

무료 구독자판

누구나 따라 하는 쉬운 웹앱 제작 실전 가이드

코딩 없이 아이디어부터 배포까지, 10단계로 끝내는 첫 웹앱

컴맹 집사 (와우굿앱)
2026년 10월 16일

표지 geralt / Pixabay

들어가며: 이 책을 쓴 사람과 읽는 방법

나는 코딩을 배운 적이 없다. 컴퓨터로 하던 일은 대부분 문서 작업이었다. 그런 내가 2026년 9월, AI와 함께 반려인을 위한 웹앱 "왕립 개집사협회(RKBA)"를 만들어 인터넷에 올렸다. 첫 버전을 공개하기까지 하루 한 시간씩 약 일주일이 걸렸고, 그 뒤로 사진 올리기, 작위증 저장, 공유, 웹툰, 미니게임, 아카데미, 휴대폰 설치(PWA)까지 하나씩 붙여 나갔다.

이 책은 두 가지를 함께 담았다.

- 일반 가이드: 웹앱을 처음 만드는 사람이라면 누구에게나 필요한 순서와 개념.
- 개집사 앱 사례: 실제로 내가 어디서 막혔고, 어떻게 풀었는지. 각 장의 「개집사 앱에서는」 상자에 적었다.

전문 개발자의 강좌가 아니다. 대신 초보자가 실제로 어디서 넘어지는지는 누구보다 잘 보여 줄 수 있다고 생각한다. 각 장 끝에는 「AI에게 이렇게 말해 보세요」 예시와 「점검 목록」을 두었다. 읽고 바로 따라 해 볼 수 있게 하려는 것이다.

이 책에 등장하는 앱, 왕립 개집사협회

강아지는 왕(폐하), 반려인은 왕을 모시는 집사라는 세계관의 웹앱이다. 폐하의 이름과 성격을 입력하고 집사로서 질문 7개에 답하면, 집사 유형과 작위(견국 수습집사부터 왕실 총집사까지 5단계)가 적힌 작위증을 받는다. 쓸모를 따지는 앱은 아니지만, 반려인이라면 한 번쯤 해 보고 친구에게 자랑하고 싶은 앱을 목표로 했다.



왕립 개집사협회 앱 홈 화면 (출처: 2026년 10월 16일 버전 화면, 직접 캡처)

목차

1. 웹앱과 노코드·AI 개발의 시대
2. 내 아이디어 정리와 서비스 기획
3. 화면 흐름과 와이어프레임 설계
4. 나에게 맞는 도구 고르기
5. 화면 구현하기
6. 데이터 저장과 로직 만들기
7. AI에게 일 잘 시키는 법 (프롬프트)
8. 로그인·결제·도메인, 외부 서비스 연결
9. 테스트, 오류 해결, 배포
10. 사람들에게 알리고 수익 내기
11. 맺음말
12. 부록: 용어 사전, 프롬프트 모음, 전체 점검 목록, 출처

1장. 웹앱과 노코드·AI 개발의 시대

1.1 웹앱이란 무엇인가

웹앱(Web App)은 크롬, 사파리 같은 웹 브라우저에서 실행되는 앱이다. 겉모습과 쓰는 방법은 휴대폰 앱과 비슷하지만, 앱스토어에서 내려받아 설치할 필요 없이 인터넷 주소(URL)만 열면 바로 쓸 수 있다.

웹앱은 크게 세 부분으로 이루어진다.

부분	하는 일	쉬운 비유
클라이언트(화면)	사용자가 보고 누르는 부분	가게의 진열대와 계산대
서버	데이터를 처리하고 계산하는 뒤쪽 시스템	가게 뒤편의 주방
데이터베이스	회원 정보, 글 같은 데이터를 보관하는 곳	창고

중요한 사실 하나. 모든 웹앱에 서버와 데이터베이스가 필요한 것은 아니다. 화면만으로 동작하는 앱도 충분히 쓸모 있고, 초보자는 오히려 그쪽에서 시작하는 게 좋다. 이 이야기는 2장과 6장에서 다시 한다.

1.2 왜 웹앱으로 시작할까: 네이티브 앱과 비교

구분	웹앱	네이티브 앱
만드는 언어	HTML, CSS, JavaScript 또는 노코드 도구	Swift(아이폰), Kotlin(안드로이드)
설치	필요 없음 (주소로 접속)	필요함 (스토어에서 내려받기)
만드는 기간	며칠에서 몇 주	몇 달 이상인 경우가 많음

구분	웹앱	네이티브 앱
업데이트	올리는 즉시 모두에게 반영	스토어 심사를 거쳐야 함
시작하기	쉬움	어려움

웹앱으로 시작해도 나중에 길이 막히지 않는다. 웹앱을 휴대폰 홈 화면에 설치할 수 있게 만드는 PWA(프로그레시브 웹앱) 기술이 있고, 이것을 포장해서 Google Play에 올리는 방법도 있다. (출처: MDN 웹 문서, Progressive web apps)

> 🐾 **개집사 앱에서는**
> 처음에는 "앱"이라고 하면 당연히 앱스토어에 올리는 것인 줄 알았다. AI에게 물어보니 웹앱으로 먼저 만들면 개발자 등록이나 심사 없이 오늘 만든 것을 오늘 친구에게 보낼 수 있다고 했다. 실제로 그렇게 했다. 링크 하나를 카카오톡으로 보내는 것이 첫 공개였다. 휴대폰 설치(PWA)는 그다음 단계에서 붙였고, Google Play 등록은 지금 준비 중이다.

1.3 코딩 없이 만드는 시대, 그러나 "할 일이 없어진" 것은 아니다

예전에는 웹앱 하나에도 화면 개발자, 서버 개발자, 데이터베이스 담당자가 필요했다. 지금은 노코드 도구와 생성형 AI 덕분에 혼자서도 몇 시간 만에 동작하는 앱을 만들 수 있다.

다만 내가 직접 해 보고 느낀 것은, 코딩이 사라진 자리에 다른 일이 생긴다는 점이다.

- 무엇을 만들지 정하는 일
- AI가 만든 결과를 눌러 보고 확인하는 일
- 이상한 점을 말로 정확히 설명하는 일

> **🐾 개집사 앱에서는**
>
> 첫 버전을 만드는 동안 시간을 나눠 보니 이랬다. AI에게 요청하고 코드를 받는 시간은 전체의 10~20% 정도로 짧았다. 받은 결과를 열어 보고 확인하는 시간이 약 30%, 문제를 설명하고 고치는 시간이 40% 가까이였다. 나머지는 GitHub, 배포 같은 처음 보는 개념을 이해하는 시간이었다. "AI가 몇 초 만에 앱을 만들어 준다"는 말은 맞지만, 그건 전체 과정의 일부다.

내 의견: 이 사실을 미리 알고 시작하면 첫 며칠 동안 훨씬 덜 조급하다. "AI는 만들고, 나는 검사한다"는 역할 분담을 처음부터 받아들이자.

2장. 내 아이디어 정리와 서비스 기획



사진: jplenio / Pixabay

2.1 좋아하는 분야에서 시작하라

거창한 앱보다 내가 좋아하고 잘 아는 분야의 앱 하나를 먼저 떠올려 보길 권한다. 좋아하는 분야일수록 아이디어가 끊이지 않고, 막혀도 끝까지 가 보고 싶어진다. 배우겠다는 마음보다 만들고 싶다는 마음이 훨씬 오래간다. 나에게는 그게 강아지와 반려인이었다.

2.2 문제와 대상 정하기

웹앱이 무엇을 해결하는지, 혹은 누구에게 어떤 즐거움을 주는지 한 줄로 정한다.

- 문제형 예시: "동네 주민끼리 중고 물건을 직거래할 때 믿을 수 있는 상대를 찾기 어렵다." → 대상: 안 쓰는 물건을 가까운 곳에서 빨리 정리하고 싶은 주민
- 재미형 예시: "반려인은 스스로를 '집사'라고 부르며 즐거워한다." → 대상: 반려견과 사는 사람

모든 앱이 문제를 해결해야 하는 것은 아니다. 재미, 자랑하고 싶은 마음, 추억도 충분한 이유가 된다. 중요한 것은 "누가 이걸 왜 쓰는가"가 분명한 것이다.

2.3 앱을 한 문장으로 정의하기

누가, 무엇을 해서, 무엇을 얻는지가 들어가게 한 문장을 쓴다. 이 한 문장이 기획서의 전부일 수도 있다.

> 🐾 **개집사 앱에서는**
> 처음에는 "강아지가 왕이고 내가 집사인 앱 만들어 줘"라는 한 문장만 AI에게 던졌다. 질문은 세 개뿐이었고, 결과는 "당신은 훌륭한 집사입니다!" 한 줄이 전부였다. AI가 잘못한 게 아니었다. 내가 원하는 걸 나도 정확히 몰랐다.
> 그래서 다시 썼다. "반려인이 우리 강아지 폐하의 정보를 넣고 집사로서 질문 7개에 답하면, 나만의 반려견 왕국과 집사 작위가 적힌 작위증을 받는 웹앱."
> 이렇게 쓰고 나니 "폐하 정보는 무엇을 받지?", "집사 유형은 몇 가지?", "작위증에는 뭐가 들어가지?" 같은 질문이 저절로 생겼다.

2.4 MVP: 처음 버전에는 꼭 필요한 것만

MVP(최소 기능 제품)는 사용자의 반응을 확인하는 데 필요한 핵심 기능만 담은 첫 제품이다. (출처: Wikipedia, Minimum viable product) 기능을 세 칸으로 나누면 정리가 쉽다.

칸	뜻	중고거래 앱 예시
꼭 필요함	없으면 앱이 성립하지 않음	물건 등록, 목록 보기, 연락하기

칸	뜻	중고거래 앱 예시
있으면 좋음	없어도 쓰는 데 지장 없음	거래 후기 점수
이번엔 제외	처음에는 생각하지 않음	실시간 방송 판매

> 🐾 **개집사 앱에서는**
> 처음 버전(반드시): 폐하 정보 입력, 질문 7개와 집사 유형 판정, 작위 5단계, 작위증 화면. 나중에(있으면 좋음): 사진 업로드, 이미지로 저장, 공유하기, 이전 결과 보기, 웹툰, 미니게임. 실제로 "나중에" 칸의 기능은 첫 공개 뒤 몇 주에 걸쳐 하나씩 붙였다. 처음부터 다 넣었다면 어디서 문제가 생겼는지 찾지 못했을 것이다.

2.5 초보자를 위한 한 가지 기준: 브라우저 안에서 끝나는 앱부터

이건 내 경험에서 나온 가장 중요한 조언이다. 아이디어를 두 종류로 나눠 보자.

처음 만들기 좋은 앱	나중에 도전할 앱
테스트·진단 앱, 계산기, 체크리스트	회원끼리 글을 주고받는 커뮤니티
결과 이미지를 만들고 저장하는 앱	개인정보를 서버에 모으는 앱
정해진 정보를 보여 주는 안내 앱	결제가 들어가는 앱

왼쪽은 사용자의 휴대폰 브라우저 안에서만 동작하는 앱이고, 오른쪽은 서버와 데이터베이스, 보안이 필요한 앱이다. 왼쪽으로 첫 앱을 완성해 보면 오른쪽에 필요한 개념도 훨씬 쉽게 들어온다. 개집사 앱도 왼쪽에서 시작했다.

AI에게 이렇게 말해 보세요

```
이 아이디어로 웹앱을 만들고 싶어: (아이디어 한 문장)
1. 사용자가 앱을 열고 닫을 때까지의 순서를 짜 줘.
2. 빠진 화면이 있으면 알려 줘.
3. 처음 버전에 꼭 필요한 기능과 나중에 해도 되는 기능을 나눠 줘.
코딩은 아직 하지 말고 기획만 같이 정리하자.
```

점검 목록

- ☐ 앱을 한 문장으로 설명할 수 있다.
- ☐ 누가 쓰는지 말할 수 있다.
- ☐ 첫 버전 기능이 3~5개 안쪽이다.
- ☐ 첫 버전에 로그인·결제가 없다(가능하면).

3장. 화면 흐름과 와이어프레임 설계



사진: Pexels / Pixabay

3.1 사용자 흐름(User Flow) 적기

사용자가 앱에 들어와서 목적을 이룰 때까지 거치는 순서를 화살표로 적는다.

중고거래 앱 예시: `홈 → 로그인 → 상품 목록 → 상품 상세 → 채팅`

> **🐾 개집사 앱에서는**
>
> 앱 열기 → 시작 화면 → 폐하 정보 입력(이름·성격·좋아하는 것·싫어하는 것·왕국 분위기) → 질문 7개 → 집사 유형과 작위 확인 → 작위증 받기 → 저장 또는 공유 순서를 적어 보니 처음 계획에는 "폐하 정보 입력"이 빠져 있었다. 폐하 이름이 없으면 "○○ 폐하의 왕국"도, 작위증의 이름 칸도 만들 수 없다. 이런 구멍은 코드를 받은 뒤에 발견하면 고치기 번거롭다. 흐름을 적는 10분이 나중의 몇 시간을 아껴 준다.

3.2 화면 목록과 와이어프레임

흐름을 바탕으로 화면을 나누고, 각 화면에 무엇이 들어가는지 상자와 글자로만 대충 그린다. 이것이 와이어프레임이다.

- 종이와 펜: 가장 빠르다. 처음에는 이것으로 충분하다.
- Miro, Whimsical: 흐름도를 그리고 다른 사람과 함께 보기 좋다.
- Figma: 디자이너와 노코드 개발자가 널리 쓰는 도구. 정교한 화면을 만들 때 쓴다.

내 의견: AI로 만드는 경우라면 정교한 와이어프레임보다 "화면별로 무엇이 있는지 적은 목록"이 더 쓸모 있다. AI는 그림보다 글로 된 설명을 더 정확히 따른다. 손그림을 사진으로 찍어 AI에게 보여 주는 방법도 잘 통한다.

3.3 초보자가 자주 하는 디자인 실수 5가지

1. 한 화면에 너무 많이 담기. 시선이 흩어져 무엇을 눌러야 할지 모른다. 한 화면에는 한 가지 할 일.
2. 주요 버튼이 눈에 안 띔. "시작하기", "등록하기" 같은 버튼은 가장 눈에 띄는 색과 자리에.
3. 휴대폰 화면을 나중에 생각함. 우리나라 웹 이용의 절반 가까이가 휴대폰에서 일어난다. 2026년 9월 기준 국내 웹 접속 비율은 데스크톱 약 51%, 휴대폰 약

48%였다. (출처: StatCounter, 한국 기기별 점유율) 앱처럼 휴대폰으로 공유되는 서비스라면 실제 비중은 더 높을 수 있으니, 처음부터 휴대폰 화면 기준으로 만들자.

4. 글꼴과 색이 제각각. 대표 색은 2~3개 안쪽으로.
5. 가입 절차가 길다. 처음에는 가입 없이 써 볼 수 있게 하는 것이 가장 좋다.

> 🐾 **개집사 앱에서는**
>
> 결과 화면에 작위를 처음에는 "1등급, 2등급"으로 표시했다. 그런데 보통 "1등급"은 가장 높은 것으로 받아들인다. 우리 앱의 1은 가장 처음 단계인 수습집사였다. "등급"을 "단계"로 바꾸자 오해가 사라졌다. 디자인은 색과 모양만이 아니라 단어 선택도 포함된다.

점검 목록

- ☐ 사용자 흐름을 화살표로 적었다.
- ☐ 화면마다 들어갈 것을 목록으로 적었다.
- ☐ 휴대폰 화면 기준으로 생각했다.
- ☐ 화면마다 가장 중요한 버튼이 하나로 정해져 있다.

4장. 나에게 맞는 도구 고르기



사진: fietzfotos / Pixabay

도구는 크게 세 갈래다. 어느 것이 정답이라기보다, 만들려는 앱과 내 성향에 맞는 것을 고르면 된다.

4.1 데이터 중심 앱: Glide, Airtable

구글 시트나 에어테이블 표에 데이터를 정리해 두면, 몇 번의 클릭으로 앱 모양으로 바꿔 주는 도구다. 맛집 목록, 사내 관리표, 강의 목차처럼 "표를 보기 좋게 보여 주는" 앱에 잘 맞는다.

4.2 자유도 높은 노코드: Bubble, FlutterFlow

거의 모든 형태의 서비스를 만들 수 있는 도구다. Bubble은 웹앱의 데이터와 동작을 화면에서 끌어다 놓는 방식으로 세밀하게 정할 수 있고, FlutterFlow는 휴대폰 앱과 웹앱을 함께 만들 수 있다. 대신 도구 자체의 사용법을 익히는 데 시간이 든다.

4.3 AI가 코드를 만들어 주는 방식: v0, Bolt.new, Replit, AI 채팅

말로 설명하면 화면과 코드를 만들어 주는 도구들이다.

- v0: Vercel이 만든 AI 앱 빌더. 2025년 v0.dev에서 v0.app으로 이름을 바꾸며 화면 생성 도구에서 전체 앱을 만드는 도구로 넓어졌고, Vercel 배포와 바로 이어진다. (출처: v0 공식 사이트)
- Bolt.new: StackBlitz가 만든 도구로, 브라우저 안에서 앱을 만들고 바로 실행해 볼 수 있다. 사용량(토큰)만큼 차감되는 방식이라 오류를 고치려고 같은 요청을 반복하면 사용량이 빨리 줄어든다. (출처: Bolt 공식 사이트)
- 일반 AI 채팅(ChatGPT, Gemini, Claude 등): 코드를 받아 내 컴퓨터에 파일로 저장하고, 직접 올리는 방식. 도구에 묶이지 않고, 앱이 커져도 내 파일로 남는다.

4.4 한눈에 비교

구분	Glide 계열	Bubble 계열	AI 빌더(v0, Bolt)	AI 채팅 + 내 파일
난이도	매우 쉬움	보통~어려움	쉬움	보통
자유도	제한적	매우 높음	높음	매우 높음
첫 결과까지	1~2시간	며칠	몇 시간	몇 시간~며칠
비용	무료로 시작, 기능별 유료	무료로 시작, 공개 시 유료 많음	무료 사용량 + 유료	AI 구독 외 거의 없음

구분	Glide 계열	Bubble 계열	AI 빌더(v0, Bolt)	AI 채팅 + 내 파일
잘 맞는 앱	목록·안내 앱	회원제 서비스	시제품, 화면 위주 앱	내가 오래 키울 앱
도구를 떠날 때	어려움	어려움	코드 내보내기 가능	처음부터 내 것

요금은 자주 바뀌니 시작 전에 각 도구의 요금 안내를 꼭 직접 확인하자.

4.5 내 의견: 컴맹 집사가 고른 길과 추천

나는 마지막 칸, AI 채팅 + 내 파일 방식을 골랐다. 이유는 세 가지였다.

1. 돈이 거의 들지 않았다. 컴퓨터, AI 서비스 계정, 크롬 브라우저, 메모장, 확인용 휴대폰이 전부였다.
2. 앱이 커져도 도구에 묶이지 않았다. 웹툰, 게임, 영어판까지 붙이면서 이 장점이 컸다.
3. 실제로 무슨 일이 일어나는지 조금씩 이해하게 됐다. 파일, 저장, 배포의 큰 흐름을 알게 되자 AI에게 설명하는 말도 정확해졌다.

> **🐾 개집사 앱의 실제 도구 조합**
> AI 채팅(코드 작성·오류 해결, 여러 AI를 역할에 따라 나눠 씀) → index.html 파일 하나 → GitHub(파일 보관과 변경 기록) → Vercel(인터넷 주소와 자동 배포). 비용은 AI 구독료 외에는 거의 없었다. 나중에 도메인을 따로 연결했다.

추천은 이렇게 정리할 수 있다.

- 일주일 안에 표 형태의 앱이 필요하다면: Glide 계열.
- 회원제 서비스를 사업으로 키울 계획이라면: Bubble 계열, 또는 개발자의 도움.
- 시제품을 빨리 보고 싶다면: v0, Bolt.new.

- 취미로 시작해 오래 키울 앱이라면: AI 채팅 + 내 파일 + GitHub + Vercel. 처음 며칠은 낯설지만, 가장 많이 배운다.

5장. 화면 구현하기

5.1 레이아웃: 위, 가운데, 아래

웹페이지는 세 구역으로 나누면 정리가 쉽다.

- 머리(Header): 로고, 메뉴, 로그인 버튼
- 본문(Body): 핵심 내용, 첫 화면의 큰 소개, 기능, 목록
- 꼬리(Footer): 운영자 정보, 이용약관, 개인정보처리방침

5.2 화면을 이루는 부품

- 글자: 제목과 본문 크기를 분명히 구분해 위아래 순서를 보여 준다.
- 입력 칸: 이름, 이메일, 숫자를 받는 칸. 무엇을 넣는지 안내 문구를 넣는다.
- 버튼: 누르면 무엇이 일어나는지 버튼 글자만 보고 알 수 있게.
- 이미지: 크기와 용량을 줄여서 넣는다(9장 참고).
- 목록·카드: 같은 모양의 상자를 반복해 여러 항목을 보여 준다.

5.3 첫 화면은 3초 안에 무슨 앱인지 알려야 한다

처음 들어온 사람은 오래 기다려 주지 않는다. 첫 화면에는 세 가지만 확실히 보이면 된다. 무슨 앱인지(제목과 한 줄 설명), 무엇을 하면 되는지(순서 안내), 어디를 누르면 되는지(시작 버튼).

> **🐾 개집사 앱에서는**
> 첫 화면에 "정보 입력 → 질문 7개 → 작위증 발급"이라는 세 단계 안내를 넣었다. 처음 와 본 친구도 무엇을 하는 앱인지 묻지 않게 됐다. 홈 화면은 기능이 늘 때마다 바뀌었다. 9월 19일 첫 버전과 10월의 화면을 나란히 놓으면 같은 앱이라고 믿기 어려울 정도다.



9월 19일 첫 버전부터 10월 12일까지 첫 화면의 변화 (출처: 앱의 시기별 화면을 직접 캡처해 비교 이미지로 제작)

5.4 휴대폰과 PC 모두에서 보기 좋게(반응형)

화면 크기에 따라 요소가 줄어들거나 세로로 쌓이게 만드는 것을 반응형이라고 한다. 두 가지 개념만 알면 AI에게 정확히 부탁할 수 있다.

- 가로 배치와 세로 배치: PC에서는 카드를 옆으로, 휴대폰에서는 아래로 쌓는다. (출처: MDN 웹 문서, Flexbox)
- 최대 너비: 큰 모니터에서 화면이 끝없이 넓어지지 않게 폭의 상한을 정한다.

> **🐾 개집사 앱에서는**
>
> 첫 배포 날 친구들이 알려 준 문제 세 가지 중 하나가 "휴대폰에서 글자가 너무 작다"였다. PC 화면만 보며 만든 탓이었다. 그 뒤로는 고칠 때마다 휴대폰으로 먼저 열어 보는 습관이 생겼다.

AI에게 이렇게 말해 보세요

휴대폰에서 질문 글자가 너무 작다는 의견이 있었어.
질문 문장과 보기 버튼 글자를 휴대폰에서 읽기 편하게 키워 줘.
화면 모양(CSS) 부분만 고치고, 다른 기능은 건드리지 마.
바뀐 부분만 보여 줘.

6장. 데이터 저장과 로직 만들기

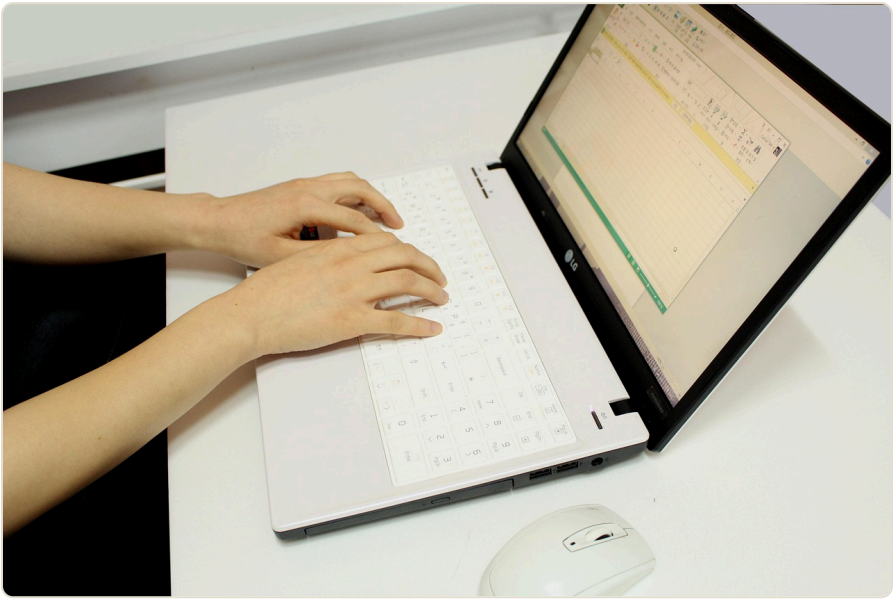


사진: janeb13 / Pixabay

6.1 데이터베이스는 엑셀 표와 비슷하다

용어	뜻	예시
테이블	한 종류의 데이터를 모은 표	사용자, 상품, 리뷰
열(필드)	표의 세로 칸, 데이터의 항목	이름, 가격, 사진, 등록일
행(레코드)	표의 가로 줄, 데이터 한 건	상품 하나

6.2 데이터로 하는 네 가지 일, CRUD

- 만들기(Create): 사용자가 폼을 채우고 제출하면 저장

- 읽기(Read): 저장된 데이터를 목록으로 보여 줌
- 고치기(Update): 작성자가 자기 글을 수정
- 지우기(Delete): 작성자가 자기 글을 삭제

어떤 앱이든 이 네 가지의 조합이다. AI에게 기능을 설명할 때도 "이 데이터를 누가 만들고, 누가 보고, 누가 고치고, 누가 지울 수 있는지"를 함께 말하면 결과가 정확해진다.

6.3 서버 없이 저장하기: 초보자에게 가장 쉬운 길

데이터를 꼭 회사 서버에 모아야 할까? 그렇지 않다. 사용자의 휴대폰 브라우저 안에도 저장 공간이 있다.

방법	쉬운 비유	잘 맞는 것	한계
LocalStorage	브라우저 속 작은 공책	설정값, 마지막 결과, 게임 점수	보통 5MB 안팎, 글자만 저장
IndexedDB	브라우저 속 큰 창고	사진 같은 큰 파일	사용법이 조금 복잡함
서버 + 데이터베이스	회사 창고	여러 기기·여러 사람이 함께 보는 데이터	로그인, 보안, 비용 필요

(출처: MDN 웹 문서, Web Storage API, MDN 웹 문서, IndexedDB)

브라우저 안에 저장하면 로그인이 필요 없다. 그 휴대폰 자체가 "이 사람 것"이기 때문이다. 대신 다른 기기에서는 볼 수 없고, 브라우저 기록을 지우면 사라질 수 있다.

🐾 개집사 앱에서는

"다른 휴대폰에서도 내 작위증 볼 수 있어?"라는 질문을 받았다. 답은 "아니요"였다. 그래도 로그인을 넣지 않기로 했다. 재미로 하는 진단에 회원가입까지 할 사람은 거의 없고, 로그인이 생기면 비밀번호와 이메일 같은 개인정보를 책임져야 하기 때문이다. 그래서 이렇게 나눴다. 작위증 결과와 게임 기록은 LocalStorage, 강아지 사진은 IndexedDB에 저장했다. 사진은 서버로 보내지 않고 사용자 휴대폰 안에서만 처리해서, 남의 반려견 사진을 내가 보관할 일 자체가 없게 만들었다. 기기를 바꾸는 사람을 위해서는 기록을 파일로 내보내고 다시 불러오는 백업 기능을 따로 붙였다.



사용자가 올린 사진이 작위증 액자에 들어간 모습. 사진은 휴대폰 밖으로 나가지 않는다 (출처: 직접 제작한 작위증, 앱에서 발급)

6.4 로직: "만약 ~라면, ~한다"

앱이 알아서 판단하는 부분은 대부분 조건문이다. 말로 쓰면 이렇다.

`결제 버튼을 누른다 → 만약 잔액이 충분하면 → 결제 완료 화면으로 가고 기록을 남긴다 → 아니면 → "잔액이 부족합니다" 안내를 띄운다`

노코드 도구에서는 이것을 블록으로 연결하고, AI에게는 이 문장을 그대로 말하면 된다.

> **🐾 개집사 앱에서는**
> 질문 7개의 보기마다 "어느 집사 유형에 1점을 주는지"를 정해 두고, 끝나면 "점수가 가장 높은 유형"이 결과가 되게 했다. 첫 버전은 보기마다 1~4점을 주고 합계로 나눴는데, 대부분 사람이 중간쯤 답하니 결과가 한쪽에 몰렸다. 게다가 가장 높은 5단계 작위는 아예 나올 수 없게 짜여 있었다. 유형별로 점수를 따로 모으는 방식으로 바꾸자 결과가 골고루 나왔다. 질문 7개 × 보기 4개 = 28개 보기가 유형별로 몇 번씩 나오는지 세어 보는 표도 AI에게 부탁해 만들었다.



가장 높은 점수의 유형이 결과로 나온다 (출처: 왕립 개집사협회 앱 2026년 9월 19일 버전 화면, 직접 캡처)

내 의견: 결과가 나오는 앱이라면 "모든 결과가 실제로 나올 수 있는지"를 꼭 확인하자. 나오지 않는 결과는 아무리 공들여 써도 아무도 보지 못한다.

6.5 내용은 한곳에 모아 두기

질문, 보기, 결과 문구 같은 내용은 화면을 움직이는 코드와 분리해 한곳에 목록으로 모아 달라고 AI에게 부탁하자. 그러면 코드를 몰라도 그 목록만 문서처럼 읽고 고칠 수 있다. 나중에 다른 언어판을 만들 때도 그 목록만 번역하면 된다. 개집사 앱의 영어판과 일본어판도 이렇게 만들었다.

7장. AI에게 일 잘 시키는 법 (프롬프트)



사진: Pexels / Pixabay

7.1 요청서의 네 칸

AI 앱 빌더든 AI 채팅이든, 구체적일수록 결과가 좋아진다. 아래 네 칸을 채우면 된다.

`[역할] + [목적과 기능] + [디자인 느낌] + [조건과 제한]`

나는 여기에 하나를 더 붙인다. [내 수준]. "코딩을 전혀 모르는 초보자"라고 밝히면 AI가 설명과 결과물을 그 눈높이에 맞춘다.

7.2 화면을 만들 때

```
당신은 경험 많은 UX/UI 디자이너입니다.
휴대폰을 먼저 생각한 할 일 관리 웹앱의 첫 화면을 만들어 주세요.
- 위쪽에 오늘 날짜와 남은 할 일 개수
- 가운데 할 일 목록 (완료 체크, 삭제 버튼)
- 아래쪽 고정된 "할 일 추가" 버튼
- 밝고 깔끔한 느낌, 대표 색은 하늘색 하나
- 코딩을 모르는 초보자라서 파일 하나(index.html)로 만들어 주세요.
```

7.3 기능을 만들 때

```
할 일을 입력하고 엔터를 누르면 목록 맨 위에 추가되게 해 줘.
각 항목에는 완료 체크박스와 삭제 버튼이 있고,
체크하면 글자에 줄이 그어지게 해 줘.
앱을 닫았다 열어도 목록이 남아 있게 브라우저에 저장해 줘.
다른 기능은 건드리지 말고, 바뀐 부분만 알려 줘.
```

🐾 개집사 앱에서는 이렇게 보냈다 (첫 요청서)

반려인을 위한 집사 작위증 웹앱을 만들고 싶어. [세계관] 강아지는 왕(폐하), 반려인은 왕을 모시는 집사. 왕립 개집사협회가 집사에게 작위를 내린다. [앱 설명] 폐하의 이름·성격·좋아하는 활동·싫어하는 활동·왕국 분위기를 입력하고, 집사가 질문 7개에 답하면 집사 유형과 작위, 왕국 이름·표어가 담긴 작위증을 보여 주는 웹앱. [화면 순서] 시작 화면 → 왕국 만들기 → 질문 7개 → 결과 → 작위증 [작위] 견국 수습집사, 왕실 생활협력관, 왕실 돌봄근위관, 발바닥 궁정대신, 왕실 총집사 (5단계) [조건] 코딩을 모르는 초보자라서 파일 하나(index.html)로. 휴대폰에서도 보기 좋게. 사진 업로드와 공유는 아직 넣지 마.

처음의 한 문장 요청과 결과가 완전히 달랐다. 마지막 줄 "아직 넣지 마"가 특히 중요했다. 하지 말아야 할 것을 말해 두지 않으면 AI는 친절하게 이것저것 덧붙인다.

7.4 효과가 컸던 습관 다섯 가지

1. 한 번에 하나만 부탁한다. "디자인 바꾸고, 질문 추가하고, 공유 버튼 넣어 줘"를 한꺼번에 시키면 무엇 때문에 망가졌는지 알 수 없다.
2. 범위를 정한다. "이 부분만 고치고 다른 기능은 건드리지 마."
3. 오류는 통째로 보낸다. 오류 메시지 전체 + 바로 전에 한 일 + 기대한 결과.
4. 방법이 여러 개면 "초보자에게 가장 쉬운 방법 하나만"을 고르게 한다.
5. AI의 "수정했습니다"를 그대로 믿지 않는다. 직접 열고, 누르고, 새로고침하고, 휴대폰으로도 확인한다. AI가 고쳤다고 해도 실제로는 그대로인 경우가 꽤 있었다.

7.5 기획은 코드와 따로 대화하라

코드를 요청하기 전에 기획만 따로 대화하는 것도 좋은 방법이다. 같은 질문을 서로 다른 AI 두 곳에 해서 겹치는 답과 참신한 답을 비교하면 생각이 넓어진다. 개집사 앱의 질문 7개도 두 AI에게서 후보 20개쯤을 받아 골랐다.

8장. 로그인·결제·도메인, 외부 서비스 연결



사진: thedigitalway / Pixabay

8.1 먼저 물어볼 것: 정말 로그인이 필요한가

로그인은 서버에게 "이 데이터는 이 사람 것"이라고 알려 주는 이름표다. 여러 기기에서 같은 데이터를 봐야 하거나, 사용자끼리 데이터를 주고받는 앱이라면 꼭 필요하다. 그렇지 않다면 6장의 브라우저 저장으로 충분할 수 있다.

로그인이 필요하다면 직접 만들지 말고 검증된 서비스를 쓰자.

- Supabase, Firebase: 이메일 로그인과 구글·카카오 같은 소셜 로그인을 비교적 쉽게 붙일 수 있다. (출처: Supabase 문서, Auth, Firebase 문서, Authentication)

- 국내 사용자 대상이라면 카카오·네이버 로그인을 함께 제공하면 가입 부담이 줄어든다. (출처: Kakao Developers, 카카오 로그인)

8.2 결제

- 국내 사용자 대상: 토스페이먼츠 같은 국내 결제 대행사. (출처: 토스페이먼츠 개발자센터)
- 해외 사용자 대상: Stripe가 널리 쓰인다. 다만 Stripe는 가입할 수 있는 사업자 나라가 정해져 있어, 시작 전에 지원 국가 목록을 꼭 확인해야 한다. (출처: Stripe, 지원 국가)

결제를 붙이려면 대부분 사업자 등록, 이용약관, 환불 정책이 필요하다. 첫 앱에서는 결제를 빼고, 사람들이 실제로 쓰는지 확인한 다음 붙이는 것을 권한다.

8.3 도메인과 검색 노출

- 도메인: 가비아, 고대디 같은 곳에서 .com, .kr 주소를 산다. Vercel 같은 배포 서비스는 산 도메인을 연결하는 방법을 안내해 준다. (출처: Vercel 문서, Domains)
- 검색 노출(SEO): 페이지마다 제목과 설명을 정확히 적는다. 국내라면 구글과 함께 네이버 검색도 함께 챙기자. 2026년 6월 기준 국내 검색 점유율은 구글 약 46%, 네이버 약 44%로 비슷했다. (출처: StatCounter, 한국 검색엔진 점유율)
- 공유 미리보기: 카카오톡에 링크를 보낼 때 뜨는 그림과 제목도 따로 정할 수 있다. 이것만 잘 만들어도 클릭이 달라진다.

🐾 개집사 앱에서는

로그인과 결제는 넣지 않았다. 대신 사람들이 결과를 친구에게 보내고 싶게 만드는 데 힘을 썼다. 작위증을 이미지로 저장하는 기능, 카카오톡으로 공유하는 버튼, 공유 링크를 받은 친구가 자기 결과와 비교하는 궁합 보기까지. 작위증 저장은 아이폰에서만 안 되는 문제가 있어서 한참 헤맸는데, 아이폰 브라우저는 저장 방식이 달라서 따로 처리해야 했다. 이렇게 기기마다 다른 문제는 9장의 테스트에서만 잡힌다.

9장. 테스트, 오류 해결, 배포

9.1 배포는 생각보다 쉽다, 그리고 무섭지 않다

배포는 내 컴퓨터의 파일을 인터넷에 올려 주소를 받는 일이다. Vercel, Netlify 같은 서비스는 GitHub에 파일을 올리면 자동으로 새 버전을 배포해 준다. (출처: Vercel 문서, Deployments, Vercel 문서, Git 연결) 노코드 도구라면 대개 "Publish" 버튼 하나로 끝난다.

꼭 알아야 할 원칙 하나: 인터넷에 있는 앱을 직접 고치는 게 아니다. 내 컴퓨터의 파일을 고친 뒤 다시 올리는 것이다.

> 🐾 **개집사 앱에서는**
> 처음에는 Vercel 화면 어딘가에서 글자를 고칠 수 있는 줄 알았다. Vercel은 파일을 보여 주는 곳이지 고치는 곳이 아니었다. 고치는 곳은 언제나 내 컴퓨터의 index.html이었다. 첫 실수는 더 앞에 있었다. 메모장에 코드를 붙여 넣고 저장했는데 파일 이름이 index.html.txt가 됐다. 윈도우 탐색기에서 "파일 확장명" 보기를 켜고 나서야 알았다. 초보자라면 이 설정부터 켜 두자.

9.2 사용자 눈으로 하는 점검 목록

- ☐ 휴대폰, 태블릿, PC 화면에서 모양이 깨지지 않는가?
- ☐ 아이폰(사파리)과 안드로이드(크롬, 삼성 인터넷)에서 모두 열어 봤는가?
- ☐ 카카오톡 안에서 링크를 열어도 제대로 동작하는가?
- ☐ 처음 쓰는 사람이 설명 없이 끝까지 갈 수 있는가?
- ☐ 뒤로 가기를 눌러도 이상하게 처음으로 돌아가지 않는가?
- ☐ 저장한 데이터가 앱을 닫았다 열어도 남아 있는가?
- ☐ 오타나 어색한 문장은 없는가?

- ☐ 하나를 고친 뒤, 원래 잘 되던 다른 기능도 여전히 되는가?

마지막 항목이 특히 중요하다. 개발에서는 이것을 회귀 테스트라고 부른다. 하나를 고쳤더니 다른 것이 깨지는 일은 초보자에게 정말 자주 일어난다.

> 🐾 **개집사 앱에서는**
> 첫 배포 날, 링크를 받은 친구들이 세 가지를 알려 줬다. 휴대폰에서 글자가 작다, 뒤로 가기를 누르면 처음으로 돌아간다, 그림이 깨진다. 혼자서는 하나도 못 찾은 문제였다. 내 의견: 첫 배포는 완벽하지 않아도 된다. 믿을 만한 친구 몇 명에게 먼저 보내는 것이 가장 좋은 테스트다.

9.3 오류 앞에서 침착하게: 두 종류만 기억하기

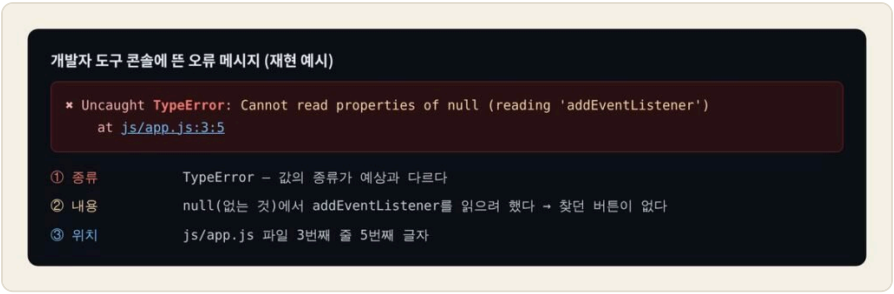
종류	보이는 모습	급한 정도
배포가 실패한 오류	배포 서비스에 빨간 Error 표시	낮음. 이전 버전이 그대로 서비스된다
배포는 됐는데 앱이 이상한 오류	화면이 하얗거나 버튼이 안 눌림	높음. 사용자가 바로 망가진 앱을 본다

Vercel은 새 배포가 실패하면 이전에 성공한 버전을 그대로 유지한다. 그리고 잘못 올린 버전은 이전 배포로 몇 초 만에 되돌릴 수 있다. (출처: Vercel 문서, Instant Rollback)

오류를 해결하는 순서:

1. 오류 기록은 맨 아래부터 위로 읽는다. Error, Failed, Invalid 같은 단어가 있는 줄을 찾는다.
2. 그 줄과 위아래 몇 줄을 복사해 AI에게 보낸다. 바로 전에 무엇을 바꿨는지도 함께.

3. 화면이 하얗게 나오면 브라우저의 개발자 도구(F12) 콘솔에 뜬 빨간 글을 그대로 보낸다.
4. 급하면 먼저 이전 버전으로 되돌리고, 차분히 원인을 고친다.



오류 메시지는 종류, 내용, 위치 세 부분으로 나눠 읽는다 (출처: 브라우저에서 같은 오류를 재현해 직접 제작)

🐾 개집사 앱에서는

사례 1. 주소 설정 파일(vercel.json)을 하나 추가했다가 배포가 실패했다. 빨간 Error에 심장이 쿵 했지만, 휴대폰으로 열어 보니 앱은 멀쩡했다. 오류 기록의 빨간 줄을 AI에게 보여 주고 30분 만에 해결했다. 사례 2. 결과 설명 문구를 고치고 올렸더니 결과 화면이 통째로 하얗게 나왔다. 따옴표 하나가 빠져서 JavaScript 전체가 멈춘 것이었다. Vercel에서 이전 배포로 되돌려 급한 불을 끄고, 원인을 고친 뒤 새로 올렸다. 이때 주의할 점: 배포 서비스에서 되돌려도 내 컴퓨터 파일은 여전히 잘못된 상태다. 파일도 꼭 고쳐야 한다.

9.4 자주 만나는 문제와 해결

문제	흔한 원인	해결
다른 사용자의 정보가 보인다	데이터베이스 접근 권한 설정 누락	행 단위 접근 규칙(RLS) 확인 (출처: Supabase 문서, Row Level Security)
화면이 느리게 뜬다	이미지 용량이 큼	WebP 같은 가벼운 형식으로 바꾸고 크기를 줄인다 (출처: Google, WebP)

문제	흔한 원인	해결
아이폰에서만 안 된다	사파리의 동작 방식 차이	아이폰 실제 기기로 확인하고, 증상을 "아이폰 사파리에서"라고 밝혀 AI에게 묻기
고쳤는데 그대로다	브라우저에 남은 옛 파일(캐시)	강력 새로고침, 시크릿 창에서 확인
하나 고쳤더니 다른 게 깨졌다	범위를 정하지 않은 수정	9.2의 점검 목록을 매번 돌리고, 한 번에 하나만 고친다

9.5 수정 루틴 5단계

배포한 앱을 고칠 때 내가 정착한 순서다.

1. 오늘 고칠 것 하나를 정한다.
2. AI에게 범위를 정해 요청한다("이 부분만, 바뀐 곳만 보여 줘").
3. 내 컴퓨터에서 먼저 열어 확인한다.
4. 휴대폰으로도 확인한다.
5. 무엇을 바꿨는지 한 줄로 적어 올린다(GitHub의 커밋 메시지).

9.6 휴대폰 앱처럼: PWA와 스토어 등록

- PWA: 웹앱에 아이콘과 설정 파일을 더하면, 사용자가 홈 화면에 앱처럼 설치할 수 있다. 주소창 없이 전체 화면으로 열린다.
- Google Play 등록: PWA를 포장(TWA)해서 올릴 수 있다. 단, 2023년 11월 13일 이후 만든 개인 개발자 계정은 테스터 12명 이상이 14일 연속 참여하는 비공개 테스트를 거쳐야 정식 출시를 신청할 수 있다. (출처: Google Play Console 고객센터, 신규 개인 개발자 계정의 앱 테스트 요건) 테스터가 중간에 빠지면 기간을 다시 채워야 할 수 있으니 12명보다 넉넉히 모으자.



PWA로 만들고 첫 화면에 넣은 앱 설치 카드 (출처: 왕립 개집사협회 앱 2026년 9월 22일 버전 화면, 직접 캡처)

10장. 사람들에게 알리고 수익 내기



사진: nattanan23 / Pixabay

10.1 수익 모델 네 가지

모델	방식	잘 맞는 앱	주의할 점
구독	매달 일정 금액	매일 쓰는 도구, 업무용 앱	계속 쓸 이유를 꾸준히 줘야 함
단건 결제	기능·콘텐츠를 한 번 구매	결과 리포트, 고급 테마	무료 부분이 먼저 충분히 좋아야 함
광고	화면에 광고를 넣음	방문이 많은 무료 앱	방문자가 많아야 의미가 생김, 사용 경험을 해치지 않게

모델	방식	잘 맞는 앱	주의할 점
중개	거래를 연결하고 수수료	예약·거래 플랫폼	공급자와 사용자를 동시에 모아야 함

내 의견: 첫 앱으로 돈을 벌겠다는 목표보다, 사람들이 실제로 쓰고 다시 오는지를 먼저 확인하자. 쓰는 사람이 생기면 수익 모델은 그다음에 고를 수 있다. 앱 자체로 돈을 받지 않아도, 앱을 만든 과정을 블로그나 영상으로 기록하는 것 자체가 또 하나의 자산이 된다. 지금 이 책도 그렇게 나왔다.

10.2 첫 사용자 모으기

- 가까운 사람부터: 첫 배포 링크는 친구와 가족에게. 가장 솔직한 의견을 준다.
- 커뮤니티: 내 앱이 도움이 될 사람들이 모인 카페, 모임, 오픈채팅. 광고 글보다 "이런 걸 만들어 봤는데 써 봐 주실래요?"가 반응이 좋다. 각 커뮤니티의 홍보 규칙은 꼭 확인하자.
- 공유하고 싶게 만들기: 결과 이미지, 카카오톡 미리보기 그림, 친구와 비교하는 기능. 사용자가 대신 홍보해 준다.
- 기록하기: 블로그, 유튜브, 인스타그램에 만드는 과정과 쓰는 팁을 올린다. 검색으로 들어온 사람이 앱까지 온다.

> **🐾 개집사 앱에서는**
> 홍보의 중심은 "자랑하고 싶은 결과"였다. 작위증을 이미지로 저장하는 기능, 카카오톡 공유 버튼, 받은 친구가 자기 왕국과 궁합을 보는 기능을 차례로 붙였다. 그리고 만드는 과정 전부를 와우굿앱 블로그에 연재했다. 매일 들어올 이유도 만들었다. 왕실 아카데미, 미니게임, 웹툰 연재, 출석부 같은 장치다.

10.3 피드백을 받고 계속 고치기

- 의견 받는 창구를 만든다: 구글 폼, 채널톡, 앱 안의 문의 버튼.

- 내가 먼저 매일 써 본다. 만든 사람이 매일 쓰면 불편한 곳이 저절로 보인다.
- 고친 내용을 짧게 알린다. "말씀해 주신 기능 넣었어요"라는 한마디가 팬을 만든다.

> **🐾 개집사 앱에서는**
>
> 앱을 직접 매일 쓰다 보니 홈 화면의 버튼 순서, 휴대폰에서 누르기 불편한 방향키, 글자 대비 같은 것들이 계속 눈에 들어왔다. 첫 버전 이후의 개선 대부분은 이렇게 "직접 써 보다 불편해서" 시작됐다.

11장. 맺음말

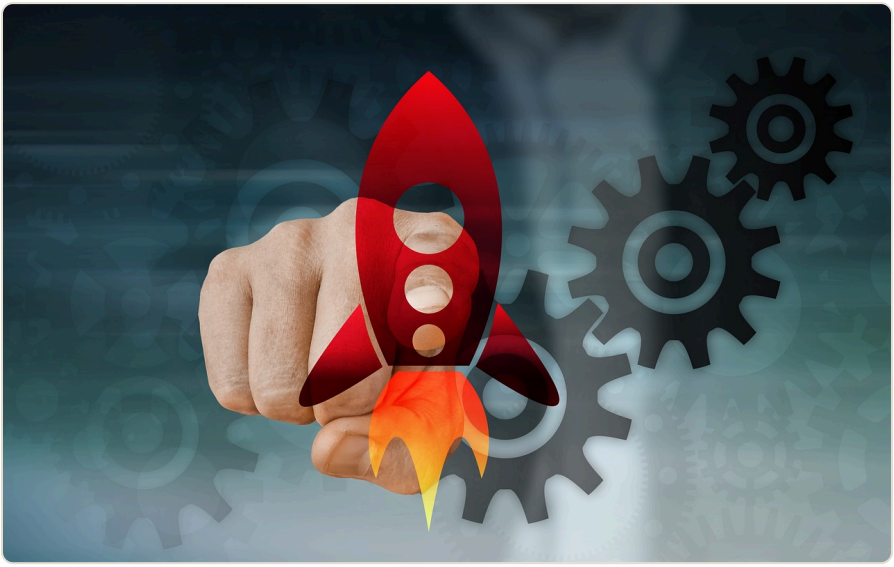


사진: geralt / Pixabay

웹앱을 만드는 일은 더 이상 거창한 기술 프로젝트가 아니다. 코딩 한 줄 몰랐던 사람이 AI와 함께 일주일 만에 첫 앱을 인터넷에 올렸고, 그 앱은 지금도 자라고 있다.

돌아보면 중요한 것은 세 가지였다.

1. 작게 정의하기. 한 문장, 핵심 기능 몇 개.
2. 빨리 보여 주기. 완벽하지 않아도 친구 몇 명에게 먼저.
3. 확인하고 고치기. AI는 만들고, 나는 검사한다. 한 번에 하나씩.

이 가이드를 발판 삼아 지금 바로 첫 번째 웹앱을 시작해 보자. 막히면 그 자리에서 AI에게 물어보면 된다. 막힌 곳이 곧 배우는 곳이다.



왕립 개집사협회 앱 아이콘 (출처: 직접 제작)

부록 A. 초보자 용어 사전

용어	쉬운 설명
HTML	화면의 뼈대(제목, 글, 버튼이 무엇인지)
CSS	화면의 옷(색, 크기, 배치)
JavaScript	화면의 움직임(누르면 무슨 일이 일어나는지)
index.html	웹사이트에 들어가면 가장 먼저 열리는 기본 파일
GitHub	내 앱 파일을 보관하고 바뀐 기록을 남기는 곳. 앱의 타임머신
커밋	지금 상태를 기록으로 남기기(사진 찍어 두기)
푸시	내 컴퓨터의 기록을 GitHub로 올리기
배포	파일을 인터넷에 올려 누구나 주소로 열 수 있게 하기
Vercel	GitHub와 연결해 자동으로 배포해 주는 서비스
도메인	인터넷 주소(예: wowgoodapp.com)
LocalStorage	브라우저 속 작은 공책. 앱을 닫아도 남는 작은 기록
IndexedDB	브라우저 속 큰 창고. 사진 같은 큰 파일 보관
PWA	웹앱을 휴대폰 홈 화면에 앱처럼 설치하게 하는 기술
TWA	PWA를 Google Play에 올릴 수 있게 포장하는 방식
MVP	반응을 보려고 핵심 기능만 담아 만든 첫 버전
반응형	화면 크기에 따라 모양이 알맞게 바뀌는 것
캐시	빨리 열려고 브라우저가 저장해 둔 옛 파일

용어	쉬운 설명
콘솔	브라우저 개발자 도구(F12) 안의 오류 알림판
롤백	이전 배포 버전으로 되돌리기

부록 B. 바로 쓰는 프롬프트 모음

기획

```
이 아이디어로 웹앱을 만들고 싶어: (한 문장)
사용자 흐름을 짜 주고, 빠진 화면이 있으면 알려 줘.
처음 버전에 꼭 필요한 기능과 나중에 해도 되는 기능을 나눠 줘. 코딩은 아직 하지 마.
```

첫 제작

```
코딩을 전혀 모르는 초보자야. (요청서: 세계관/설명/화면 순서/조건)
파일 하나(index.html)로, 휴대폰에서도 보기 좋게 만들어 줘.
(아직 넣지 말 기능)은 넣지 마.
```

파일 열기

```
이 코드를 받았는데, 내 컴퓨터(윈도우)에서 어떻게 열어 봐? 순서대로 알려 줘.
```

수정

```
(문제 상황)이 있어. (고칠 부분)만 고치고 다른 기능은 건드리지 마.
바뀐 부분만 보여 줘. 자신 없으면 전체 코드를 다시 줘.
```

오류

```
(바로 전에 한 일)을 했더니 이런 오류가 나왔어. (오류 메시지 전체)
원인과, 초보자가 따라 할 수 있는 해결 방법 하나만 알려 줘.
```

개념 이해

(용어)가 뭔지 코딩을 전혀 모르는 사람에게 비유로 설명해 줘. 세 문장 안으로.

점검

지금 설계에서 (결과 유형)이 각각 몇 번씩 나올 수 있는지 세어 줘.
한 번도 나올 수 없는 결과가 있으면 알려 줘.

부록 C. 첫 웹앱 전체 점검 목록

기획 - ☐ 앱을 한 문장으로 설명할 수 있다 - ☐ 사용자 흐름을 적었다 - ☐ 첫 버전 기능을 3~5개로 줄였다

만들기 - ☐ 휴대폰 화면 기준으로 만들었다 - ☐ 첫 화면에서 3초 안에 무슨 앱인지 알 수 있다 - ☐ 모든 결과(화면)가 실제로 나올 수 있다 - ☐ 내용(문구)이 한곳에 모여 있다

공개 전 - ☐ 아이폰·안드로이드·PC에서 열어 봤다 - ☐ 카카오톡 안에서 열어 봤다 - ☐ 개인정보를 모은다면 처리방침을 마련했다 - ☐ 친구 몇 명에게 먼저 보내 의견을 받았다

공개 후 - ☐ 고칠 때마다 한 번에 하나씩 - ☐ 고친 뒤 다른 기능도 확인 - ☐ 의견 받는 창구가 있다 - ☐ 만드는 과정을 기록하고 있다

부록 D. 출처

- MDN 웹 문서, Progressive web apps
- MDN 웹 문서, Web Storage API
- MDN 웹 문서, IndexedDB API
- MDN 웹 문서, Flexbox
- Wikipedia, Minimum viable product
- StatCounter, 한국 기기별 점유율 (2026년 9월)
- StatCounter, 한국 검색엔진 점유율 (2026년 6월)
- v0 공식 사이트
- Bolt 공식 사이트
- Supabase 문서, Auth
- Supabase 문서, Row Level Security
- Firebase 문서, Authentication
- Kakao Developers, 카카오 로그인
- 토스페이먼츠 개발자센터
- Stripe, 지원 국가
- Vercel 문서, Deployments
- Vercel 문서, Git 연결
- Vercel 문서, Domains
- Vercel 문서, Instant Rollback
- Google, WebP
- Google Play Console 고객센터, 신규 개인 개발자 계정의 앱 테스트 요건
- Pixabay 콘텐츠 라이선스

사진 안내: 일반 사진은 모두 Pixabay 라이선스에 따라 무료로 사용했으며, 출처 표기는 의무가 아니지만 각 사진 아래에 작가를 밝혔다. 앱 화면과 작위증, 아이콘은 직접 캡처하거나 제작한 것이다.