

# 실습 1단계

## 프로젝트 요약

로그인(구글 등) 후, 왼쪽(Editor)에서 명함 카드 데이터를 추가/수정/삭제하면 오른쪽(Preview)에서 즉시 미리보기로 렌더링되는 CRUD 실습 프로젝트입니다.

```
Marker.jsx  
CardEditor.jsx  
Preview.jsx
```

## 화면 구성과 컴포넌트 트리(전체 구조)

가장 상위 화면은 Marker이고, 내부에 CardEditor(편집) + Preview(미리보기) 두 영역이 나란히 배치됩니다.

### 컴포넌트 역할 요약

Marker: 카드 상태(cards)를 “진짜 주인”으로 들고 있는 부모(상태 관리 + 함수 전달)

CardEditor: cards 목록을 돌면서 CardEditorForm을 카드 개수만큼 렌더링 + 맨 아래에 CardAddForm(추가 폼)

CardEditorForm: 카드 1개를 수정/삭제하는 폼(테마 변경, 파일 변경, Delete 버튼)

CardAddForm: 새 카드 추가 폼(테마 선택 + 파일 선택 + Add 버튼)

Preview: cards를 돌면서 Card 컴포넌트로 명함을 출력(“보여주기 전용”)

Login: 구글 로그인 버튼 클릭 → 로그인 성공 시 /marker로 이동

데이터(상태) 흐름: “부모가 상태를 갖고, 자식은 요청한다”

이 프로젝트의 핵심은 상태(cards)는 Marker 한 곳에만 두고, 자식들은

“수정해줘/추가해줘/삭제해줘”라고 함수로 요청한다는 점입니다.

### 3-1. cards 상태는 Marker에 있음

Marker에서 useState로 cards를 관리합니다. 초기 예시 카드 2개가 들어있죠.

### 3-2. 추가/수정은 같은 함수(createOrUpdateCard)로 처리

createOrUpdateCard(card)는 “id를 키로 해서 card를 저장”합니다.

즉 id가 이미 있으면 수정, 없으면 추가가 됩니다(= upsert). Marker.jsx

### 3-3. 삭제(deleteCard)

객체 복사 후 delete updated[card.id]로 제거합니다.

Marker.jsx

### 3-4. 자식에게 props로 전달

CardEditor에 addCard/updateCard/deleteCard를 내려주고, Editor 내부에서 다시 Form들에게 전달합니다.

Marker.jsx

CardEditor.jsx

CardEditorForm.jsx  
 CardAddForm.jsx  
 Preview.jsx

#### 4. 파일 업로드 흐름: FileInput 컴포넌트를 “주입” 받는 방식

코드에서 FileInput은 직접 import하지 않고, props로 받아서 사용합니다.

CardAddForm({ FileInput, onAdd })에서 <FileInput ... onChange={...} /> 사용

CardEditorForm({ FileInput, card, updateCard })에서도 <FileInput ... onChange={...} /> 사용

이 구조의 장점(수업 포인트):

“파일 업로드 방식”을 나중에 Cloudinary / Firebase Storage / 로컬 업로드 등으로 바꿔도,

Form 코드는 그대로 두고 FileInput만 교체 가능

즉, 컴포넌트 재사용 + 의존성 분리(주입) 개념을 맛볼 수 있음

#### 5. CardAddForm 분석(추가 품): useRef + useState 조합

##### 5-1. 왜 useRef를 쓰나?

formRef: Add 후 폼을 통째로 reset()하기 위해 DOM 참조

themeRef: select 값을 state로 만들지 않고, 제출 시점에만 읽기 위해 DOM 참조

우리가 선택할 수 있는 방식에는 “\*\*입력값을 매번 state로 관리하는 방식(Controlled)\*\*도 있고,

\*\*필요할 때만 DOM에서 읽는 방식(useRef)\*\*도 있다”

##### 5-2. file은 useState로 관리

파일 선택 결과를 { fileName, fileURL } 형태로 저장합니다.

##### 5-3. Add 버튼 클릭 → onSubmit

event.preventDefault()로 새로고침 방지

CardAddForm.jsx

새 card 객체 생성: id: Date.now() (간단한 유니크 id)

CardAddForm.jsx

폼 reset + file state 초기화 + onAdd(card) 호출

CardAddForm.jsx

##### 실습 포인트

버튼 기본 타입이 submit인 경우가 많아 새로고침이 날 수 있음 → preventDefault()가 왜 필요한지

CardAddForm.jsx

“부모에게 추가 요청”은 onAdd(card) 한 줄로 끝난다(상태는 부모가 바꾸기 때문)

CardAddForm.jsx

#### 6. CardEditorForm 분석(수정/삭제 폼): 불변성 업데이트

CardEditorForm은 card 1개를 받아서 수정합니다.

CardEditorForm.jsx

## 6-1. 수정(update)의 핵심 문법: 스프레드 + 계산된 키

```
updateCard({ ...card, [event.currentTarget.name]: event.currentTarget.value, }); CardEditorForm.jsx
```

이 패턴은 “기존 card 복사 후, 특정 필드만 덮어쓰기”입니다.

CardEditorForm.jsx

## 6-2. 파일 변경도 같은 방식

파일을 바꾸면 fileName, fileURL만 덮어써서 updateCard 호출합니다.

CardEditorForm.jsx

## 6-3. 삭제는 deleteCard(card) 요청

폼 내부에서 삭제 버튼 클릭 → deleteCard(card) 호출

CardEditorForm.jsx

## 7. CardEditor 분석: “Object.keys(cards)”를 쓰는 이유

cards가 배열이 아니라 객체(Object) 형태라서, key 목록을 뽑아 map으로 돌립니다.

CardEditor.jsx

장점: updated[card.id] = card처럼 id로 빠르게 접근/수정 가능

Marker.jsx

수업 질문거리: “배열로 만들면 뭐가 달라질까?”

배열이면 find/map/filter 필요(수정 시 탐색 비용 + 코드 길어짐)

Marker.jsx

## 8. Preview 분석: 보여주기 전용 컴포넌트

Preview는 cards를 받아서 그대로 Card로 출력만 합니다(상태 없음).

Preview.jsx

- Editor: 입력/수정/삭제(행동)
- Preview: 출력(UI)(표현)

## 9. Login/인증 흐름(간단 정리)

Login 화면에서 Google 버튼 클릭 시:

(1)loginGoogle(authLogic.auth, authLogic.googleProvider) 호출

Login.jsx

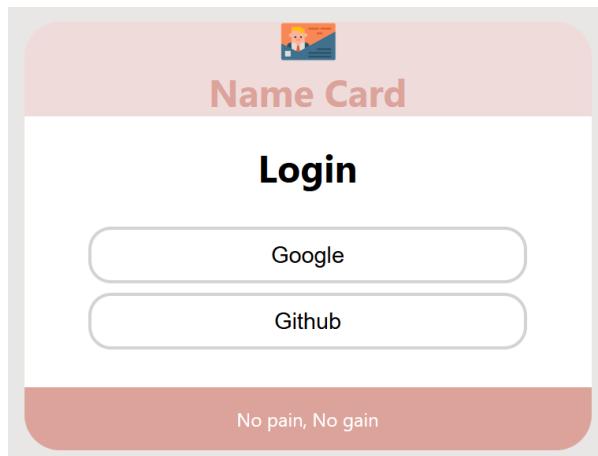
(2)성공하면 user.uid를 state에 저장하고 /marker로 이동

Login.jsx

Marker에서는 onAuthChange(authLogic.auth)로 로그인 상태를 확인해서,

- 로그인 되어 있으면 userId 저장
- 아니면 /로 돌려보냄

Marker.jsx



The interface consists of two main sections: "Card Editor" on the left and "Card Preview" on the right.

**Card Editor:** This section contains four groups of card components, each with a profile photo placeholder, name, position, email, and a "Delete" button.

- Group 1:** 이순신 (조선), colorful, lee@hot.com, 엔지니어, 죽고자해야 산다, man2, Delete.
- Group 2:** 유관순 (대전), light, 기획자, yoo@hot.com, 하고야 마다, woman1, Delete.
- Group 3:** 강감찬 (부산), dark, 페블리서, pub@hot.com, 할 수 있다, man2, Delete.
- Group 4:** Name (Company), light, Title, Email, Message, No file, Add.

**Card Preview:** This section shows four preview cards corresponding to the editor groups.

- Card 1:** 이순신 (조선), lee@hot.com, 죽고자해야 산다.
- Card 2:** 유관순 (대전), yoo@hot.com, 하고야 마다.
- Card 3:** 강감찬 (부산), pub@hot.com, 할 수 있다.
- Card 4:** Name (Company) - This card has a black background and displays the message "No file".

A red banner at the bottom of the screen says "No pain, No gain".