

J O M A L O N E
L O N D O N

조말론 런던 리액트 네이티브 안드로이드 앱 4차 프로젝트
발표

CONTENTS

- 
- 01**
기획
 - 1. 선정 배경
 - 2. 마인드 맵
 - 3. 사이트 맵
 - 02**
요구사항 정의서
 - 03**
프로젝트
정보
 - . 시간 계획서
 - 2. 작업도구
 - 3. 역할 분담
 - 04**
시연화면

01

기획

대한민국은 '향수홀릭'...1년간 1만6000리터 쏟아부었다 [최수진의 패션채널]

그런데 요즘은 '립스틱'이 스몰럭셔리를 대표하지 않습니다. 그 자리를 '향수'가 차지했거든요. 특히, 문 조향사가 소수를 위해 만든 프리미엄 향수인 '니치향수'가 스몰럭셔리의 대표 제품으로 젊은 소비자들의 관심을 받고 있습니다. 3~5만원대 립스틱보다는 적게는 5배, 많게는 10배 이상 비싸면서도 고가의 의류나 가방에 비해 부담이 덜해 딱 인스타그램에 '인증샷'을 남기기 좋다는 거죠.

글로벌 시장조사업체 유로모니터에 따르면 국내 향수 시장은 2019년 6000억원 규모였는데, 오는 2025년 9800억원 수준까지 커진다고 합니다. 특히, 니치향수에 대한 관심이 늘어나고 있는데요. 한해에 팔리는 향수 10개 가운데 9개는 니치향수라고 합니다.

그래서 얼마나 잘 되냐고요? 지난해 팔린 니치향수는 1만6000리터(?)라고 합니다. 향수 1병당 평균 용량 75ML를 기준으로 했을 때 평균을 낸 겁니다. 전국 2030세대 인구인 1270만명이 1인당 13번씩(향수 1회 분사 시 사용량 0.1ml 기준) 뿌릴 수 있는 양인데요. 갯수로 따지면 약 21만병이 판매된 셈입니다.

해외 여행이 증가하면서
면세를 통해 향수 수요 증가

↓
향수가 스몰럭셔리 제품으로 자리 잡게 됨

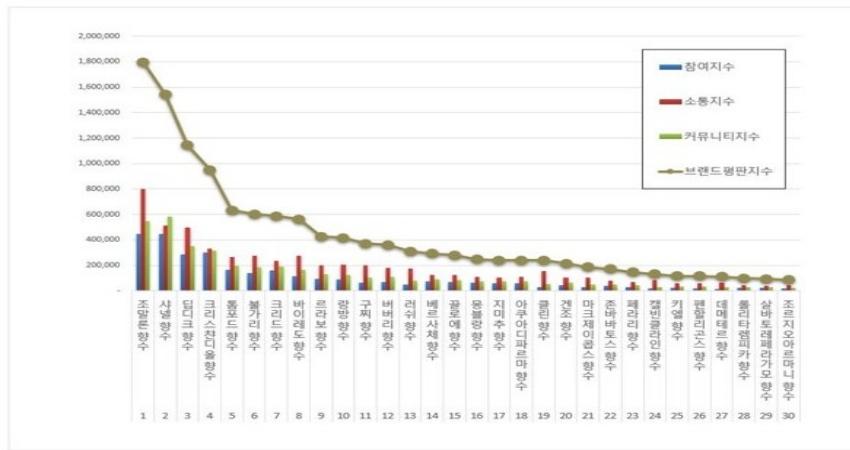
따라서 향수를 주제로 선정

01 기획 선정 배경

HOME > 경제

[브랜드평판] 향수 2024년 9월 빅데이터 1위 조말론 향수, 2위 샤넬 향수, 3위 딥디크 향수

박준영 기자 | 승인 2024.09.12 10:02



브랜드에 대한 평판은 브랜드에 대한 소비자들의 활동 빅데이터를 참여가치, 소통가치, 소셜가치, 시장가치, 재무가치로 나누게 된다. 향수 브랜드평판지수는 참여지수, 소통지수, 커뮤니티지수로 분석했다.

브랜드평판지수는 소비자들의 온라인 습관이 브랜드 소비에 큰 영향을 끼친다는 것을 찾아내서 브랜드 빅데이터 분석을 통해 만들어진 지표로 브랜드에 대한 긍부정 평가, 미디어 관심도, 소비자의 소통량, 커뮤니티 노출량, 소셜에서의 대화량, 소비자와 브랜드의 디지털 트랜스포메이션 관계분석으로 측정된다. 브랜드평판 분석에는 브랜드 영향력을 측정한 브랜드 가치평가 분석과 브랜드평판 모니터의 정성평가도 포함했다.

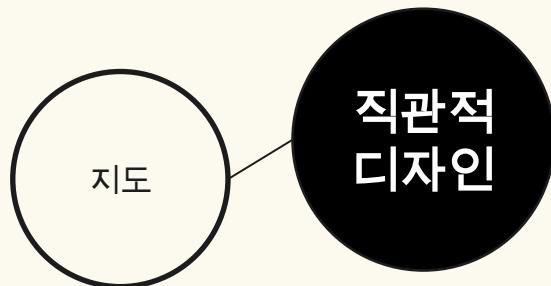
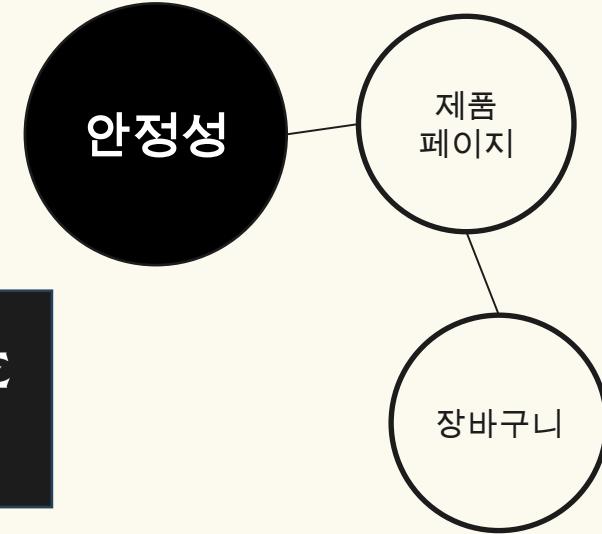
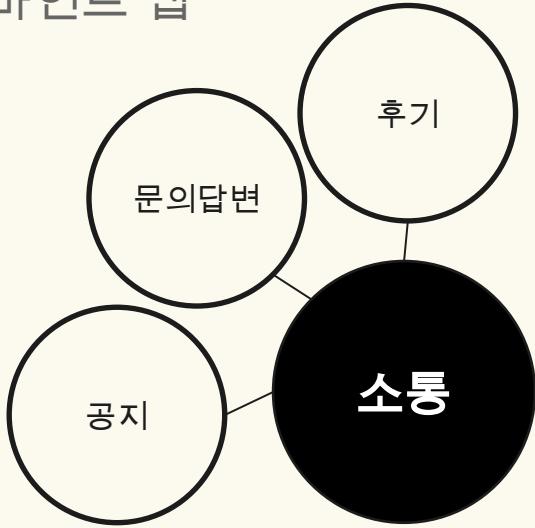
조말론의 높은 브랜드 평판

브랜드 평판 지수는 소비자들의 온라인 습관을 토대로 만들어진 지표

조말론에 대한 관심도 증가

↓
조말론 앱을 주제로 선정

01 기획 마인드 맵



01 기획 사이트 맵

헤더	메인화면
SNS	슬라이드
인스타그램 카카오채널 유튜브	섹션1 2 3

메인메뉴				
브랜드소개	제품	공지사항	문의답변	후기
지도	제품리스트	공지사항 리스트	문의답변 리스트	후기 리스트
	제품상세	공지사항 글읽기 공지사항 글쓰기	문의답변 글읽기 문의답변 글쓰기	후기 글읽기 후기 글쓰기
	장바구니			
	결제하기			

02

요구사항
정의서

02 요구사항 정의서 메인화면, 헤더

기능이름	세부	내용	우선순위	중요도	전제조건
메인화면	메인 슬라이드	양쪽 화살표 클릭 시 이미지 슬라이드	5	하	메인
	슬라이드	제품 클릭 시 상품 상세 페이지로 이동	2	중	메인 > 베스트셀러 > 상품리스트
	네비게이션	브랜드소개, 공지사항 클릭 시 페이지 이동	3	중	메인 > 브랜드소개 메인 > 공지사항
	헤더	로고, 장바구니, 햄버거 메뉴 이동	3	중	메인 > 햄버거 > 메뉴 클릭 이동 메인 > 로고 메인 > 장바구니
헤더	로고	로고 누르면 메인 페이지로 이동	4	하	헤더 > 로고
	모바일GNB	모달이 열리고 각 메뉴로 이동하는 버튼 노출되어 클릭하면 각 화면으로 이동	2	중	헤더 > 햄버거버튼
	장바구니 (리스트)	Firebase에서 데이터를 가져와 제품의 이름, 가격을 출력. 개수를 늘리고 줄일 수 있으며, 해당 상품 리스트를 삭제 가능	1	상	헤더> 장바구니아이콘

02 요구사항 정의서 제품, 후기게시판

기능이름	세부	내용	우선순위	중요도	전제조건
제품	목록	제품을 목록 형태로 출력. '자세히보기' 클릭 시 상세로 이동	3	중	헤더>햄버거버튼>제품
	상세	제품 상세 설명 출력. 용량별 작은 이미지 누르면 큰 이미지와 가격도 용량별로 맞춰서 변경. 장바구니 버튼 누르면 장바구니에 제품 추가	2	중	헤더>햄버거버튼>제품>자세히보기
후기게시판	목록	Firebase에서 데이터를 가져와 후기 글 목록으로 출력	3	중	헤더>햄버거버튼> 후기
	글쓰기	후기글 작성 후 Firebase로 데이터 저장	3	중	헤더>햄버거버튼> 후기>글쓰기버튼
	상세보기	Firebase에서 데이터를 가져와 후기글 내용 상세 보기	3	중	헤더>햄버거버튼> 후기>글제목
	수정하기	글 제목, 내용 수정 후 Firebase로 데이터 업데이트	3	중	헤더>햄버거버튼> 후기>글제목>수정
	삭제하기	버튼 클릭하면 Firebase에서 데이터 삭제	3	중	헤더>햄버거버튼> 후기>글제목>삭제

02 요구사항 정의서 브랜드 소개, 공지사항 게시판

기능이름	세부	내용	우선순위	중요도	전제조건
브랜드 소개	지도	조말론 브랜드 소개와 매장안내로 Google 지도 API를 연결	4	하	메인 > 햄버거버튼 > 브랜드 소개
	리스트	Google 지도 API를 연결하여 조 말론 모든 매장의 위치 정보 출력	4	하	메인 > 브랜드 소개
공지사항 게시판	목록	Firebase에서 데이터를 가져와 공지사항 글 목록으로 출력	3	중	헤더>햄버거버튼>공지사항
	글쓰기	공지사항 글 작성 후 Firebase로 데이터 저장	3	중	헤더>햄버거버튼>공지사항>글쓰기버튼
	상세보기	Firebase에서 데이터를 가져와 공지사항 글 내용 상세 보기	3	중	헤더>햄버거버튼>공지사항>글제목
	수정하기	글 제목, 내용 수정 후 Firebase로 데이터 업데이트	3	중	헤더>햄버거버튼>공지사항>글제목>수정
	삭제하기	버튼 클릭하면 Firebase에서 데이터 삭제	3	중	헤더>햄버거버튼>공지사항>글제목>삭제

02 요구사항 정의서 문의 게시판, 소셜미디어

기능이름	세부	내용	우선순위	중요도	전제조건
문의 게시판	목록	Firebase에서 데이터를 가져와 문의 글 목록으로 출력	3	중	헤더>햄버거버튼> 문의
	글쓰기	문의 글 작성 후 Firebase로 데이터 저장	3	중	헤더>햄버거버튼> 문의>글쓰기버튼
	상세보기	Firebase에서 데이터를 가져와 문의 글 내용 상세 보기	3	중	헤더>햄버거버튼> 문의>글제목
	수정하기	글 제목, 내용 수정 후 Firebase로 데이터 업데이트	3	중	헤더>햄버거버튼> 문의>글제목>수정
	삭제하기	버튼 클릭하면 Firebase에서 데이터 삭제	3	중	헤더>햄버거버튼> 문의>글제목>삭제
소셜 미디어	인스타그램	아이콘 클릭하면 조말론 인스타그램 계정으로 웹뷰 이동	4	하	헤더>햄버거버튼> 인스타그램 아이콘
	카카오채널	아이콘 클릭하면 조말론 카카오톡 채널로 웹뷰 이동	4	하	헤더>햄버거버튼> 말풍선 아이콘
	유튜브	아이콘 클릭하면 조말론 유튜브 채널로 웹뷰 이동	4	하	헤더>햄버거버튼> 유튜브 아이콘

03

프로젝트 정보

03 프로젝트 정보

시간계획서01

우선순위	기능	1월				2월			
		22	23	24	31	3	4	5	6
-	역할 분배								
1	장바구니(리스트)								
2	슬라이드								
2	모바일GNB								
2	제품상세								
3	네비게이션								
3	헤더								
3	후기 게시판								
3	공지사항 게시판								

발표

03 프로젝트 정보

시간계획서02

우선순위	기능	1월				2월			
		22	23	24	31	3	4	5	6
3	문의 게시판								
3	제품목록								
4	로고								
4	브랜드 소개 지도								
4	브랜드 소개 리스트								
4	소셜미디어 인스타그램								
4	소셜미디어 카카오채널								
4	소셜미디어 유튜브								

발표

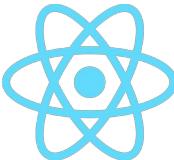
03 프로젝트 정보

작업도구

개발



VS CODE
v1.96.3



React v18.3.1
React Native v0.77



Node.js
v22.13.1



Android Studio
2024.2.2.13
windows



Gradle Build Tool
8.12.1



Firebase
v21.7.1

협업



Sourcetree

Sourcetree
v 3.4.19



Google Docs



Github



Kakaotalk

기타



Window 10, 11



React Navigation v7.x



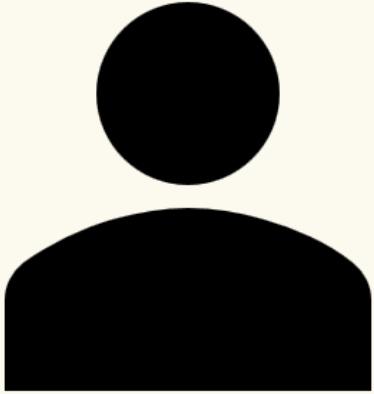
Swiper v11.2.2



Fontawesome v6.7.2

03 프로젝트 정보

역할분담

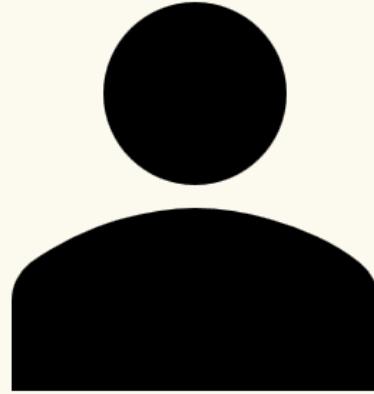


이경근

장바구니

공지사항 게시판

브랜드소개 (지도)

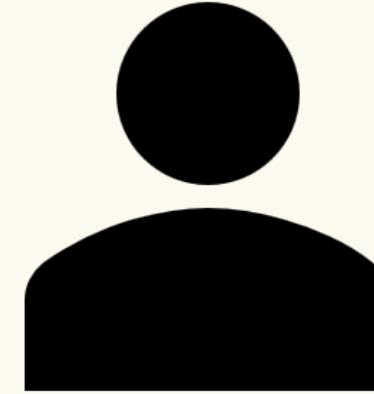


손주혜

헤더

후기 게시판

제품리스트
제품 상세

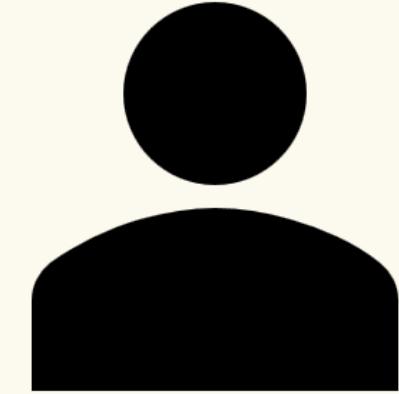


우수정

메인화면 (메인슬라이드)

문의 게시판

소셜미디어
(인스타그램, 카카오채널, 유튜브)



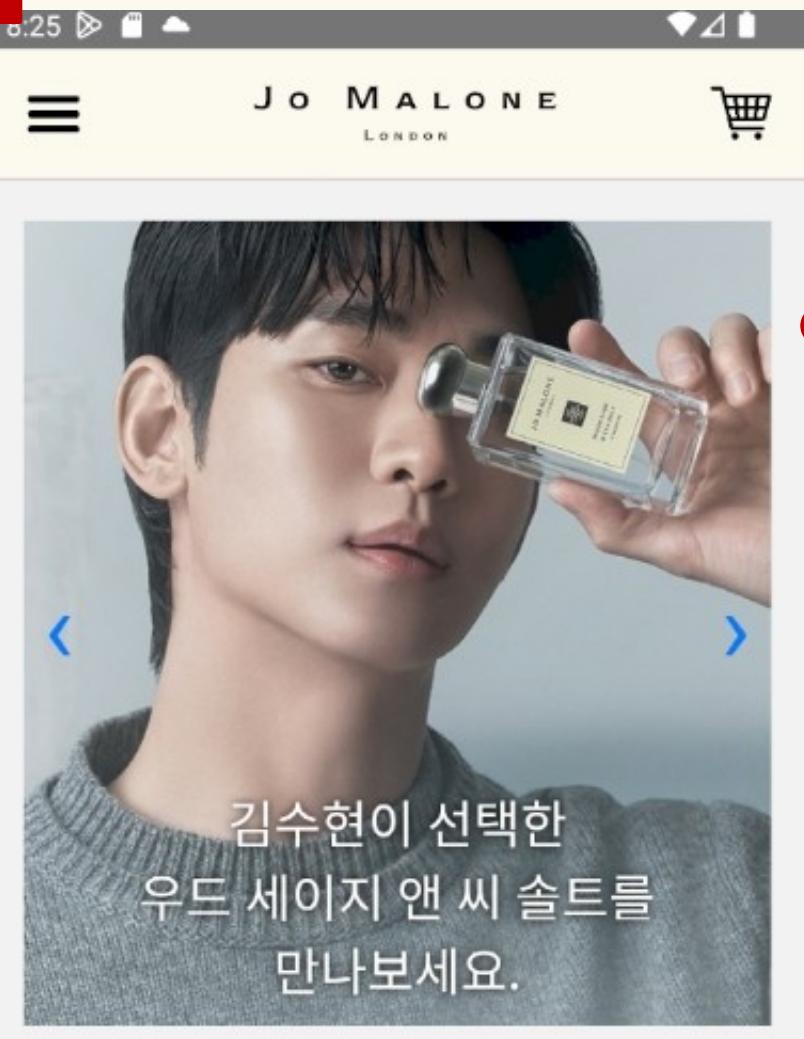
이규만

메인화면

04

시연화면

1



1

```
function MainSwiper(props) {
  const imgs = [
    {uri : 'https://www.jomalone.co.kr/media/export/cms/homepage/2025/202501/Homepage_1_mob_wood-sage-sea-salt.jpg'},
    {uri : 'https://www.jomalone.co.kr/media/export/cms/homepage/2025/202501/Homepage_1_mob_personalization.jpg'},
    {uri : 'https://www.jomalone.co.kr/media/export/cms/homepage/2025/202502/Homepage_3_mob_Cologne_best.jpg'}
  ]
  const swiperTxt = [
    '김수현이 선택한\n우드 세이지 앤 씨 솔트를\n만나보세요.',
    '알파벳 참으로\n나만의 코롱을\n만들어보세요.',
    '베스트셀러 기프트',
  ]
}
```

2

```
return (
  <View style={styles.wrapper}>
    <Swiper
      containerStyle={{width:'100%', height:400}}
      loop={true}
      autoplayTimeout={7}
      autoplay={true}
      showsPagination={false}
      showsButtons
    >
      {imgs.map((item, i)=>(
        <View key={i} style={styles.imgWrap}>
          <Text style={styles.swiperTxt}>{swiperTxt[i]}
```

3

```
const styles = StyleSheet.create({
  wrapper: {
    flex:1,
    justifyContent: 'center',
    alignItems: 'center',
  },
  imgWrap: {
    position:'relative'
  },
  img: {
    width:'100%',
    height:'100%',
    resizeMode:"cover"
  },
  swiperTxt: {
    position: 'absolute',
    color:'#fff',
    fontSize: 25,
    textAlign: 'center',
    left: 0,
    width: '100%',
    bottom: 0,
    zIndex: 1,
    paddingVertical: 10,
    paddingHorizontal: 15,
    textShadowColor:'#000',
    textShadowOffset: { width: 0, height: 0 },
    textShadowRadius: 10,
  }
})
```

1

브랜드 소개

조말론 런던을 유래와 역사를 소개합니다.



```
1 <View style={styles.container}>
  <MainSwiper slides={slides} />
  <View style={styles.contentBox}>
    <Text style={styles.titleTxt}>브랜드 소개</Text>
    <Text style={styles.detailTxt}>조말론 런던을 유래와 역사를 소개합니다.</Text>

    <TouchableOpacity style={styles.contentBox} onPress={() => navigation.navigate("Brand")}>
      <Image source={require('./img/main/main.jpg')} style={styles.container} resizeMode="contain"/>
    </TouchableOpacity>
  </View>
2 contentBox: {
  width: "100%",
  height: 500,
  marginLeft: 0,
  marginRight: 0,
  textAlign: "center",
  alignItems: "center",
},

container: {
  width: '100%',
  marginLeft: 0,
  marginRight: 0,
}

titleTxt: {
  fontSize: 25,
  marginTop: 30,
  marginBottom: 0,
  color: "#333",
  textAlign: "center",
}

detailTxt: {
  fontSize: 14,
  marginTop: 10,
  textAlign: 'center',
},
```

1

베스트셀러

조말론 런던의 핫한 상품을 만나보세요.



<

>

Grapefruit Cologne

그레이프프루트 코롱

100ml

₩ 235,000



```

1 <TouchableOpacity style={styles.contentBestsellerBox} onPress={() => navigation.navigate("ProductDetail")}>
    <Text style={styles.titleTxt}>베스트셀러</Text>
    <Text style={styles.detailTxt}>조말론 런던의 핫한 상품을 만나보세요.</Text>

    <Swiper
        from={1}
        minDistanceForAction={0.1}
        containerStyle={{ width: 300, height: 300, padding: 0, margin: 0 }}
        loop={true}
        controlsEnabled={true}
        controlsProps={{
            dotsTouchable: true,
            dotActiveStyle: { backgroundColor: 'black' },
            prevPos: 'left',
            nextPos: 'right',
            nextTitle: '>',
            nextTitleStyle: { color: '#777', fontSize: 24, fontWeight: '500' },
            PrevComponent: ({ onPress }) => {
                <TouchableOpacity onPress={onPress}>
                    <Text style={{ color: '#777', fontSize: 24, fontWeight: '500' }}>{'<'}</Text>
                </TouchableOpacity>
            },
        }}
    >
        {slides.map((item, i) => (
            <View key={i}>
                <Image source={item.img} style={styles.img} resizeMode="contain" />
                <View>
                    <Text style={styles.slideTitleTxt}>{item.title}</Text>
                    <Text style={styles.descriptionTxt}>{item.description}</Text>
                    <Text style={styles.volumeTxt}>{item.volume}</Text>
                    <Text style={styles.priceTxt}>{item.price}</Text>
                </View>
            </View>
        ))}
    </Swiper>
</TouchableOpacity>

```

3

```

titleTxt: {
    fontSize: 25,
    marginTop: 30,
    marginBottom: 0,
    color: "#333",
    textAlign: "center",
},
detailTxt: {
    fontSize: 14,
    marginTop: 10,
    textAlign: "center",
},
slideTitleTxt: {
    fontSize: 18,
    marginBottom: 5,
    textAlign: "center",
},
descriptionTxt: {
    fontSize: 14,
    color: "#666",
    marginBottom: 5,
    textAlign: "center",
},
volumeTxt: {
    fontSize: 12,
    color: "#999",
    marginBottom: 5,
    textAlign: "center",
},
priceTxt: {
    fontSize: 16,
    fontWeight: "bold",
    color: "#000",
    textAlign: "center",
},

```

1

공지사항

조말론 런던의 공지사항입니다.



1

```
View style={styles.contentNoticeBox}>
  <Text style={styles.titleTxt}>공지사항</Text>
  <Text style={styles.detailTxt}>조말론 런던의 공지사항입니다.</Text>
  <TouchableOpacity onPress={() => navigation.navigate("NoticeNavi")}>
    <Image source={require('./img/main/notice.png')} style={styles.container} resizeMode="contain"/>
  </TouchableOpacity>
</View>
```

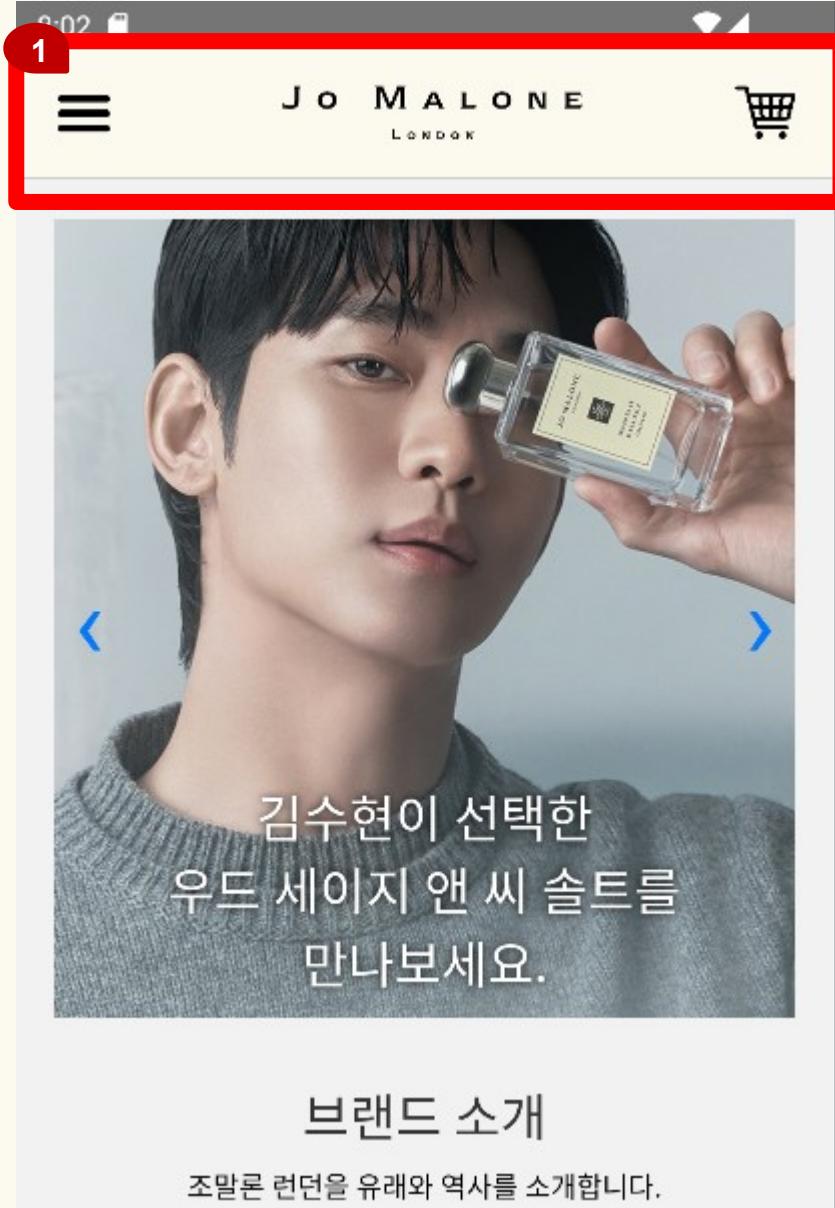
2

```
contentNoticeBox: {
  width: '100%',
  height: 450,
  marginTop: 10,
  marginBottom: 10,
  margin: 'auto',
},

titleTxt: {
  fontSize: 25,
  marginTop: 30,
  marginBottom: 0,
  color: "#333",
  textAlign: "center",
},

detailTxt: {
  fontSize: 14,
  marginTop: 10,
  textAlign: 'center',
},

container: {
  width: '100%',
  marginLeft: 0,
  marginRight: 0,
},
```

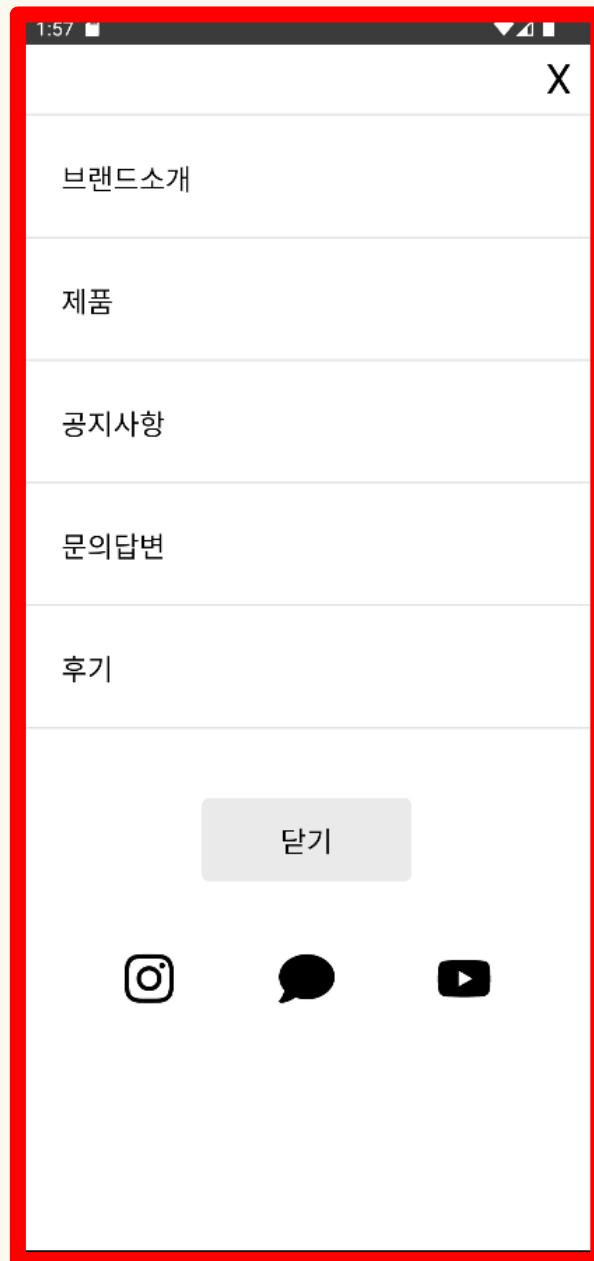


```

1 1
1 const MainHeader = ({ navigation }) => {
2
3   const [modalVisible, setModalVisible] = useState(false);
4
5   return (
6     <View style={styles.header}>
7       /* <TouchableOpacity style={styles.logo}>
8         <Text style={{textAlign:'center',lineHeight:45}}>LOGO</Text>
9       </TouchableOpacity> */
10
11    <TouchableOpacity style={styles.logo} onPress={() => navigation.navigate('Main')}>
12      <Image source={require('../img/common/logo.png')}style={styles.logoImg} resizeMode="contain"/>
13    </TouchableOpacity>
14
15    <TouchableOpacity style={styles.gnbBtn} onPress={()=>setModalVisible(true)}>
16      <Image source={require('../img/common/gnbBtn.png')}style={styles.gnbImg} resizeMode="contain"/>
17    </TouchableOpacity>
18
19    <TouchableOpacity style={styles.cart} onPress={()=> navigation.navigate('Cart')}>
20      <Image source={require('../img/common/cart.png')}style={styles.cartImg} resizeMode="contain"/>
21    </TouchableOpacity>
22
23    <Modal Visible={modalVisible}>
24      onRequestClose={()=>console.log('모달을 닫습니다.')}
25      onShow={()=>console.log('모달이 열립니다.')}
26    >
27      <View style={styles.modalWrap}>
28        <TouchableOpacity style={styles.menuCloseBtn1} onPress={()=>setModalVisible(false)}>
29          <Text style={styles.menuCloseBtn1Text}>X</Text>
30        </TouchableOpacity>
31
32        <View style={styles.menuListWrap}>
33          <TouchableOpacity style={styles.menuList} onPress={() => navigation.navigate('Brand')}>
34            <Text style={styles.menuText}>브랜드소개</Text>
35          </TouchableOpacity>
36          <TouchableOpacity style={styles.menuList} onPress={() => navigation.navigate('ProductList')}>
37            <Text style={styles.menuText}>제품</Text>
38          </TouchableOpacity>
39          <TouchableOpacity style={styles.menuList} onPress={() => navigation.navigate('NoticeNavi')}>
40            <Text style={styles.menuText}>공지사항</Text>
41          </TouchableOpacity>
42          <TouchableOpacity style={styles.menuList} onPress={() => navigation.navigate('QnaNavi', { screen: 'Qnalist' })}>
43            <Text style={styles.menuText}>문의답변</Text>
44          </TouchableOpacity>
45          <TouchableOpacity style={styles.menuList} onPress={() => navigation.navigate('ReviewNavi', { screen: 'ReviewList' })}>
46            <Text style={styles.menuText}>후기</Text>
47          </TouchableOpacity>
48        </View>
49
50        <TouchableOpacity style={styles.menuCloseBtn2} onPress={()=>setModalVisible(false)}>
51          <Text style={styles.menuCloseBtn2Text}>닫기</Text>
52        </TouchableOpacity>
53
54      /* sns 웹뷰 */
55      <SNSWeb/>
56    </View>
57  </Modal>
58 </View>
59 )
60 }

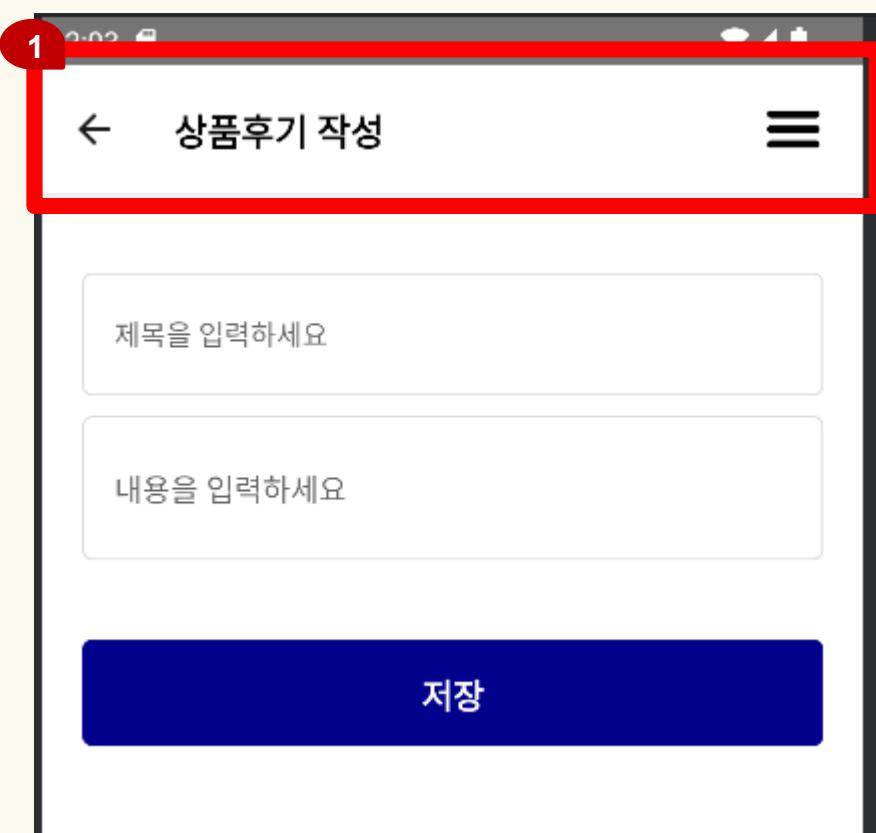
```

1



1

```
1 <Modal visible={modalVisible}
2   onRequestClose={()=>console.log('모달을 닫습니다.')}
3   onShow={()=>console.log('모달이 열립니다.')}
4   >
5     <View style={styles.modalWrap}>
6       <TouchableOpacity style={styles.menuCloseBtn1} onPress={()=>setModalVisible(false)}>
7         <Text style={styles.menuCloseBtn1Text}>X</Text>
8       </TouchableOpacity>
9
10    <View style={styles.menuListWrap}>
11      <TouchableOpacity style={styles.menuList} onPress={() => navigation.navigate('Brand')}>
12        <Text style={styles.menuText}>브랜드소개</Text>
13      </TouchableOpacity>
14      <TouchableOpacity style={styles.menuList} onPress={() => navigation.navigate('ProductList')}>
15        <Text style={styles.menuText}>제품</Text>
16      </TouchableOpacity>
17      <TouchableOpacity style={styles.menuList} onPress={() => navigation.navigate('NoticeNavi')}>
18        <Text style={styles.menuText}>공지사항</Text>
19      </TouchableOpacity>
20      <TouchableOpacity style={styles.menuList} onPress={() => navigation.navigate('QnaNavi', { screen: 'QnaList' })}>
21        <Text style={styles.menuText}>문의답변</Text>
22      </TouchableOpacity>
23      <TouchableOpacity style={styles.menuList} onPress={() => navigation.navigate('ReviewNavi', { screen: 'ReviewList' })}>
24        <Text style={styles.menuText}>후기</Text>
25      </TouchableOpacity>
26    </View>
27
28    <TouchableOpacity style={styles.menuCloseBtn2} onPress={()=>setModalVisible(false)}>
29      <Text style={styles.menuCloseBtn2Text}>닫기</Text>
30    </TouchableOpacity>
31
32    {/* sns 웹뷰 */}
33    <SNSWeb/>
34  </View>
35 </Modal>
```



1 const { width } = Dimensions.get('window');
2
3 const styles = StyleSheet.create({
4 header: {
5 padding: 10,
6 width: width,
7 height: 65,
8 backgroundColor: '#fcf9ee',
9 flexDirection: 'row',
10 justifyContent: 'space-between',
11 alignItems: 'center',
12 borderBottomWidth: 1,
13 borderBottomColor: '#ccc',
14 },
15})

1



1

```

1 const ColognesHomeTop = () => {
2   return (
3     <View style={styles.homeTop}>
4       <View>
5         <Text style={styles.title}>코롱</Text>
6         <View style={styles.content}>
7           <Text style={styles.text}>
8             조 말론 런던의 심플하면서도 우아한 코롱은 엄선된 원료를 바탕으로 서로 다른 코롱과 함께 페어링 할 수 있도록 디자인되었습니다.
9           </Text>
10          <Text style={styles.text}>나만을 위한, 가장 잘 어울리는 향을 완성해 보세요.</Text>
11        </View>
12        <Image source={{uri:"https://www.jomalone.co.kr/media/export/cms/campaigns/WOJML_FY24/MPP_Banners/2_Cologne/2408_WOJML_2_Cologne_MPP_Banner_Mobile.jpg"}} style={styles.image} resizeMode="cover"/>
13      </View>
14    );
15  );
16 };

```

2

```

1 const ColognesTotal = () => {
2   return (
3     <View style={styles.prodTotal}>
4       <View style={styles.content}>
5         <Text style={styles.text}>우리의 가장 사랑받는 코롱을 만나 보세요.</Text>
6         <Text style={styles.text}>베스트셀러부터 신제품까지 다양한 조 말론 런던의 컬렉션이 있습니다.</Text>
7       </View>
8     </View>
9   );
10 };
11

```



1

```
1 const ProductBlock = ({ navigation }) => {
2   return (
3     <View style={styles.blockBox}>
4       {products.map(product => (
5         <View key={product.id} style={styles.block}>
6           /* <View style={styles.wishIcon}></View> */
7           <Image source={{ uri: product.imageUrl }} style={styles.productImage} resizeMode="contain"/>
8           <View style={styles.textWrap}>
9             <Text style={styles.productText}>{product.name}</Text>
10            <Text style={styles.productText}>{product.category}</Text>
11            <Text style={styles.productText}>{product.koreanName}</Text>
12            <View style={styles.blockPrice}>
13              <Text style={styles.productText}>{product.price}</Text>
14            </View>
15            <TouchableOpacity
16              onPress={() => navigation.navigate('ProductDetail')}
17            >
18              <Text style={styles.addToCartText}>자세히 보기</Text>
19            </TouchableOpacity>
20          </View>
21        </View>
22      ))}
23    </View>
24  );
25};
26};
```

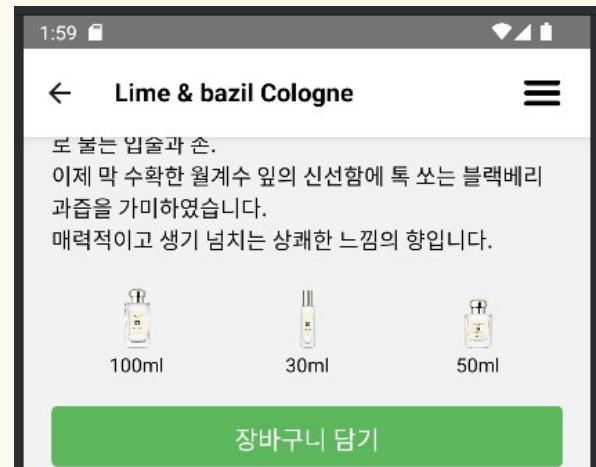


1

```

1  <View style={styles.imageBox}>
2    <Image source={image} style={styles.productImage} />
3  </View>
4
5  /* Product Details */
6  <View style={styles.asideBox}>
7    <Text style={styles.bestSeller}>베스트 셀러</Text>
8    <Text style={styles.productName}>Blackberry & Bay Cologne</Text>
9    <Text style={styles.productName}>블랙베리 앤 베이 코롱</Text>
10   <Text style={styles.reviews}>★★★★★ 13리뷰</Text>
11   <Text style={styles.price}>{price}</Text>
12   <Text style={styles.description}>
13     순수의 향. 블랙베리를 따던 어린 시절의 추억, 블랙베리로 물든 입술과 손. {'\n'}
14     이제 막 수확한 월계수 잎의 신선함에 톡 쏘는 블랙베리 과즙을 가미하였습니다. {'\n'}
15     매력적이고 생기 넘치는 상쾌한 느낌의 향입니다.
16   </Text>
17
18  /* Size Selection */
19  <View style={styles.sizeBox}>
20    <TouchableOpacity onPress={() => handleSizeChange('100ml')} style={styles.sizeButton}>
21      <Image source={{uri:'https://www.jomalone.co.kr/media/export/cms/products/1000x1000/jo_sku_L32R01_1000x1000_0.png'}} style={styles.sizeImage} />
22      <Text>100ml</Text>
23    </TouchableOpacity>
24
25    <TouchableOpacity onPress={() => handleSizeChange('30ml')} style={styles.sizeButton}>
26      <Image source={{uri:'https://www.jomalone.co.kr/media/export/cms/products/1000x1000/jo_sku_L32P01_1000x1000_0.png'}} style={styles.sizeImage} />
27      <Text>30ml</Text>
28    </TouchableOpacity>
29
30    <TouchableOpacity onPress={() => handleSizeChange('50ml')} style={styles.sizeButton}>
31      <Image source={{uri:'https://www.jomalone.co.kr/media/export/cms/products/1000x1000/jo_sku_LAKE01_1000x1000_0.png'}} style={styles.sizeImage} />
32      <Text>50ml</Text>
33    </TouchableOpacity>
34  </View>
35
36  /* Add to Basket */
37  <TouchableOpacity style={styles.addButton} onPress={handleAddToCart}>
38    <Text style={styles.addButtonText}>장바구니 담기</Text>
39  </TouchableOpacity>
40
41  </View>

```



1

성분

전성분: 변성알코올, 정제수, 향료, 유제놀, 하이드록시이소헥실 3-사이클로헥센카복스알데하이드, ...

테이스팅 노트

탑 노트: 블랙베리



톡하고 과즙을 내뿜는 천연 블랙커런트 꽃봉 오리가 과일 향과 새콤한 느낌을 더해줍니다.

하트 노트: 월계수 잎



갈바님의 천연향을 바탕으로 한 어코드가 향에 식물의 선명함과 눈부신 녹음의 느낌을 더해줍니다.

베이스 노트: 시더우드



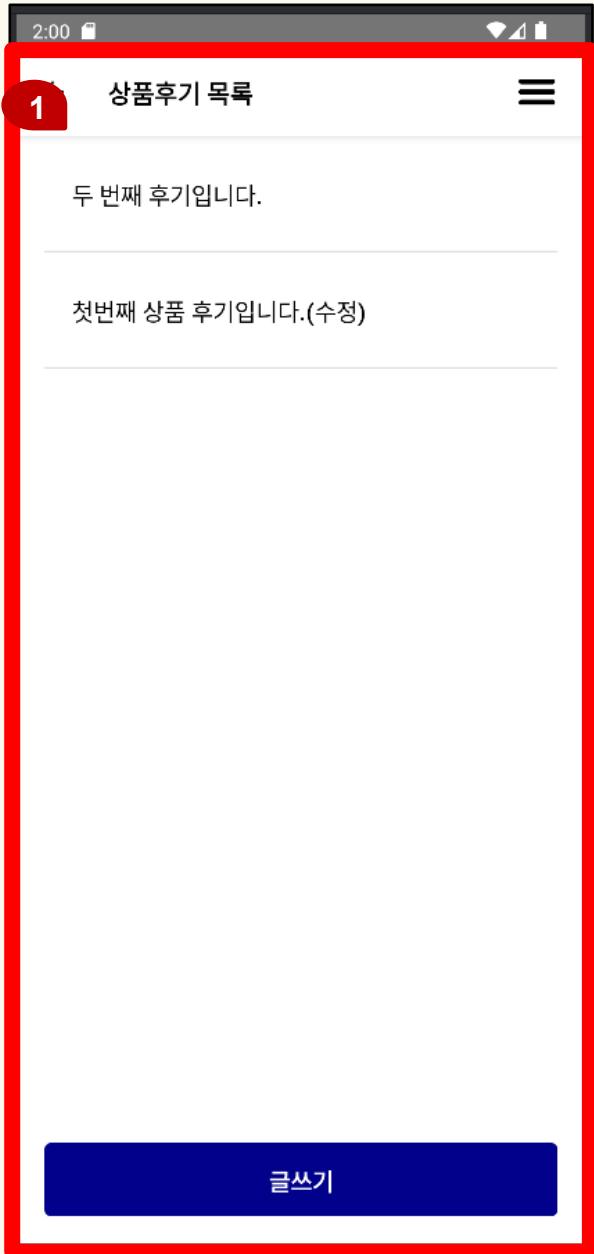
우디하고 드라이한 느낌이 하트 노트에 카리스마 있는 강렬함을 채워줍니다.

1

```

1  <View style={styles.tastingNotes}>
2      <Text style={styles.tastingNotesTitle}>테이스팅 노트</Text>
3
4      /* Top Note */
5      <View style={styles.tastingNote}>
6          <Image source={{uri: 'https://www.jomalone.co.kr/media/export/cms/spp/tasing_notes/mobile/Blackberry_Mobile.png'}} style={styles.tastingNoteImage} />
7          <View style={styles.tastingNoteText}>
8              <Text>탑 노트: 블랙베리</Text>
9              <Text>톡하고 과즙을 내뿜는 천연 블랙커런트 꽃봉오리가 과일 향과 새콤한 느낌을 더해줍니다.</Text>
10             </View>
11         </View>
12
13     /* Heart Note */
14     <View style={styles.tastingNote}>
15         <Image source={{uri: 'https://www.jomalone.co.kr/media/export/cms/spp/tasing_notes/mobile/Bay_Leaves_Mobile.png'}} style={styles.tastingNoteImage} />
16         <View style={styles.tastingNoteText}>
17             <Text>하트 노트: 월계수 잎</Text>
18             <Text>갈바님의 천연향을 바탕으로 한 어코드가 향에 식물의 선명함과 눈부신 녹음의 느낌을 더해줍니다.</Text>
19             </View>
20         </View>
21
22     /* Base Note */
23     <View style={styles.tastingNote}>
24         <Image source={{uri: 'https://www.jomalone.co.kr/media/export/cms/spp/tasing_notes/mobile/Cedarwood_Mobile.png'}} style={styles.tastingNoteImage} />
25         <View style={styles.tastingNoteText}>
26             <Text>베이스 노트: 시더우드</Text>
27             <Text>우디하고 드라이한 느낌이 하트 노트에 카리스마 있는 강렬함을 채워줍니다.</Text>
28             </View>
29         </View>
30     </View>

```



```

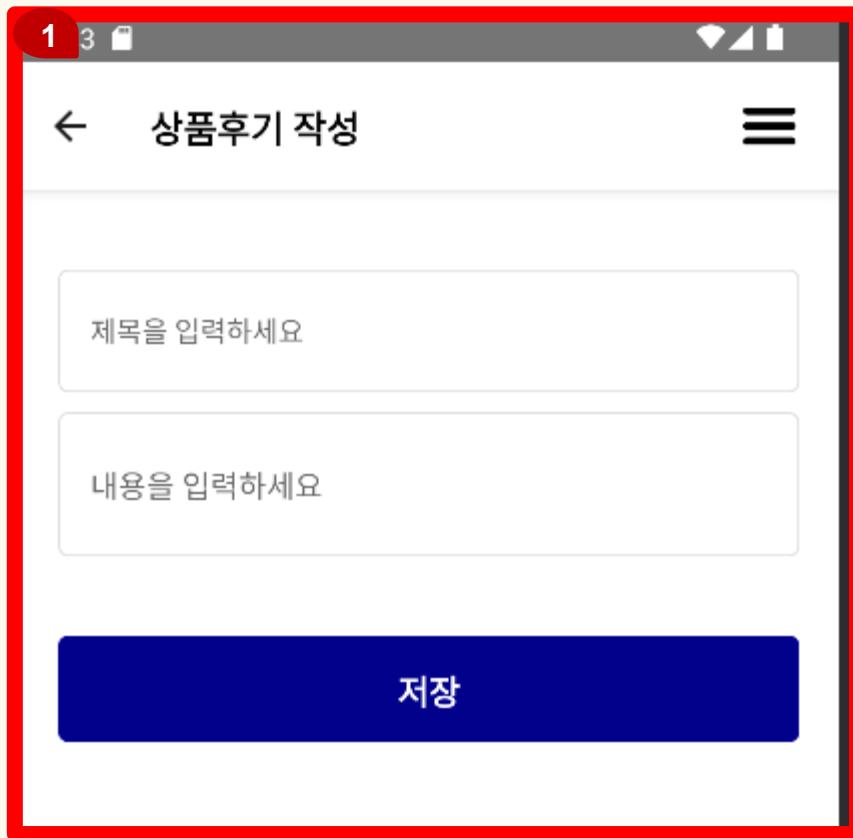
1  const ReviewList = ({navigation}) => {
2
3      const [nowId, setNowId] = useState(4)
4
5      const [datas, setDatas] = useState([])
6
7      useEffect(()=>{
8          const reviewRef = database().ref('review')
9
10         reviewRef.on('value',snapshot=>{
11             const data = snapshot.val()
12             const arr = []
13             for (const key in data){
14                 arr.push({id : key, ...data[key]})
15             }
16
17             setDatas(arr)
18             console.log('데이터리스트:', arr)
19         })
20     },[])
21
22     const writeGo = () => {
23         navigation.navigate('ReviewWrite', {setDatas, nowId, setNowId});
24     }
25
26     const detailGo = (data) => {
27         navigation.navigate('ReviewDetail', {data, deleteReg})
28     }
29
30     const deleteReg = (delId) => {
31         setDatas((prevDatas) => prevDatas.filter((item)
32             )
33
34         const listModifyReg = (newItem) => {
35             setDatas((prevDatas) => prevDatas.map((item) =>
36             )

```

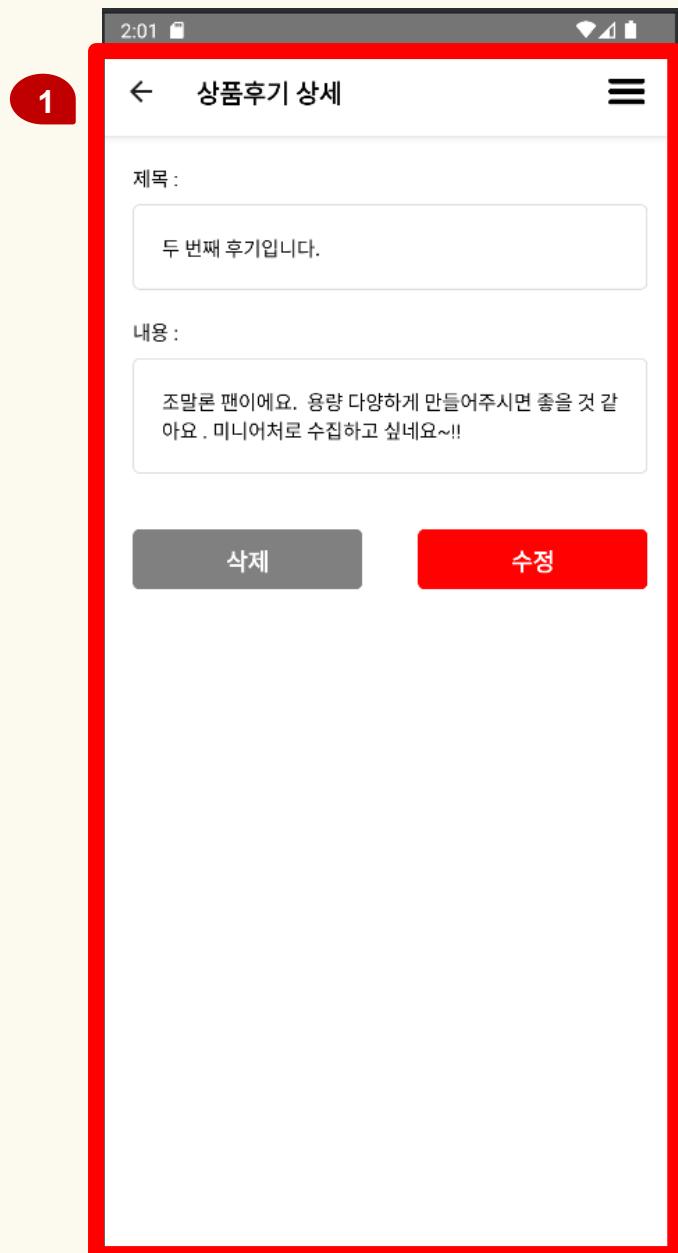
```

1  return (
2      <SafeAreaView style={styles.pageWrap}>
3          <FlatList style={styles.listWrap}
4              data={datas}
5              keyExtractor={(item) => item.id.toString()}
6              renderItem={({item}) => (
7                  <TouchableOpacity onPress={() => detailGo(item)} style={styles.list}>
8                      <Text style={styles.listTitle}>{item.title}</Text>
9                  </TouchableOpacity>
10             )
11         }
12     )
13
14
15     <View style={styles.btnWrap}>
16         <TouchableOpacity style={styles.btn} onPress={writeGo}>
17             <Text style={styles.btnText}>글쓰기</Text>
18         </TouchableOpacity>
19     </View>
20  </SafeAreaView>
21 )

```



```
1  const [title,setTitle] = useState('')
2  const [content,setContent] = useState('')
3  const [author,setAuthor] = useState('')
4  const [date, setDate] = useState(new Date().toISOString())
5
6  const saveGo = ()=> {
7    const newItem = {
8      id:nowId, title:title, content:content, author:author
9    }
10
11  setNowId(nowId + 1);
12
13  setDatas((prevDatas)=>[...prevDatas, newItem]);
14
15
16  const reviewRef = database().ref('review').push()
17
18  reviewRef.set({
19    title : title,
20    content : content,
21    regdate : date,
22    author: author
23  })
24
25  navigation.goBack() // 화면 뒤로 가기
26
27
28  return (
29    <SafeAreaView style={styles.pageWrap}>
30      <View style={styles.listWrap}>
31        {/* <Text>글쓰기입니다.</Text> */}
32        <TextInput value={title} onChangeText={setTitle} placeholder='제목을 입력하세요' style={styles.titleWrite}/>
33        <TextInput value={content} onChangeText={setContent} placeholder='내용을 입력하세요' multiline={true} numberOfLines={6} style={styles.contentWrite}/>
34
35
36      <View style={styles.btnWrap}>
37        <TouchableOpacity style={styles.btn} onPress={saveGo}>
38          <Text style={styles.btnText}>저장</Text>
39        </TouchableOpacity>
40      </View>
41    </SafeAreaView>
42  )
}
```



```
1  const modifyGo = ( )=> {
2      navigation.navigate('ReviewModify',{data, modifyReg});
3  }
4
5  const modifyReg = (updateItem) => {
6      setDetailData(updateItem);
7      listModifyReg(updateItem)
8  }
9
10 return (
11     <SafeAreaView style={styles.pageWrap}>
12         <View style={styles.listWrap}>
13             {/* <Text>게시글 상세보기입니다.</Text> */}
14             <View style={styles.titleWrap}>
15                 <Text>제목 : </Text>
16                 <Text style={styles.title}>{detailData.title}</Text>
17             </View>
18             <View style={styles.contentWrap}>
19                 <Text>내용 : </Text>
20                 <Text style={styles.content}>{detailData.content}</Text>
21             </View>
22
23         </View>
24
25         <View style={styles.btnWrap}>
26             <TouchableOpacity style={styles.delBtn} onPress={delGo}>
27                 <Text style={styles.btnClose}>삭제</Text>
28             </TouchableOpacity>
29             <TouchableOpacity style={styles.modifyBtn} onPress={modifyGo}>
30                 <Text style={styles.btnClose}>수정</Text>
31             </TouchableOpacity>
32         </View>
33     </SafeAreaView>
34 )
35 
```

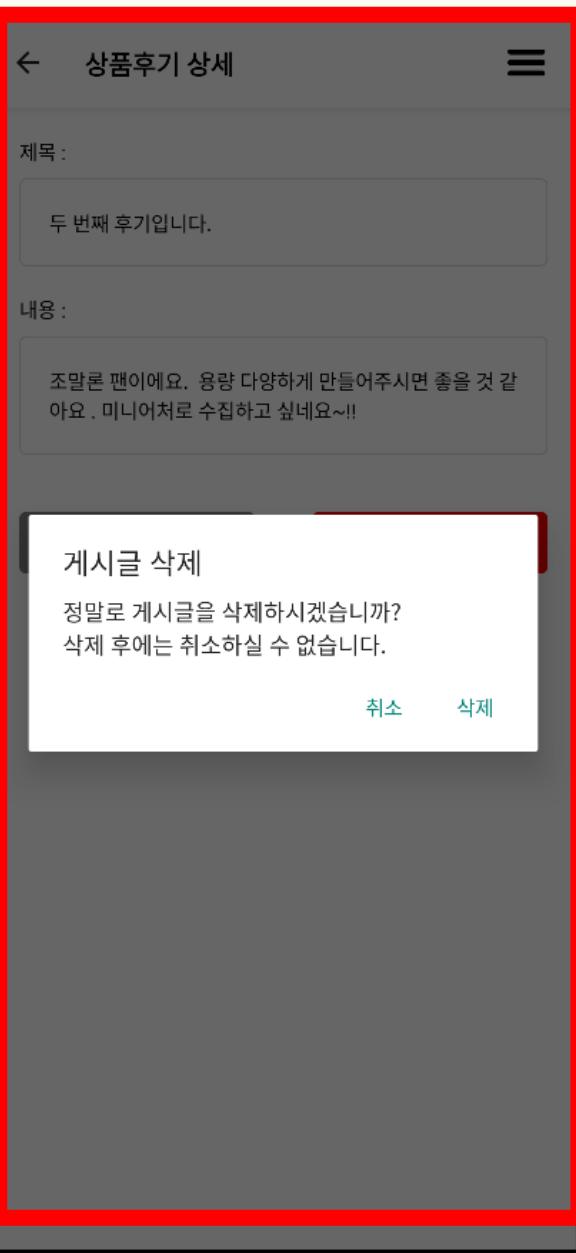
1



1

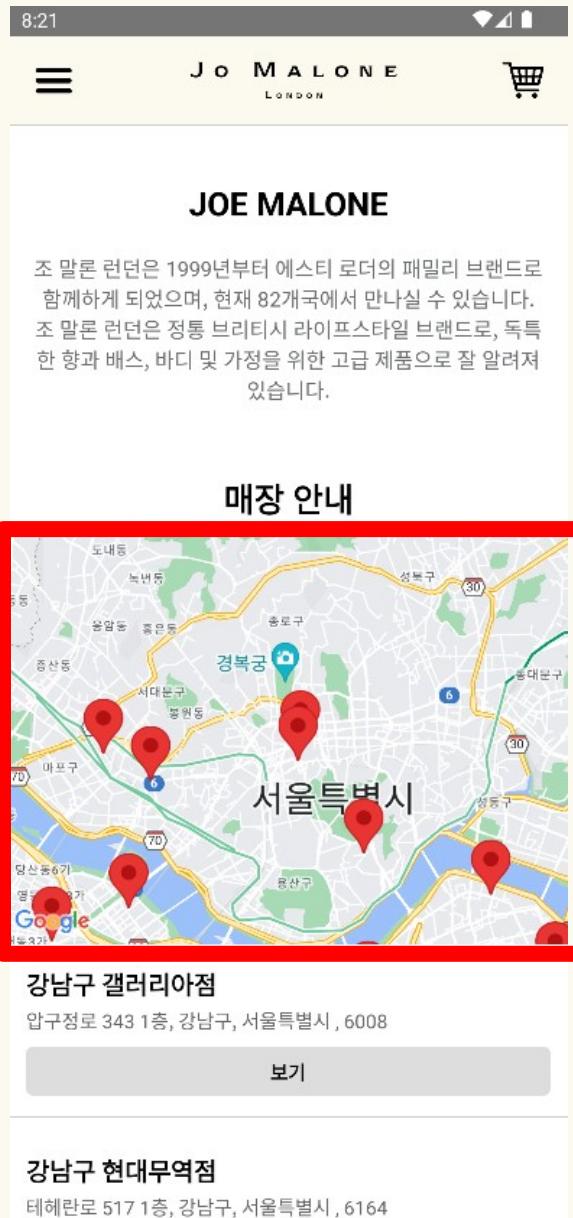
```
1  const {data, modifyReg} = route.params;
2
3  const [title,setTitle] = useState(data.title)
4  const [content,setContent] = useState(data.content)
5  const [author,setAuthor] = useState(data.author)
6
7  const modifyClk = ( ) => {
8      const reviewRef = database().ref(`/review/${data.id}`);
9      reviewRef.update({
10          title: title,
11          content: content,
12          update: new Date().toISOString(),
13          author: author,
14      });
15
16  const newItem = {id : data.id, title, content, author}
17  modifyReg(newItem)
18  navigation.goBack();
19 }
20
21 return (
22     <SafeAreaView style={styles.pageWrap}>
23         <View style={styles.listWrap}>
24             {/* <Text>게시글 수정입니다.</Text> */}
25             <TextInput value={title} onChangeText={setTitle} style={styles.titleWrite}/>
26             <TextInput value={content} onChangeText={setContent} style={styles.contentWrite} multiline={true} numberOfLines={4}/>
27         </View>
28
29         <View style={styles.btnWrap}>
30             /* <TouchableOpacity style={styles.delBtn}>
31                 <Text style={styles.btnClose}>취소</Text>
32             </TouchableOpacity> */
33             <TouchableOpacity style={styles.modifyBtn} >
34                 <Text style={styles.btnClose} onPress={modifyClk}>수정완료</Text>
35             </TouchableOpacity>
36         </View>
37     </SafeAreaView>
38 )
```

1



1

```
1  const delGo = () => {
2      Alert.alert('게시글 삭제', '정말로 게시글을 삭제하시겠습니까? \n삭제 후에는 취소하실 수 없습니다.', [
3          {
4              text:'취소',
5              style:'cancel'
6          },
7          {
8              text:'삭제',
9              style:'destructive',
10             onPress:()=>{
11                 const reviewRef = database().ref(`/review/${data.id}`)
12                 reviewRef.remove();
13
14                 deleteReg(data.id);
15                 navigation.goBack();
16             }
17         },
18     ],
19 )
20 }
```



1

```

1 import React, { useRef, useEffect } from 'react';
2 import { View, StyleSheet } from 'react-native';
3 import MapView, { Marker } from 'react-native-maps';
4
5 const GoogleMap = ({ storeData, selectedLocation }) => {
6   const mapRef = useRef(null);
7
8   // 선택된 위치로 이동
9   useEffect(() => {
10     if (selectedLocation && mapRef.current) {
11       mapRef.current.animateToRegion({
12         latitude: selectedLocation.lat,
13         longitude: selectedLocation.lng,
14         latitudeDelta: 0.1,
15         longitudeDelta: 0.1,
16       });
17     }
18   }, [selectedLocation]);
19
20   return (
21     <View style={styles.container}>
22       <MapView
23         ref={mapRef}
24         style={styles.map}
25         initialRegion={{
26           latitude: 37.5665, // 서울 중심
27           longitude: 126.9780,
28           latitudeDelta: 0.1,
29           longitudeDelta: 0.1,
30         }}
31       >
32         {storeData.map((store, index) => (
33           <Marker
34             key={index}
35             coordinate={{
36               latitude: store.latlng.lat, // store.latlng.lat 사용
37               longitude: store.latlng.lng, // store.latlng.lng 사용
38             }}
39             title={store.store}
40             description={store.address} // 주소를 마커의 설명에 추가
41           />)
42         )}
43       </MapView>
44     </View>
45   );
46 };

```



1

```

1 import React from 'react';
2 import { View, Text, TouchableOpacity, StyleSheet, ScrollView } from 'react-native';
3 import locationData from './locationData';
4
5 const StoreList = ({ onStoreSelect }) => {
6   return (
7     <ScrollView>
8       {locationData.map((item, index) => (
9         <View key={index} style={styles.storeItem}>
10           <Text style={styles.storeTitle}>{item.store}</Text>
11           <Text style={styles.address}>{item.address}</Text>
12           <TouchableOpacity
13             style={styles.guideButton}
14             onPress={() => onStoreSelect(item.latlng)}
15           >
16             <Text style={styles.guideText}>{item.guide}</Text>
17           </TouchableOpacity>
18         </View>
19       )));
20     </ScrollView>
21   );
22 };
23
24 const styles = StyleSheet.create({
25   storeItem: {
26     padding: 15,
27     borderBottomWidth: 1,
28     borderBottomColor: '#ccc',
29     marginBottom: 10,
30   },
31   storeTitle: {
32     fontSize: 18,
33     fontWeight: 'bold',
34   },
35   address: {
36     fontSize: 14,
37     color: '#666',
38     marginTop: 5,
39   },
40   guideButton: {
41     marginTop: 10,
42     padding: 8,
43     backgroundColor: '#ddd',
44     borderRadius: 5,
45     alignItems: 'center',
46   },
47   guideText: {
48     fontSize: 14,
49     fontWeight: 'bold',
50   },
51 });
52
53 export default StoreList;

```

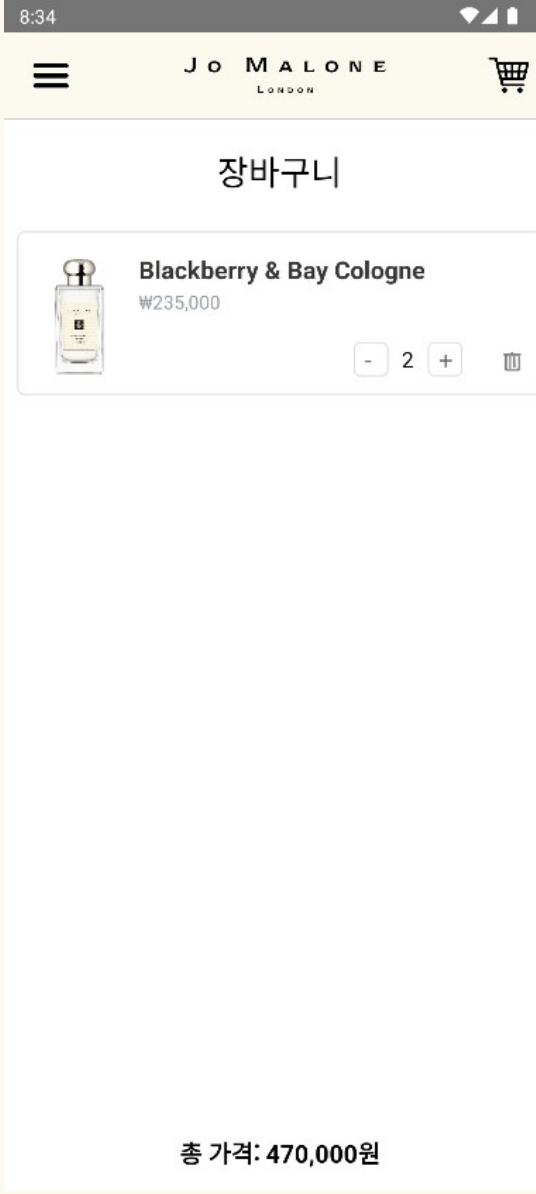


1

```

1 import React, {useEffect, useState} from 'react';
2 import {
3   View,
4   Text,
5   TouchableOpacity,
6   StyleSheet,
7   FlatList,
8   ActivityIndicator,
9 } from 'react-native';
10 import database from '@react-native-firebase/database';
11
12 function Notice({navigation}) {
13   const [notices, setNotices] = useState([]);
14   const [loading, setLoading] = useState(true);
15
16   useEffect(() => {
17     const noticesRef = database().ref('notices');
18
19     const unsubscribe = noticesRef.on('value', snapshot => {
20       const data = snapshot.val();
21       if (data) {
22         const noticesList = Object.keys(data).map(key => ({
23           id: key,
24           ...data[key],
25         }));
26         setNotices(noticesList);
27       } else {
28         setNotices([]);
29       }
30       setLoading(false);
31     });
32
33     return () => noticesRef.off('value', unsubscribe);
34   }, []);
35
36   return (
37     <View style={styles.container}>
38       <Text style={styles.title}>공지사항</Text>
39
40       {loading ? (
41         <ActivityIndicator size="large" color="#0000ff" />
42       ) : (
43         <FlatList
44           data={notices}
45           keyExtractor={(item) => item.id}
46           renderItem={({ item }) => (
47             // 리스트 전체 항목을 클릭할 수 있게 변경
48             <TouchableOpacity
49               onPress={() => navigation.navigate('NoticeDetail', { noticeId: item.id })}
50               style={styles.noticeItem} // TouchableOpacity로 감싸서 전체 항목 클릭 가능
51             >
52               <Text style={styles.noticeTitle}>{item.title}</Text>
53               <Text>{item.content}</Text>
54               <Text style={styles.date}>{new Date(item.createdAt).toLocaleString()}</Text>
55             </TouchableOpacity>
56           )
57         )
58       )
59
60       <TouchableOpacity
61         onPress={() => navigation.navigate('AddNotice')}
62         style={styles.button}
63         <Text>공지사항 추가</Text>
64       </TouchableOpacity>
65     </View>
66   );
67 }

```



```

1 import React, { useEffect, useState } from 'react';
2 import { FlatList, StyleSheet, Text, View } from 'react-native';
3 import MainHeader from '../MainHeader';
4 import { SafeAreaView } from 'react-native-safe-area-context';
5 import database from '@react-native-firebase/database';
6 import CartDetail from './CartDetail';
7
8 function Cart({ navigation }) {
9   const [cartItems, setCartItems] = useState([]);
10  const [quantities, setQuantities] = useState({});
11
12  useEffect(() => {
13    const ref = database().ref('/cart');
14    const listener = ref.on('value', snapshot => {
15      const items = [];
16      snapshot.forEach(child => {
17        items.push({
18          key: child.key,
19          ...child.val(),
20        });
21      });
22      setCartItems(items);
23    });
24
25    return () => ref.off('value', listener);
26  }, []);
27
28 // 장바구니 휴지통 기능
29 const handleRemove = (itemKey) => {
30  database().ref('/cart').child(itemKey).remove()
31  .then(() => {
32    console.log('Firebase에서 삭제');
33  })
34  .catch(error => {
35    console.error('삭제하기 에러:', error);
36  });
37}
38
39 // 장바구니 개수 변화
40 const handleQuantityChange = (itemKey, quantity) => {
41  setQuantities(prev => ({...prev, [itemKey]: quantity}));
42}
43
44 const getTotalPrice = () => {
45  return cartItems.reduce((total, item) => {
46    const itemPrice = parseFloat(item.price.replace('₩', '').replace(',', '')) || 0;
47    const itemQuantity = quantities[item.key] || 1;
48    return total + itemPrice * itemQuantity;
49  }, 0);
50}
51
52 const renderItem = ({ item }) => (
53  <CartDetail item={item} onQuantityChange={handleQuantityChange} onRemove={handleRemove} />
54)
55
56 return (
57  <SafeAreaView style={styles.safeContainer}>
58    <MainHeader navigation={navigation} />
59    <Text style={styles.mainText}>장바구니</Text>
60    <FlatList
61      data={cartItems}
62      renderItem={renderItem}
63      keyExtractor={item => item.key}
64      contentContainerStyle={styles.flatListStyle}
65    />
66    <Text style={styles.totalPriceText}>
67      총 가격: {getTotalPrice().toLocaleString()} 원
68    </Text>
69  </SafeAreaView>
70);
71 }

```

```

1
2
3
4 const CartDetail = ({ item, onQuantityChange, onRemove }) => {
5  const [quantity, setQuantity] = useState(1);
6
7  const handleIncrease = () => {
8    setQuantity(prev => {
9      const newQuantity = prev + 1;
10     onQuantityChange(item.key, newQuantity);
11     return newQuantity;
12   });
13 }
14
15 const handleDecrease = () => {
16  setQuantity(prev => {
17    if (prev > 1) {
18      const newQuantity = prev - 1;
19      onQuantityChange(item.key, newQuantity);
20      return newQuantity;
21    }
22    return prev;
23  });
24 }
25
26 return (
27  <View style={styles.item}>
28    <View style={styles.row}>
29      {/* 이미지 공간 */}
30      <View style={styles.imageContainer}>
31        <Image
32          source={{ uri: 'https://www.jomalone.co.kr/media/export/cms/products/1000x1000/jo_sku_L32R01_1000x1000_0.png' }}
33          style={styles.productImage}
34        />
35      </View>
36      {/* 텍스트와 비트 공간 */}
37      <View style={styles.textContainer}>
38        <Text style={styles.productName}>{item.productName}</Text>
39        <Text style={styles.price}>{item.price}</Text>
40        <View style={styles.quantityContainer}>
41          <TouchableOpacity onPress={handleDecrease} style={styles.quantityButton}>
42            <Text style={styles.quantityText}>-</Text>
43          </TouchableOpacity>
44          <Text style={styles.quantityNumber}>{quantity}</Text>
45          <TouchableOpacity onPress={handleIncrease} style={styles.quantityButton}>
46            <Text style={styles.quantityText}>+</Text>
47          </TouchableOpacity>
48          <TouchableOpacity onPress={() => onRemove(item.key)} >
49            <Image source={require('../img/common/trash.png')} style={styles.trashLogo} />
50          </TouchableOpacity>
51        </View>
52      </View>
53    </View>
54  );
55};
56

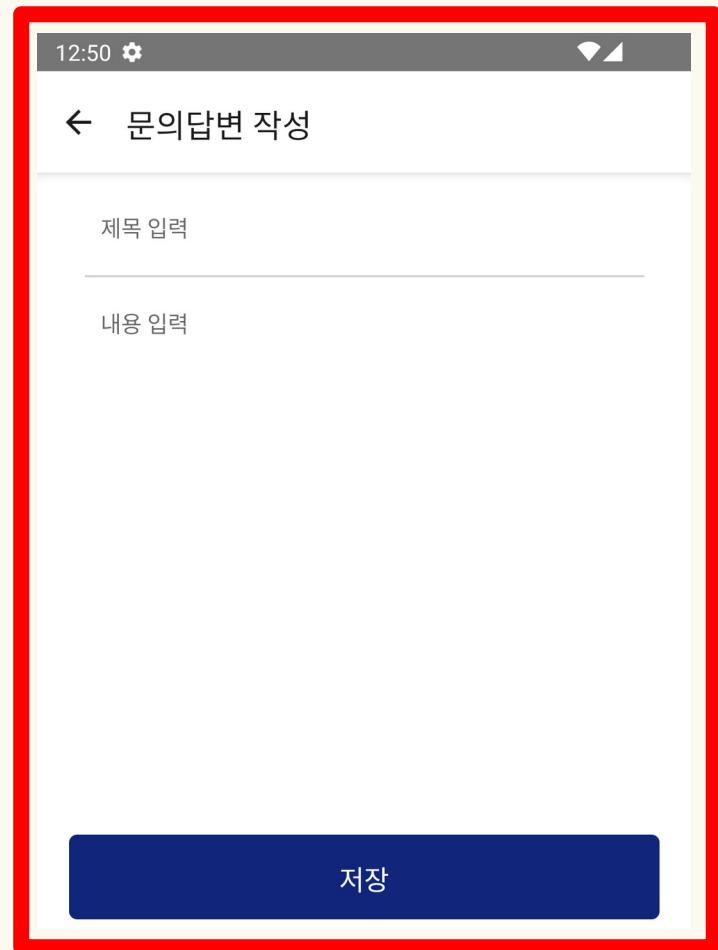
```



1

```
1  const [datas,setData] = useState([])
2
3  useEffect(() => {
4      const qnaRef = database().ref('qna'); // qna : db 컬렉션 이름
5      qnaRef.on('value', snapshot => {
6          const data = snapshot.val();
7          const arr = [];
8          for(const key in data){
9              arr.push({ id: key, ...data[key] }); //Firebase 데이터 상태에 저장
10         }
11         setData(arr); // 데이터를 상태에 저장
12     });
13
14     //컴포넌트 언마운트 시 Firebase 해제
15     return () => qnaRef.off('value');
16 }, []);
17
18 const writeGo = () => {
19     // Alert.alert("클릭")
20     navigation.navigate('QnaWrite', {setDatas});
21 }
22 // console.log('navigation02 : ', navigation)
23 const deleteReg = (delId)=>{
24     setData((prevDatas)=> prevDatas.filter((item)=> item.id != delId))
25 }
26 const listModifyReg = (newItem)=>{
27     setData(
28         (prevDatas)=> prevDatas.map((item)=> item.id === newItem.id ? newItem : item )
29     )
30 }
31 const detailGo = (data) => {
32     navigation.navigate('QnaDetail', {
33         data, deleteReg, listModifyReg
34     });
35 }
```

1

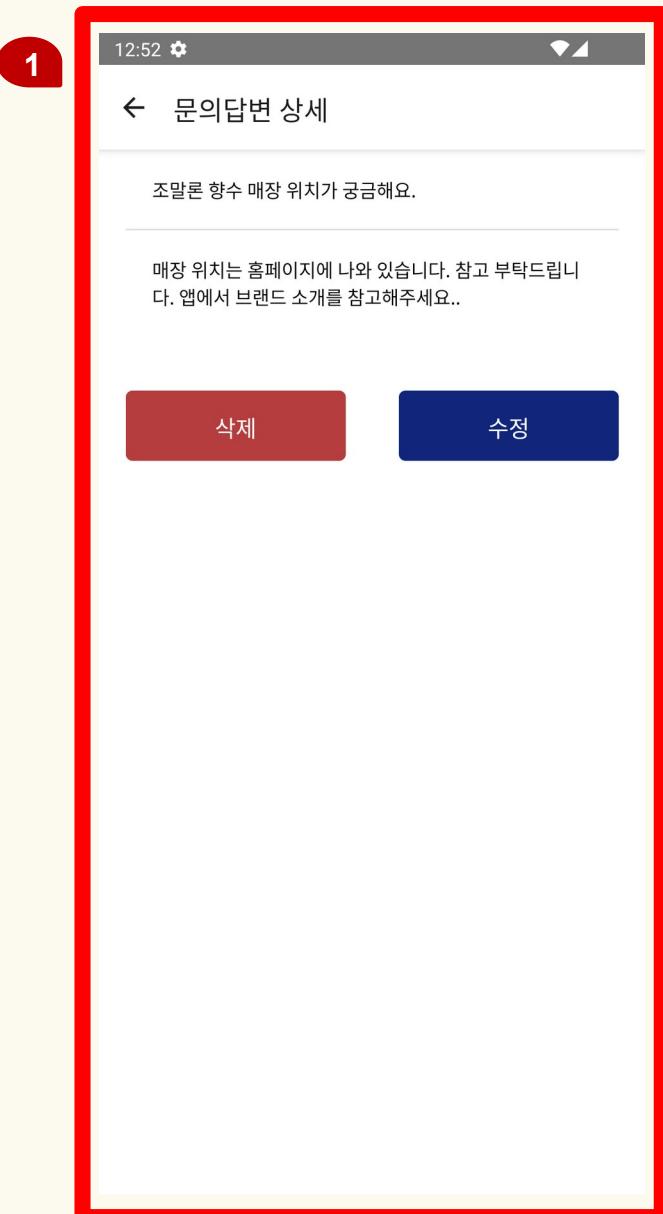


1

```

1  const {setDatas, nowId, setNowId} = route.params
2
3  const [title, setTitle] = useState('')
4  const [content, setContent] = useState('')
5  const [author, setAuthor] = useState('')
6  const [date, setDate] = useState(new Date().toISOString())
7
8  const saveGo = () =>{
9      if (!title && !content) {
10          Alert.alert("제목과 내용을 입력하세요.");
11          return;
12      }else if (!title && content){
13          Alert.alert("제목을 입력하세요.");
14          return;
15      }else if (title && !content){
16          Alert.alert("내용을 입력하세요.");
17          return;
18      }
19
20      const newItem ={
21          id:nowId, title:title, content:content, author:author
22      }
23
24      setNowId(nowId+1)
25      setDatas((prevDatas)=> [...prevDatas,newItem])
26      const qnaRef = database().ref('qna').push()
27      qnaRef.set({
28          title : title,
29          content : content,
30          regdate : date,
31          author: author
32      })
33
34      navigation.goBack() // 화면 뒤로

```



1

```
1  const {data, deleteReg, listModifyReg} = route.params;
2
3  const [detailData, setDetailData] = useState(data);
4
5  const delGo = ()=>{
6      Alert.alert('글삭제','삭제하시겠습니까?', [
7          {
8              text:'취소',
9              style:'cancel'
10         },
11
12         {
13             text:'삭제',
14             style:'destructive',
15             onPress:()=>{
16                 const qnaRef = database().ref(`/qna/${data.id}`);
17                 qnaRef.remove();
18                 deleteReg(data.id)
19                 navigation.goBack()
20             }
21         }
22     ])
23 }
24
25
26  const modifyGo = ()=>{
27      navigation.navigate('QnaModify',{data, modifyReg})
28 }
29
30  const modifyReg = (updateItem)=>{
31      setDetailData(updateItem)
32      listModifyReg(updateItem)
33 }
```

1

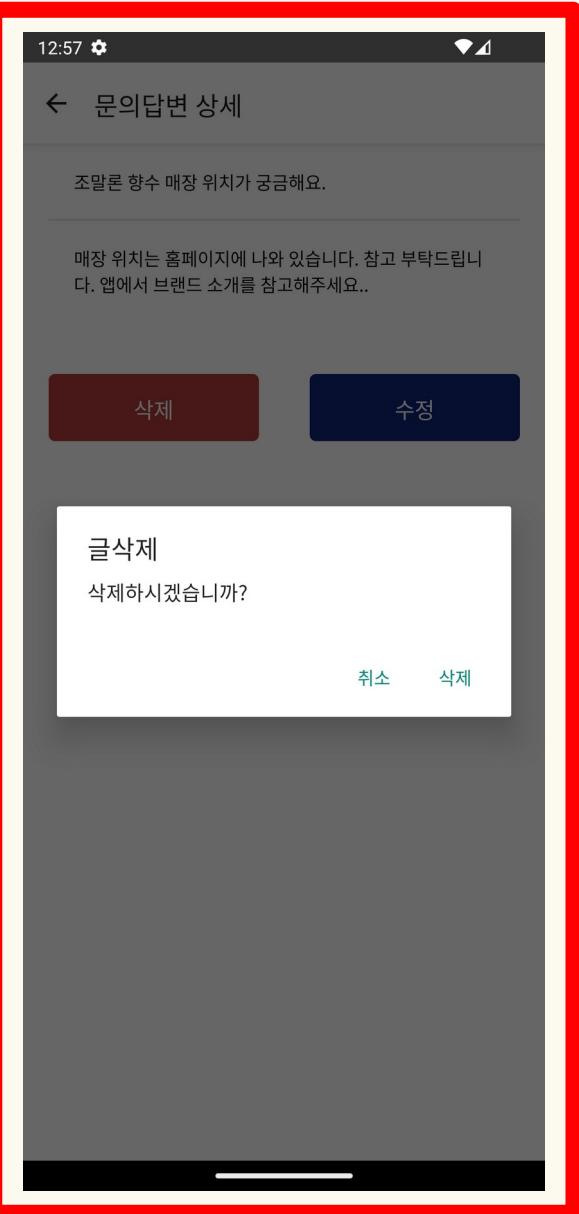


1

```

1  const {data,modifyReg} = route.params;
2
3  const [title, setTitle] = useState(data.title)
4  const [content, setContent] = useState(data.content)
5  const [author, setAuthor] = useState(data.author)
6
7  const modifyClk = ()=>{
8      if (!title && !content) {
9          Alert.alert("제목과 내용을 입력하세요.");
10         return;
11     }else if (!title && content){
12         Alert.alert("제목을 입력하세요.");
13         return;
14     }else if (title && !content){
15         Alert.alert("내용을 입력하세요.");
16         return;
17     }
18
19     const qnaRef = database().ref(`/qna/${data.id}`);
20     qnaRef.update({
21         title: title,
22         content: content,
23         update: new Date().toISOString(),
24         author: author,
25     });
26     const newItem = {
27         id : data.id, title, content, author
28     }
29
30     modifyReg(newItem)
31     navigation.goBack()
32 }
```

1



1

```
1 const delGo = ()=>{
2   Alert.alert('글삭제','삭제하시겠습니까?', [
3     {
4       text:'취소',
5       style:'cancel'
6     },
7     {
8       text:'삭제',
9       style:'destructive',
10      onPress:()=>{
11        const qnaRef = database().ref(`/qna/${data.id}`);
12        qnaRef.remove();
13        deleteReg(data.id)
14        navigation.goBack()
15      }
16    }
17  ])
18}
19])
20}
```

```
1 <TouchableOpacity style={[styles.button, styles.delButton]} onPress={delGo}>
2   <Text style={styles.btnTxt}>삭제</Text>
3 </TouchableOpacity>
```

X

브랜드소개

제품

공지사항

문의답변

후기

닫기

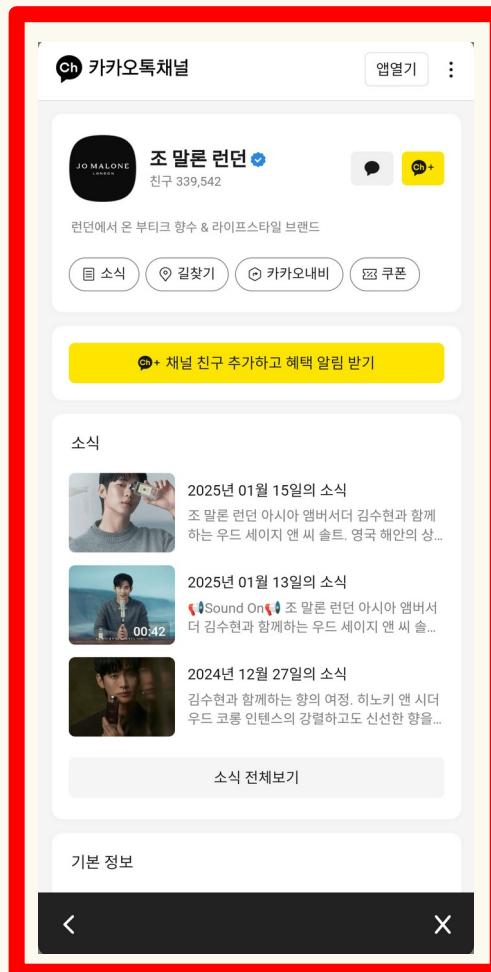
1



```
1 import WebView from 'react-native-webview'  
2  
3 // 폰트 어썸 import  
4 import { FontAwesomeIcon } from '@fortawesome/react-native-fontawesome'  
5 import { faInstagram, faYoutube } from '@fortawesome/free-brands-svg-icons'  
6 import { faChevronLeft, faX, faComment } from '@fortawesome/free-solid-svg-icons'
```

```
1 <View style={styles.snsWrap}>  
2   <TouchableOpacity onPress={openInstaApp}>  
3     <FontAwesomeIcon icon={faInstagram} size={40} />  
4   </TouchableOpacity>  
5   <TouchableOpacity onPress={openKakaoApp}>  
6     /* 페북 링크가 자꾸 변경돼서 열림 > 카카오채널로 변경 */  
7     /* <FontAwesomeIcon icon={faSquareFacebook} size={40} /> */  
8     <FontAwesomeIcon icon={faComment} size={40}/>  
9   </TouchableOpacity>  
10  <TouchableOpacity onPress={openYoutubeApp}>  
11    <FontAwesomeIcon icon={faYoutube} size={40} />  
12  </TouchableOpacity>  
13 </View>
```

2



```

1 <Modal visible={modalOpen} transparent={true}
2   animationType="slide"
3   onRequestClose={() => setModalOpen(false)} // 뒤로가기 시 모달 닫기
4 >
5   <View style={styles.modalView}>
6     <View style={styles.modalContent}>
7
8       <WebView ref={SNSWebView}
9         source={{ uri: url }} style={{ flex: 1 }}
10      />
11
12     <View style={styles.btnWrap}>
13
14       /* 웹뷰 뒤로가기 */
15       <TouchableOpacity style={styles.webBtn} onPress={goBack}>
16         <FontAwesomeIcon icon={faChevronLeft}
17           style={styles.btnClose} size={20}/>
18       </TouchableOpacity>
19
20       /* 모달 닫기 버튼 */
21       <TouchableOpacity style={styles.webBtn}
22         onPress={() => setModalOpen(false)}
23       >
24         <FontAwesomeIcon icon={faX}
25           style={styles.btnClose} size={20}
26         />
27       </TouchableOpacity>
28
29     </View>
30
31   </View>
32 </View>
33 </Modal>
```

```

1 const [modalOpen, setModalOpen] = useState(false);
2 const [url, setUrl] = useState('');
3 const SNSWebView = useRef(null)
4
5 // 카카오웹뷰처리
6 const openKakaoApp = async () => {
7   // const snsurl = 'https://www.facebook.com/JoMaloneLondon/'
8   const snsurl = 'https://pf.kakao.com/_sxoxyj'
9   // console.log('openFacebookApp URL: ', snsurl);
10  setUrl(snsurl);
11  setModalOpen(true)
12 }
13
14 // 인스타웹뷰처리
15 const openInstaApp = ()=>{
16   const snsurl = 'https://www.instagram.com/jomalonelondon/'
17   // console.log('openInstaApp URL: ', snsurl);
18   setUrl(snsurl);
19   setModalOpen(true)
20 }
21
22 // 유튜브웹뷰처리
23 const openYoutubeApp =()=>{
24   const snsurl = 'https://www.youtube.com/user/JoMaloneLondon2012'
25   // console.log('openYoutubeApp URL: ', snsurl);
26   setUrl(snsurl);
27   setModalOpen(true)
28 }
29
30 const goBack = () => {
31   if (SNSWebView.current && SNSWebView.current.canGoBack) {
32     SNSWebView.current.goBack(); // 웹뷰에서 뒤로가기
33   } else {
34     setModalOpen(false); // 뒤로갈 페이지가 없다면 모달 닫기
35   }
36};
```

THANK
YOU