

디지털 명함 시스템 구축 강의 계획서

디지털 명함 시스템 구축 강의 계획서

강의 목표

전체 시스템 프로세스 도식 요약

[1] 전체 제작 과정 도식 요약

[기획 및 요청]



[디지털 명함 프로토타입 제작 (index.html)]



[Firebase 구성: Firestore + Storage + Auth]



[Kakao API 연동하여 위치 주소화]



[관리자 페이지 제작 (admin.html)]



[QR 생성 및 GitHub Pages 배포]



[배포 URL 을 도우미에게 제출하여 테스트 요청]



[최종 ZIP 및 문서화 완료]

[기획 및 요청]

↓

[디지털 명함 프로토타입 제작 (index.html)]

↓

[Firebase 구성: Firestore + Storage + Auth]

↓

[Kakao API 연동하여 위치 주소화]

↓

[관리자 페이지 제작 (admin.html)]

↓

[QR 생성 및 GitHub Pages 배포]

↓

[최종 ZIP 및 문서화 완료]

[2] 최종 결과물 작동 구조 요약

[사용자 QR 스캔]

↓

[index.html 열림]

↓ 사용자 이름/이미지 입력

↓ 위치(GPS) 수집

↓ Kakao API 로 주소 변환

↓ Firebase 에 정보 저장

↓

[관리자 페이지 (admin.html)]

↳ 로그인 (Firebase Auth)

↳ scan_records 컬렉션에서 실시간 조회

↳ 필터/날짜 검색, 엑셀 다운로드, 접속 기록까지 확인 가능

[QR 스캔]

↓

[index.html (디지털 명함)]

↓ 사용자 이름/사진 입력

↓ 위치(GPS) 수집

↓ Kakao API 로 주소 변환

↓ Firebase (Firestore + Storage)에 저장

↓

[admin.html (관리자 페이지)]

↳ Firebase Auth 로그인

↳ 명함 기록 조회 (scan_records)

↳ 이름/주소 필터, 날짜 필터, 차트 확인, 엑셀 다운로드

↳ 접속 기록 저장 (logins)

- QR 코드로 연결되는 웹 기반 디지털 명함을 구축한다.
- 명함을 받은 사람의 위치와 정보를 기록하는 관리자 시스템을 만든다.
- Firebase 와 Kakao API 를 연동해 실시간 데이터 저장 및 주소 변환을 구현한다.
- 기록 이력을 확인할 수 있는 관리자 페이지까지 완성한다.

 강의 구성

1 프로젝트 개요 및 목표 설정

질문: "내 프로필을 담은 QR 코드를 만들고 싶어"

- 사용자의 명함 정보를 기반으로 QR 코드 생성
- GitHub Pages 를 활용한 웹 배포 계획 수립

2 디지털 명함 프로토타입 제작

질문: "상대방이 캡처하면 바로 전화번호 입력창으로 전환되는 기능이 있으면 좋겠어"

- index.html 제작
- 전화번호, 이메일, 주소, 링크, vCard 다운로드 버튼 구현

[사용된 프롬프트]

내 명함 정보를 담은 HTML 을 만들어줘. 전화번호, 주소, 회사, 링크가 포함되고 깔끔한 디자인이면 좋겠어.

3 Firebase 연동

질문: "Firebase 에 기록을 저장하고 싶어"

- Firebase 프로젝트 생성
- Firestore, Storage 연동
- 명함 수신자 정보(이름, 이미지)를 업로드하는 기능 구현

[사용된 프롬프트]

Firestore 와 Storage 를 사용해서 이름, 이미지, 시간정보를 기록하는 코드를 작성해줘.

4 Kakao API 를 통한 위치 → 주소 변환 구현

질문: "교부한 위치를 주소 형태로 저장하고 싶어"

- navigator.geolocation 을 활용하여 좌표 수집
- Kakao REST API 로 주소 변환
- Firestore 에 주소 필드 저장 추가

[사용된 프롬프트]

이 위치정보를 Kakao API 로 주소로 바꿔서 Firestore 에 저장되게 해줘.

5 관리자 페이지 구현

질문: "명함 교부 기록을 웹에서 확인하고 싶어"

- 관리자 전용 admin.html 페이지 제작
- Firebase SDK 를 활용해 scan_records 컬렉션에서 데이터 불러오기
- HTML 테이블로 이름, 주소, 이미지, 시간 정보 출력
- 검색 필터 (이름/주소) 및 날짜 필터 기능 구현
- 엑셀 다운로드 버튼 추가 (SheetJS 라이브러리 사용)
- Firebase Auth 로그인 인터페이스 구현

관리자 페이지 구성 요소:

- loginSection: 이메일/비밀번호 입력창
- adminSection: 로그인 후 기록 확인 테이블
- 필터 입력창, 차트, 통계 카운트, 로그아웃 버튼 등 포함

질문: "명함 교부 기록을 웹에서 확인하고 싶어"

- 관리자용 admin.html 구현
- Firestore scan_records 조회
- 이름, 주소, 이미지, 시간 출력

6 관리자 기능 강화 요청 반영

질문: "다음 기능을 추가해줘: 필터 검색, 엑셀 다운로드, 로그인"

- 필터: 이름/주소로 검색
- 다운로드: SheetJS 로 Excel 출력
- 로그인: Firebase Auth 이메일/비밀번호 방식 연동

[사용된 프롬프트]

관리자 페이지에 이름, 주소 필터 기능과 엑셀로 내보내기 버튼을 추가해줘. Firebase Auth 로그인도 필요해.

7 접속 기록 저장 기능

질문: "로그인한 관리자 접속 기록도 Firestore 에 남기고 싶어"

- Auth 로그인 성공 시 logins 컬렉션에 기록 저장

8 최종 배포 및 ZIP 패키징

질문: "모든 파일을 ZIP 으로 묶어줘" 질문: "모든 파일을 ZIP 으로 묶어줘"

- index.html
- admin.html
- QR 이미지
- 프로필 사진
- README.txt 포함된 최종 ZIP 제공

9 정상 작동 유무 확인

수강생은 사이트를 GitHub Pages 등에 배포한 후, 그 URL 을 도우미(ChatGPT)에게 전달하여 작동 유무를 대신 확인받을 수 있습니다.

💬 프롬프트 입력 방식

아래와 같은 문장을 ChatGPT 에게 입력하여 요청합니다:

<https://username.github.io/project-namecard/> 사이트에 디지털 명함을 배포했습니다.
정상 작동하는지 확인해 주세요.

- vCard 다운로드 및 저장 화면 전환

- 명함 받기 전송 → Firebase 기록

- 주소 저장 (Kakao API 연동)

이 3 가지가 작동하는지 확인 바랍니다.

ChatGPT 는 해당 사이트에 직접 접속하여 아래 항목을 테스트한 뒤, 작동 유무를 바로 확인해줍니다.

※ 주소는 예시이며, 실제 배포 주소를 전달하면 ChatGPT 가 직접 아래와 같이 확인해 줍니다.

✅ 실제 확인 예시 화면

🔗 확인 URL: <https://username.github.io/project-namecard/>

📎 연락처 저장 버튼 클릭 시 → Kim_Gil_Ho.vcf 파일 다운로드 → 연락처 앱 실행 여부 확인됨

📄 '명함 받기 전송' 버튼 클릭 → Firebase scan_records 컬렉션에 새 문서 저장 확인됨

📍 위치 좌표 → Kakao API 로 도로명 주소 변환 후 Firestore 저장 확인됨

위처럼 확인 결과를 상세히 응답받을 수 있습니다.

🌐 배포 및 연결 사이트 구성

✅ GitHub Pages 배포

1. GitHub 에서 새 저장소 생성 (예: yourproject-namecard)
2. index.html, admin.html, profile.jpg, QR 이미지 등을 업로드
3. 저장소 설정 → Pages → main 브랜치의 루트 경로 선택
4. 주소 예시: <https://your-username.github.io/yourproject-namecard/>

✅ Firebase 구성

- 콘솔 주소: <https://console.firebase.google.com/>
1. Firebase 에 로그인 후 새 프로젝트 생성 (예: yourproject-namecard)
 2. '프로젝트 개요 > 웹앱 추가'에서 웹 앱 등록 및 Firebase SDK 설정 정보 복사

3. Firestore Database 메뉴에서 '시작하기' 클릭 → 보안 규칙 설정 후 scan_records, logins 컬렉션 자동 생성됨
4. Storage 메뉴에서 '시작하기' 클릭 → 이미지 저장을 위한 버킷 활성화
5. Authentication 메뉴에서 '로그인 방법' > '이메일/비밀번호' 방식 활성화
6. 사용자 추가 메뉴에서 관리자 이메일 계정 등록 (예: admin@yourdomain.com)

※ Firestore 구조 예시:

- scan_records: 이름, 이미지, 주소, 위도, 경도, timestamp
 - logins: email, uid, userAgent, timestamp
 - 콘솔 주소: <https://console.firebase.google.com/>
1. 새 프로젝트 생성 (예: sudesign-namecard)
 2. Firestore Database → 설정 후 컬렉션 자동 생성됨
 3. Storage 활성화 → 이미지 저장 용도
 4. Authentication → 이메일/비밀번호 로그인 활성화

✅ Kakao API Key 발급 과정

- Kakao Developers: <https://developers.kakao.com>
1. 애플리케이션 생성
 2. 플랫폼 설정 없이 REST API 키 발급
 3. "카카오 지도 API > 좌표 → 주소 변환" 기능 사용
 4. 이 키를 다음과 같이 HTML 에 삽입:

Authorization: 'KakaoAK 123abc456def789ghi0jklmnopqrstu' // (예시입니다)

⚠ ※ 위 API 키는 실제 유효하지 않은 무작위 예시입니다. 실습 시 반드시 직접 발급받은 REST API 키로 대체해야 합니다.