찐대충인
로컬 인증 기반 지역 커뮤니티 서비스

SOGRA 최종발표

팀: 구미라면축제만관부

Profile Presentation

목차

1. 문제 인식

기존 웹 의 문제

1. 정보 신뢰성 부족 - 광고/비전문가 후기 혼재
2. 로컬성 결여 - 전국 단위로 지역 특성 반영 안됨

⇒

해결 제안

"로컬 인증 기반의 커뮤니티형 관광 앱"

실제 지역민이 검증한 정보만 모아 보여주는 서비스

2. 아이디어 개요

서비스명: 찐대충인

"로컬 인증으로 신뢰를 더한 진짜 대전·충청 이야기"

3가지 핵심가치

- 로컬 인증: GPS 기반 '로컬 뱃지'로 신뢰 보장
- 지도 시각화: 게시물 지도 핀으로 직관적 탐색
- 커뮤니티: 지역민의 생생한 후기 팁 공유

3. 주요기능

1. 회원가입 & 로컬 인증

- 사용자가 회원가입 시 현재 위치를 확인
- 대전·충청 지역 내 좌표로 판별되면 자동으로 '로컬 인증'
- React + Firebase Auth + Geolocation API를 이용

2. 게시판 (카테고리 기반)

- '맛집·교통·핫플·꿀팁' 4개 카테고리
- 게시물은 제목, 내용, 사진, 위치정보를 함께 등록
- 이미지 업로드는 Firebase Storage

3. 지도 보기 (시각화 기능)

- Google Maps API를 활용해 게시물을 지도 상의 마커
- 마커 클릭 시 게시물 사이드바 노출

4. 상호작용 기능 (피드백 시스템)

- 게시물에 좋아요, 댓글, 조회수 기능
- 실시간 반응 데이터를 업데이트

5. 통계 / 인사이트 (확장 기능)

- 조회수 통계로 시각화
- `찐명소' 탭에서는 대전·충남·충북의
인기 명소 TOP 5를 매월 1일 자동 갱신

4. 아키텍처

Frontend

- React (Vite)
- Tailwind CSS
- Geolocation API 연동

외부 API

- Google Maps API
- Geolocation API (Browser)

Backend

- Firebase Authentication — 이메일 기반 로그인
- Firestore (Database) — 사용자 / 게시물 / 댓글 관리
- Firebase Storage — 이미지 파일 저장 (크기/타입 검증 규칙 적용)

5. 보안 설계

XSS 공격 방지

```
// 사용자 입력 정제
const sanitizeInput = (input) => {
  if (!input) return ''
  return input.trim()
    .replace(/[<>]/g, '')           // HTML 태그 제거
    .replace(/javascript:/gi, '')   // JS 프로토콜 제거
    .replace(/on\w+=/gi, '')        // 이벤트 핸들러 제거
    .replace(/data:/gi, '')         // Data URI 제거
}
```

파일 업로드 공격 방지

```
// 파일 검증
const MAX_FILE_SIZE = 5 * 1024 * 1024 // 5MB
const ALLOWED_TYPES = ['image/jpeg', 'image/png', 'image/gif', 'image/webp']
const allowedExtensions = ['jpg', 'jpeg', 'png', 'gif', 'webp']

// 타입, 크기, 확장자 검증
if (!ALLOWED_TYPES.includes(file.type)) return
if (file.size > MAX_FILE_SIZE) return
if (!allowedExtensions.includes(fileExtension)) return

// 안전한 파일명 생성
const safeFileName = `${userId}_${Date.now()}.${extension}`
const sanitizedFileName = safeFileName.replace(/[^a-zA-Z0-9._-]/g, '_')
```

경로 탐색 공격 방지

```
// 파일명 검증
if (file.name.includes('..') || file.name.includes('/')) {
  setError('파일 이름에 허용되지 않는 문자가 포함되어 있습니다.')
  return
}
```

봇 공격 방지

```
// 사람 눈에 안 보이는 숨겨진 입력 필드
const honeypotRef = useRef(null)

// 봇이 자동으로 채우면 차단
if (honeypotRef.current && honeypotRef.current.value) {
  setError('의심스러운 요청입니다.')
  return
}
```

입력 길이 제한

```
// 길이 검증
if (formData.title.trim().length < 2) return
if (formData.title.trim().length > 100) return
if (formData.content.trim().length < 10) return
if (formData.content.trim().length > 1000) return
```

브루트 포스

```
// 로그인 시도 횟수 제한 (지수 백오프)
const recordFailure = () => {
  const at = getAttempts()
  at.count = (at.count || 0) + 1

  const threshold = 5
  if (at.count >= threshold) {
    // 지수 백오프: 60초 → 120초 → 240초...
    const exponent = Math.max(0, at.count - threshold)
    const lockSec = 60 * Math.pow(2, exponent)
    at.lockUntil = Date.now() + lockSec * 1000
```

감사합니다.