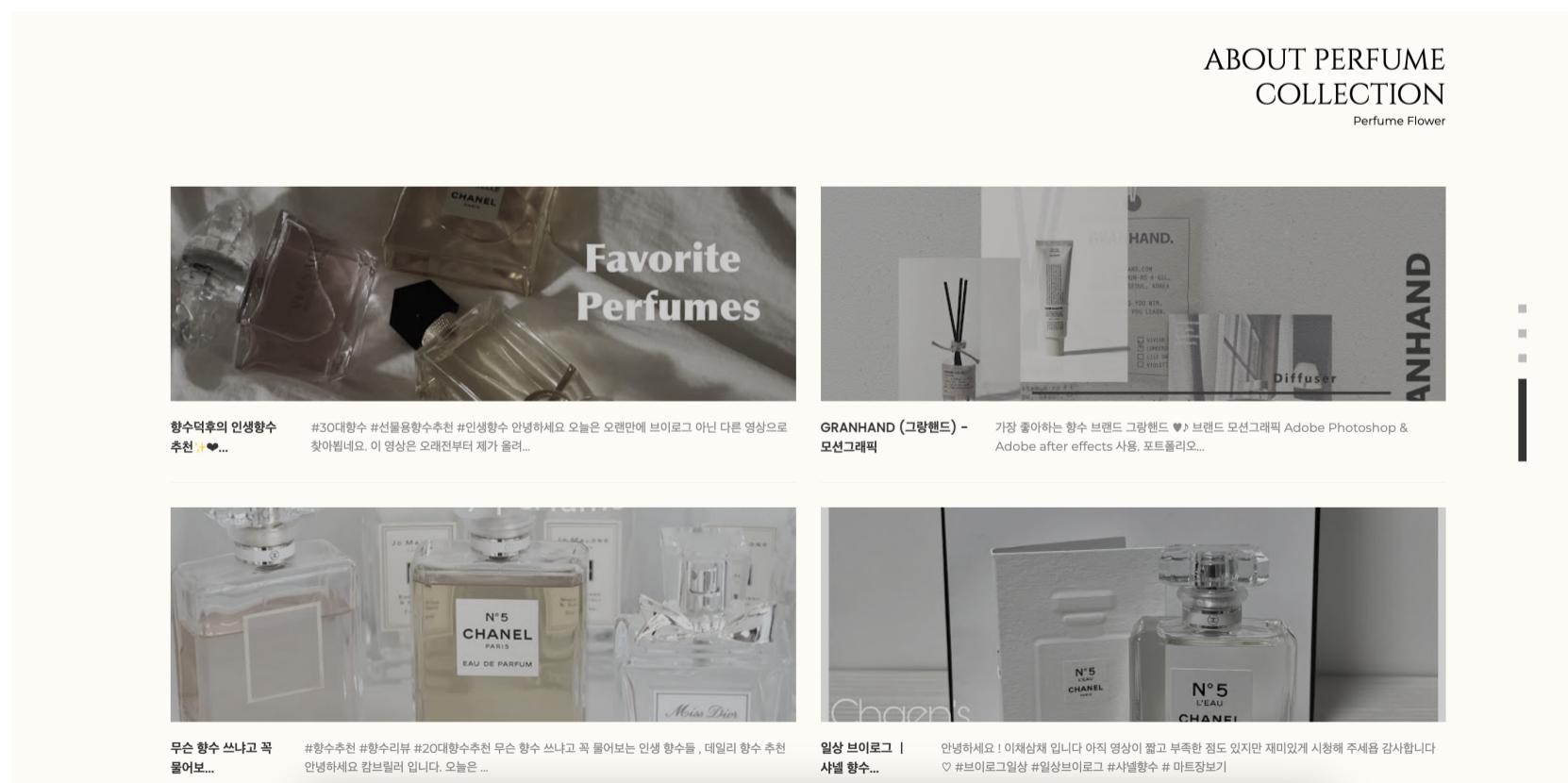
- 다른 컴포넌트에서도 data에 접근하기 위해 redux를 통해 data를 전역으로 관리

```
const Vids = useSelector((store) => store.youtubeReducer.youtube);
```

### • App.js

```
useEffect(() => {
  dispatch({ type: types.YOUTUBE.start });
  dispatch({ type: types.FLICKR.start, Opt: { type: 'user', user: '195427004@N07' } });
}, [dispatch]);
```

📁 main <Vids />



### 다른 컴포넌트에서 data연동

The diagram shows a flow of data from a main component to a sub-component, which then interacts with a YouTube page. A red arrow points from the 'main' component to the 'sub' component. Another red arrow points from the 'sub' component to the 'YouTube' page.

**GRANHAND.** BRAND GALLERY YOUTUBE MYPAGE NOTICE CONTACT

**YOUTUBE**

**KANDRA 4 YEARS ANNIVERSARY PLAYLIST**

**Kandra**  
THANK YOU FOR BEING HERE ❤️❤️ I present you my newest masterpiece, a special fashion music playlist in celebration of Kandra's...

**Take your CHANCE – CHANEL Fragrance**

**CHANEL**  
The enveloping softness of CHANCE EAU TENDRE, the mysterious alchemy of CHANCE, the invigorating freshness of CHANCE EAU FRAÎCHE, ...

**FALL-WINTER 2022 COLLECTION - CHANEL Makeup**

**CHANEL**  
A collection inspired by skin tones, for tone-on-tone or contrasting makeup looks that range from natural to bold. #FallWinter #C...

**I AM 18 PLACE VENDÔME – CHANEL Watches and Fine Jewelry**

**CHANEL**

CHANEL

CHANEL

# Redux saga

sub

JS Gallery.js

마운트시 기본 interest 실행

```
useEffect(() => {
  dispatch({ type: types.FLICKR.start, Opt });
}, [Opt, dispatch]);
```

redux

JS actionTypes.js

- actionTypes 지정

JS api.js

- option을 받아서 data요청

JS reducer.js

- actionTypes요청을 saga로 넘김

JS saga.js

- action요청을 전달 받아 youtube/flickr호출함수 실행

JS store.js

- redux를 통해 전역데이터 상태관리
  - \_ reducer데이터 전달 시 sagaMiddleware설정해서 연동

## • saga.js

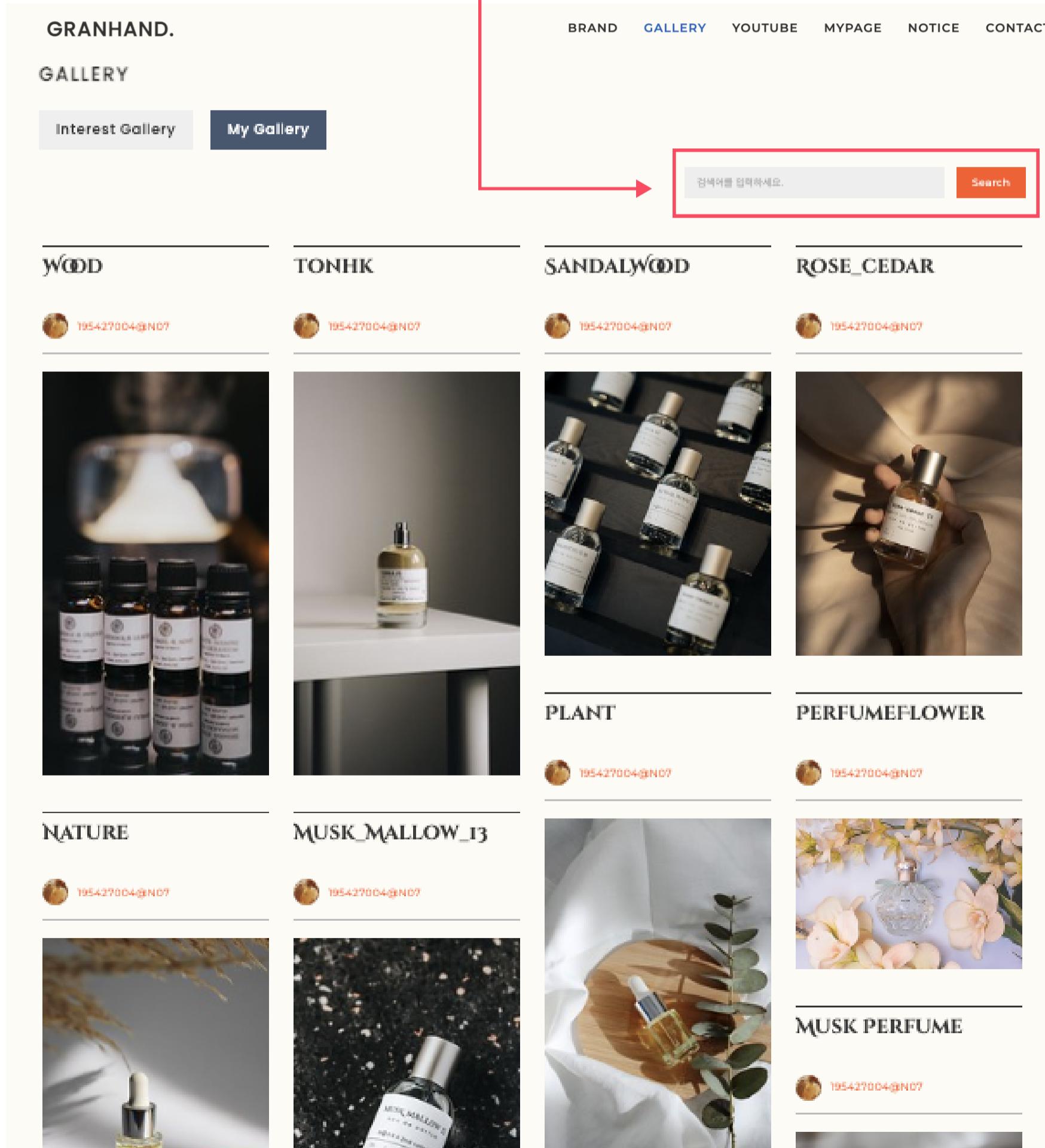
FLICKR\_START라는 액션요청을 reducer 함수를 통해서 전달받으면 Flickr 호출함수를 실행

```
function* callFlickr() {
  yield takeLatest(types.FLICKR.start, returnFlickr);
}
```

데이터 호출한 뒤 결과값을 가지고 다시 새로운 액션객체를 반환해서 reducer로 전달

```
function* returnFlickr() {
  try {
    const response = yield call(returnFlickr);
    yield put({ type: types.FLICKR.success, payload: response.data.items });
  } catch (err) {
    yield put({ type: types.FLICKR.fail, payload: err });
  }
}
```

- 버튼 클릭 시 option이 바뀌면서 dispatch로 data 전송 후 출력



# CRUD localStorage 01

## localStorage

- 각 브라우저에 있는 로컬 저장공간으로 게시글 추가/삭제/수정 기능 구현
- localStorage의 값이 있으면 해당 값을 다시 json 형태로 변경해서 반환 값이 없으면 빈 배열을 반환
- 더미데이터 내용을 초기값으로 지정

```
const getLocalData = () => {
  const data = localStorage.getItem('post');

  if (data) return JSON.parse(data);
  else return [];
};
```

- Posts 값이 변경될 때마다 해당 데이터를 문자화해서 다시 localStorage에 저장

```
useEffect(() => {
  localStorage.setItem('post', JSON.stringify(Posts));
}, [Posts]);
```

### JS Notice.js

#### NOTICE

Inside the drawer, which had never been opened, there was a familiar smell of minutes between scarves, accessories, and various miscellaneous things.

Please enter a title

Please enter the Comments

CANCEL WRITE

#### Newest

01  
Inside the drawer,  
there was a familiar smell of minutes between scarves, accessories,  
and various miscellaneous things.  
[EDIT](#) [DELETE](#)

02  
New Arrival  
Grandhand is introducing a new scent and product for the first time  
in two years. Fragrance can not be seen, it can not be caught, it  
imprints on us a lot of memories and emotions, and it determines a  
lot in our lives.  
[EDIT](#) [DELETE](#)

03

Susie Salmon  
The nap after eating sweet fruits, Grandhand believes in the value of  
scent, and dreams of becoming a daily life of scent through  
constant attempts through it. It imprints a lot of memories and  
emotions, and it further determines a lot of things in our lives.  
[EDIT](#) [DELETE](#)

04

NOLL  
Meet the scent of three perfume lines: NOLL, NOL, MUNAKI Monaki,  
and NUBE Nouve. Grandhand wants to convey our hearts and  
warmth to everyone with a strong presence in invisible, not  
something that will be easily  
[EDIT](#) [DELETE](#)

05

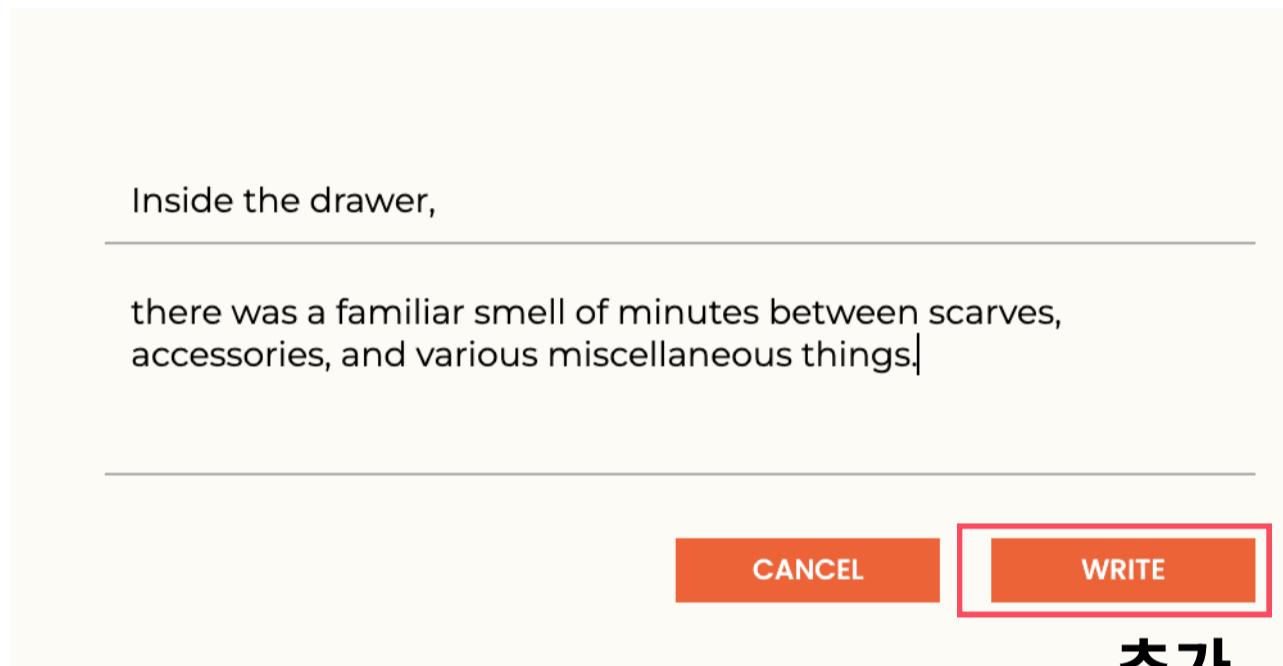
Slogan Milk Glass  
Sometimes you win, Sometimes you learn. New Renewal Open:  
GRANHAND Sogyeok  
[EDIT](#) [DELETE](#)

06

Susie Salmon  
Inside the drawer, which had never been opened, there was a  
familiar smell of minutes between scarves, accessories, and various  
miscellaneous things. When I opened an old notebook with a  
coated four-leaf clover  
[EDIT](#) [DELETE](#)

# CRUD localStorage 02

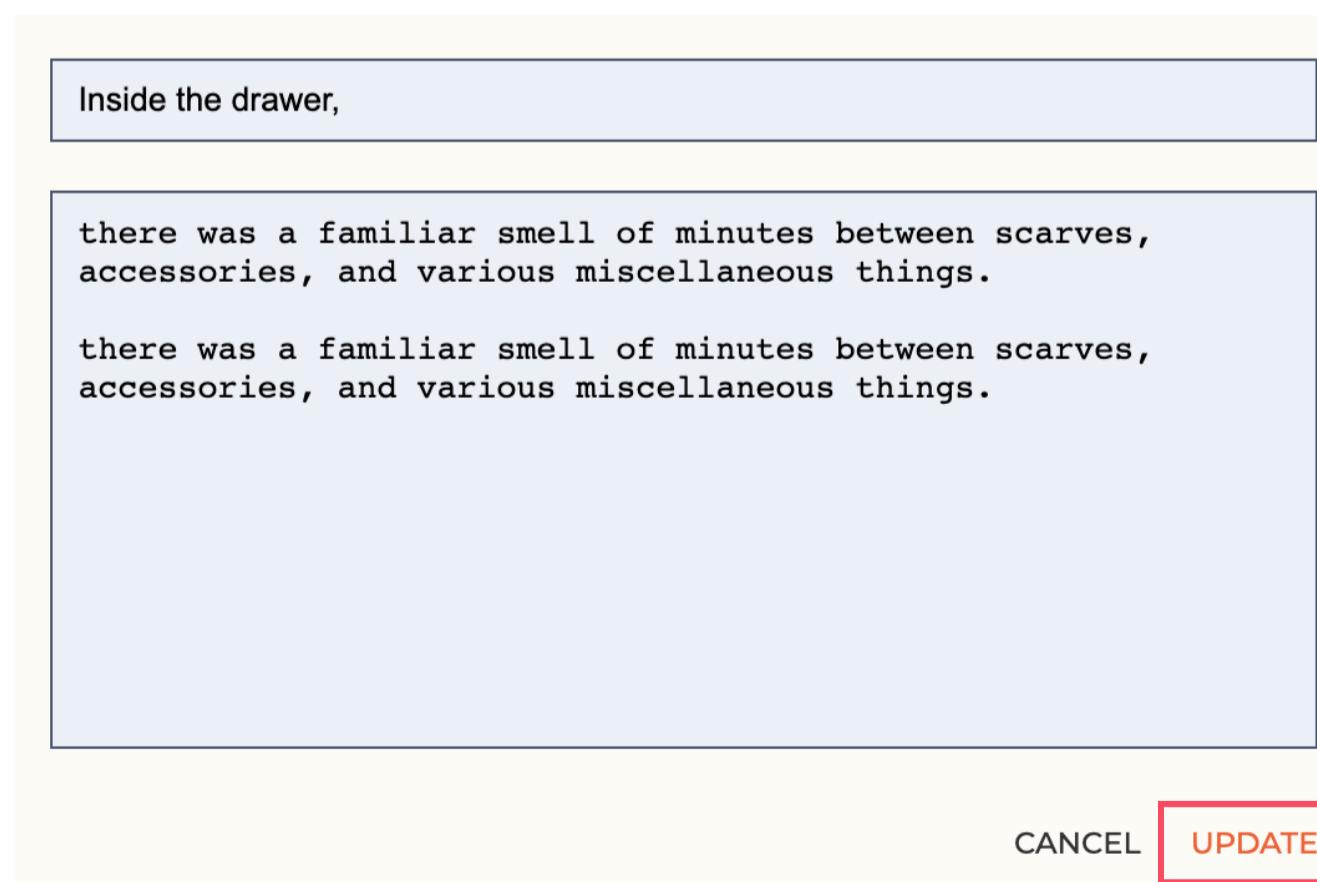
## 1) 게시글 작성 후



setPosts([{새게시글}, ...기존게시글})

```
const createPost = () => {
  if (!input.current.value.trim() || !textarea.current.value.trim()) {
    resetForm();
    return alert('제목과 본문을 모두 입력하세요');
  }
  setPosts([{ title: input.current.value, content: textarea.current.value }, ...Posts]);
  resetForm();
};
```

## 2) 게시글 수정



반복도는 post순번과 수정할 포스트 순번이 같으면 수정모드 출력

```
const updatePost = (updateIndex) => {
  if (!inputEdit.current.value.trim() || !textareaEdit.current.value.trim()) {
    return alert('수정할 제목과 본문을 모두 입력하세요');
  }
  setPosts(
    Posts.map((post, postIndex) => {
      if (postIndex === updateIndex) {
        post.title = inputEdit.current.value;
        post.content = textareaEdit.current.value;
        post.enableUpdate = false;
      }
      return post;
    })
  );
  setAllowed(true);
};
```

# CRUD localStorage 03

## 3) 게시글 추가

Inside the drawer,  
there was a familiar smell of minutes between scarves, accessories,  
and various miscellaneous things.

EDIT    DELETE

📁 sub

JS Notice.js

반복되는 순번이랑 파라미터로 전달된 index순번이 같지 않은것들만 반환  
**filter 자체가 기존 배열을 deep copy하면서 새롭게 필터링된 배열을 반환**  
따라서 기존 posts를 전개 연산자로 카피할 필요가 없다.



```
const deletePost = (delIndex) => {
  if (!window.confirm('해당 게시글을 삭제하시겠습니까?'))
    return;
  setPosts(Posts.filter(_, postIndex) => postIndex !== delIndex);
};
```

## 메인페이지에 최신글 4개만 출력

localStorage에 데이터 담고, JSON형태로 변환 후 반환

```
const getLocalData = () => {
  const data = localStorage.getItem('post');
  return JSON.parse(data);
};

const [Posts] = useState(getLocalData());
```



📁 main

JS News.js

O1

Inside the drawer,

there was a familiar smell of minutes  
between scarves, accessories, and various ...

O2

New Arrival

Grandhand is introducing a new scent and  
product for the first time in two years...

# Mypage : Login

- name값을 객체로 담기
- 회원가입 완료시 메인페이지로 이동

```
const initVal = {
  userid: '',
  pwd1: '',
  pwd2: '',
  email: '',
  sns: false,
  perfume: false,
  comments: '',
  age: '',
  mail: '',
};
```

1

- value 값 체크

```
const check = (value) => {
  const errs = {};
  const eng = /[a-zA-Z]/;
  const num = /[0-9]/;
  const spc = /[~!@#$%^&*()]/;

  if (value.userid.length < 5) {errs.userid = '아이디를 5글자 이상 입력하세요';}

  if (value.pwd1.length < 5 || !eng.test(value.pwd1) || !num.test(value.pwd1) || !spc.test(value.pwd1))
    ) {errs.pwd1 = '비밀번호는 5글자 이상, 영문, 숫자, 특수문자를 모두 포함하세요';}

  if (value.pwd1 !== value.pwd2 || !value.pwd2) {errs.pwd2 = '두개의 비밀번호를 동일하게 입력하세요';}

  if (value.email === "" || value.mail === "") {errs.email = '이메일 주소를 입력하세요';}

  if (value.email === "" || value.mail === "") {errs.mail = '이메일 주소를 입력하세요';}

  if (!value.gender) {errs.gender = '성별을 선택하세요.';}

  if (!value.perfume) {errs.perfume = '좋아하는 종류를 하나만 선택해주세요.';}

  if (value.comments.length < 20) {errs.comments = '20글자 이상 입력하세요';}

  if (value.age === "") {errs.age = '연령을 선택하세요';}

}

return errs;
```

3

## JS Mypage.js

- 조건에 부합하지 않으면 errs 반환

The screenshot shows a login form with the following fields and validation messages:

- USER ID:** 아이디를 입력하세요 (Input field)
- PASSWORD:** 비밀번호를 입력하세요 (Input field)
- RE-PASSWORD:** 비밀번호를 재입력하세요 (Input field)
- E-MAIL:** 이메일 주소를 입력하세요 (Input field)
- GENDER:** 성별을 선택하세요. (Radio buttons: Male, Female)
- KIND OF:** 좋아하는 종류를 하나만 선택해주세요 (Checkboxes: Citrus, Floral, Marine, Musk)
- Age:** 연령을 선택하세요 (Select dropdown)
- COMMENTS:** 남기는 말을 입력하세요 (Input field)

Validation errors are displayed in red text:

- User ID: 아이디를 5글자 이상 입력하세요
- Password: 비밀번호는 5글자 이상, 영문, 숫자, 특수문자를 모두 포함하세요
- Re-Password: 두개의 비밀번호를 동일하게 입력하세요
- Email: 이메일 주소를 입력하세요
- Gender: 성별을 선택하세요.
- Kind of: 좋아하는 종류를 하나만 선택해주세요
- Age: 연령을 선택하세요
- Comments: 20글자 이상 입력하세요

Buttons at the bottom: **cancel** and **send**.

# Contact : Location

- 지도의 위치값과 줌 레벨이 있는 옵션 객체 담기

```
option.current = {
  center: info.current[Index].latlng,
  level: 3,
};
```

- 지점 정보를 배열로 묶고 화면에 출력

```
info.current = [
{
  title: 'Mapo',
  latlng: new kakao.maps.LatLng(37.5559436, 126.9302956),
  imgUrl: `${process.env.PUBLIC_URL}/img/contact(marker1.png`,
  imgSize: new kakao.maps.Size(232, 73),
  imgPos: { offset: new kakao.maps.Point(116, 36) },
  add: '9, Wausan-ro 37-gil, Mapo-gu, Seoul',
  tel: '02-333-6525' },
{
  title: 'SeoCheon',
  latlng: new kakao.maps.LatLng(37.5771945, 126.9730108),
  imgUrl: `${process.env.PUBLIC_URL}/img/contact(marker2.png`,
  imgSize: new kakao.maps.Size(232, 73),
  imgPos: { offset: new kakao.maps.Point(116, 36) },
  add: '14-2, Jahamun-ro 4-gil, Jongno-gu, Seoul',
  tel: '02-333-6525' },
{
  title: 'NamSan',
  latlng: new kakao.maps.LatLng(37.5489259, 126.9822293),
  imgUrl: `${process.env.PUBLIC_URL}/img/contact(marker3.png`,
  imgSize: new kakao.maps.Size(232, 73),
  imgPos: { offset: new kakao.maps.Point(116, 36) },
  add: '358-194, Huam-dong, Yongsan-gu, Seoul',
  tel: '02-333-6525' },
];
```

Traffic OFF



Mapo

9, Wausan-ro 37-gil, Mapo-gu, Seoul  
02-333-6525

SeoCheon

14-2, Jahamun-ro 4-gil, Jongno-gu, Seoul  
02-333-6525

NamSan

358-194, Huam-dong, Yongsan-gu, Seoul  
02-333-6525

- 버튼 클릭시 Traffic state값 변경

Traffic ON



```
<button ref={txt} onClick={() => setTraffic(!Traffic)}>
  {Traffic ? 'Traffic ON' : 'Traffic OFF'}
</button>
```