

## 바이브 코딩을 시작하기 전에 — 개발 환경 준비

컴퓨터에 4가지 도구를 설치하면 바이브 코딩을 할 준비가 끝난다

바이브 코딩은 AI에게 말로 지시해서 프로그램을 만드는 것이다. 그런데 AI에게 말을 걸려면, 먼저 AI가 일할 수 있는 환경을 컴퓨터에 만들어줘야 한다. 이 과정은 딱 한 번만 하면 된다. 한 번 설치하면 앞으로 계속 쓸 수 있다.

설치할 도구는 4가지다:

순 서	도구	한마디 설명
1	VS Code	AI가 만든 코드를 보고, 결과물을 미리보기 하는 편집기
2	Node.js	Claude Code를 실행하기 위한 기반 프로그램 (직접 쓸 일 없음)
3	Git	작업 내용을 저장하고 되돌릴 수 있게 해주는 도구
4	Claude Code	AI 코딩 파트너. 말하면 코드를 만들어준다

### 0단계. 폴더 만들고 터미널 열기

모든 것의 시작은 빈 폴더 하나를 만드는 것이다.

#### 0-1. 바이브코딩 폴더 만들기

1. 바탕화면이든, 문서 폴더든, 원하는 위치를 연다
2. 마우스 오른쪽 버튼 → 새로 만들기 → 폴더를 클릭한다
3. 폴더 이름을 **바이브코딩**이라고 입력한다

(폴더 생성 스크린샷 삽입)

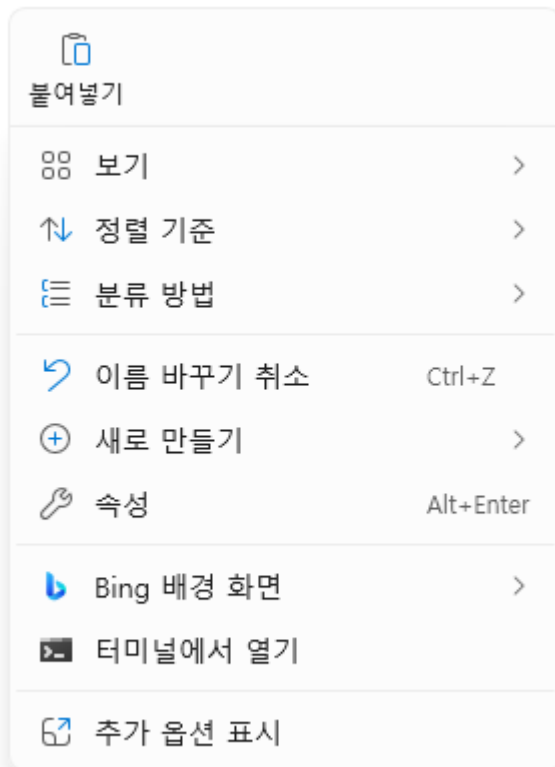
이 폴더가 앞으로 우리가 바이브 코딩으로 만드는 모든 프로젝트의 집이 된다.

#### 0-2. 터미널 열기

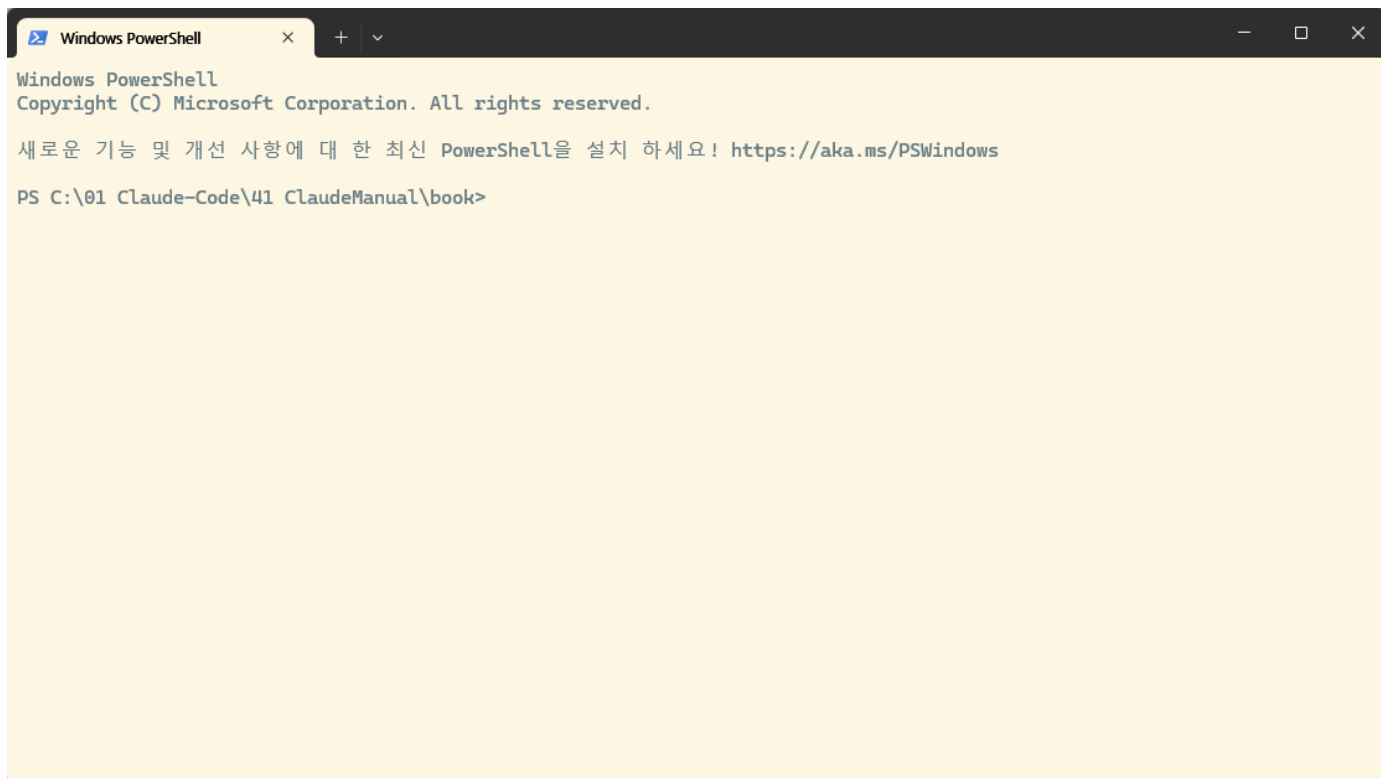
방금 만든 **바이브코딩** 폴더를 열어서 들어간다. 폴더 안은 비어 있다. 여기서:

1. 폴더 안의 빈 공간에서 마우스 오른쪽 버튼을 클릭한다

2. 메뉴에서 "터미널에서 열기"를 클릭한다



검은색(또는 파란색) 창이 팝업으로 뜬다. 이것이 **터미널**이다.



**터미널이란?** 컴퓨터에게 글자로 명령을 내리는 창이다. 평소에는 마우스 클릭으로 컴퓨터를 조작하지만, 터미널에서는 글자를 입력해서 조작한다. 무서워할 필요 없다. 잘못 입력해도 컴퓨터가 망가지지 않는다. 그냥 "명령을 인식할 수 없습니다"라고 나올 뿐이다.

터미널이 열리면 아래와 비슷한 화면이 보인다:

```
C:\Users\사용자이름\바탕화면\바이트코딩> _
```

깜빡이는 커서가 보이면 준비 완료다. 이제 여기에 설치 명령어를 하나씩 입력하면 된다.

### 1단계. VS Code 설치

VS Code는 **코드 편집기**다. 메모장의 업그레이드 버전이라고 생각하면 된다. AI가 만들어주는 코드를 확인하고, 결과물을 미리보기 하는 데 사용한다.

터미널에 아래 명령어를 복사해서 붙여넣고, Enter 키를 누른다:

```
winget install Microsoft.VisualStudioCode
```

설치가 진행되면서 글자가 쭉 올라간다. **가만히 기다리면 된다.** 설치가 끝나면 다시 커서가 깜빡인다.

**winget이 뭐야?** Windows에 내장된 프로그램 설치 도구다. 앱스토어에서 앱을 설치하는 것과 같다. 다만 마우스 클릭 대신 글자로 설치할 프로그램 이름을 적는 것뿐이다.

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

새로운 기능 및 개선 사항에 대 한 최신 PowerShell을 설치 하세요! https://aka.ms/PSWindows

PS C:\01 Claude-Code\41 ClaudeManual\book> winget install Microsoft.VisualStudioCode
Found an existing package already installed. Trying to upgrade the installed package...
Found Microsoft Visual Studio Code [Microsoft.VisualStudioCode] Version 1.109.3
This application is licensed to you by its owner.
Microsoft is not responsible for, nor does it grant any licenses to, third-party packages.
Downloading https://vscode.download.prss.microsoft.com/dbazure/download/stable/b6a47e94e326b5c209d118cf0f994d6065585705/VSCodeUserSetup-x64-1.109.3.exe
120 MB / 120 MB
Successfully verified installer hash
Starting package install...
You cancelled the installation.
Installer failed with exit code: 5
Installer log is available at: C:\Users\USER\AppData\Local\Packages\Microsoft.DesktopAppInstaller_8wekyb3d8bbwe\LocalState\DiagOutputDir\Microsoft.VisualStudioCode.1.109.3-26-02-16-01-13-46.log
PS C:\01 Claude-Code\41 ClaudeManual\book> |
```

설치가 완료되면 시작 메뉴에서 "Visual Studio Code"를 찾을 수 있다.

## 2단계. Node.js 설치

Node.js는 Claude Code를 실행하기 위한 **기반 프로그램**이다. 직접 사용할 일은 없다. Claude Code가 이것 위에서 돌아가기 때문에 설치해야 한다. 한마디로 "Claude Code의 연료"라고 생각하면 된다.

같은 터미널에서 아래 명령어를 입력하고 Enter:

```
winget install OpenJS.NodeJS.LTS
```

역시 설치가 진행된다. 기다리면 된다.

설치가 끝나면 제대로 설치됐는지 확인해본다:

```
node -v
```

아래처럼 버전 번호가 뜨면 성공이다:

v22.13.1

"node를 찾을 수 없습니다"라고 나온다면? 터미널을 닫고, 다시 바이브코딩 폴더에서 마우스 오른쪽 버튼 → "터미널에서 열기"를 해본다. 설치 직후에는 터미널을 새로 열어야 인식되는 경우가 있다.

### 3단계. Git 설치

Git은 작업 내용을 저장하고 되돌릴 수 있게 해주는 도구다. 워드 문서에서 Ctrl+Z로 이전 상태로 되돌리는 것처럼, Git은 프로젝트 전체를 과거 시점으로 되돌릴 수 있다. 실수해도 괜찮다는 안전장치다.

같은 터미널에서:

```
winget install Git.Git
```

설치 완료 후 확인:

```
git --version
```

아래처럼 버전이 뜨면 성공이다:

```
git version 2.47.1.windows.2
```

버전이 안 뜨면? Node.js 때와 마찬가지로 터미널을 닫고 다시 열어본다.

### 4단계. Claude Code 설치

드디어 핵심이다. **Claude Code가 바로 우리의 AI 코딩 파트너다.** 터미널에서 대화하듯이 말하면, AI가 코드를 직접 작성해준다.

앞에서 설치한 Node.js에 포함된 npm이라는 도구로 설치한다:

```
npm install -g @anthropic-ai/claude-code
```

이 명령어는 다른 것보다 시간이 좀 더 걸릴 수 있다. 1~2분 정도 기다린다.

설치가 끝나면 제대로 됐는지 확인한다:

```
claude --version
```

버전 번호가 뜨면 설치 완료다.

**npm이 뭐야?** Node.js를 설치하면 자동으로 따라오는 프로그램 설치 도구다. winget이 Windows 프로그램을 설치하는 도구라면, npm은 개발 도구를 설치하는 도구다. 이름만 다를 뿐 하는 일은 같다.

### 환경 준비 완료!

4가지 도구를 모두 설치했다. 정리하면:

순서	입력한 명령어	결과
1	winget install Microsoft.VisualStudioCode	코드 편집기 설치됨
2	winget install OpenJS.NodeJS.LTS	Claude Code 기반 설치됨
3	winget install Git.Git	버전 관리 도구 설치됨
4	npm install -g @anthropic-ai/claude-code	AI 코딩 파트너 설치됨

터미널에서 명령어 4줄을 입력한 것이 전부다. 이제 바이브 코딩을 본격적으로 시작할 수 있다.

**이 설치의 딱 한 번만 하면 된다.** 다음부터는 터미널을 열고 바로 claude를 입력하면 AI와 대화를 시작할 수 있다.

## 챕터 1. 바이브 코딩으로 2시간 만에 매뉴얼 사이트 만들기

### 이번에 배울 것은 실습이다

이론부터 읽고, 명령어를 외우고, 개념을 이해한 다음에 시작하려고 하면 지루하고, 효과도 낮다. 그래서 이 책은 바로 실습부터 시작한다. 바이브 코딩의 방법과 맛을 직접 경험하는 것이 가장 빠른 길이다. 필자 역시 실습을 통해 가장 많이 배우고, 가장 깊이 이해했다.

이번 실습에서 만들 것은 **바이브코딩 완벽 매뉴얼 홈페이지**다. 필자가 바이브 코딩을 배운 지 **1주일** 만에, **2시간** 동안, **혼자 힘으로** 완성한 결과물이다.

그날의 기억이 선명하다. 조용한 카페에서 커피 한 잔을 앞에 두고 노트북을 열었다. AI에게 말로 지시하고, 결과를 확인하고, 또 지시하고. 그렇게 2시간이 흘렀을 때, 화면에 완성된 웹사이트가 떠 있었다. 자신도 모르게 두 주먹을 하늘로 올리며 "야호!"를 외쳤다. 카페에 있던 사람들이 쳐다봤지만, 신경 쓸 겨를이 없었다.

**디지털 노마드**라는 말을 들어봤을 것이다. 노트북 하나만 들고 카페로, 해외로, 휴양지로 이동하면서 일하는 사람

들. 그동안 이 단어는 코딩을 전문으로 하는 개발자들의 전유물이었다. 코딩을 할 줄 알아야 어디서든 일할 수 있었으니까. 필자도 주변에서 그런 사람들을 볼 때마다 부러웠다. "나는 코딩을 모르니까, 저건 나와는 다른 세계의 이야기구나."

그런데 바이브 코딩을 시작하고 나서, **2주 만에** 필자도 그렇게 되어 있었다. 카페에서, 도서관에서, 이동 중에, 노트북을 열고 AI에게 말을 걸며 웹사이트를 만들고 있었다.

필자가 바이브 코딩을 처음 시작한 날은 **2026년 1월 1일**이다. 새해 첫날, 새로운 것을 시작해보자는 마음이었다. 처음에는 어색했다. 컴퓨터에게 "이렇게 만들어줘", "저걸 바꿔줘"라고 말로 명령하는 것이 낯설었다. 타이핑도 아니고, 클릭도 아니고, **말로** 지시한다니. 이상한 느낌이었다.

하지만 명령을 내리고 잠시 후, 모니터에 웹사이트가 나타나는 순간 — 이상하다는 생각은 사라지고 **흥분**이 차올랐다. 목소리 톤이 높아졌다. "이번엔 배경색을 바꿔!" "카드를 추가해!" 점점 자신감이 붙으면서 명령이 강해졌다. AI가 만들어내는 결과물을 보면서, "**이게 된다고?**"라는 놀라움이 "**이것도 되겠는데?**"라는 도전으로 바뀌었다.

그 결과:

- **1월 한 달** — 30개의 웹사이트 개발
- **2월 15일 기준** — 총 45개 개발 완료
- **평균 하루에 1개씩**, 코딩을 전혀 모르는 사람이 만든 숫자다

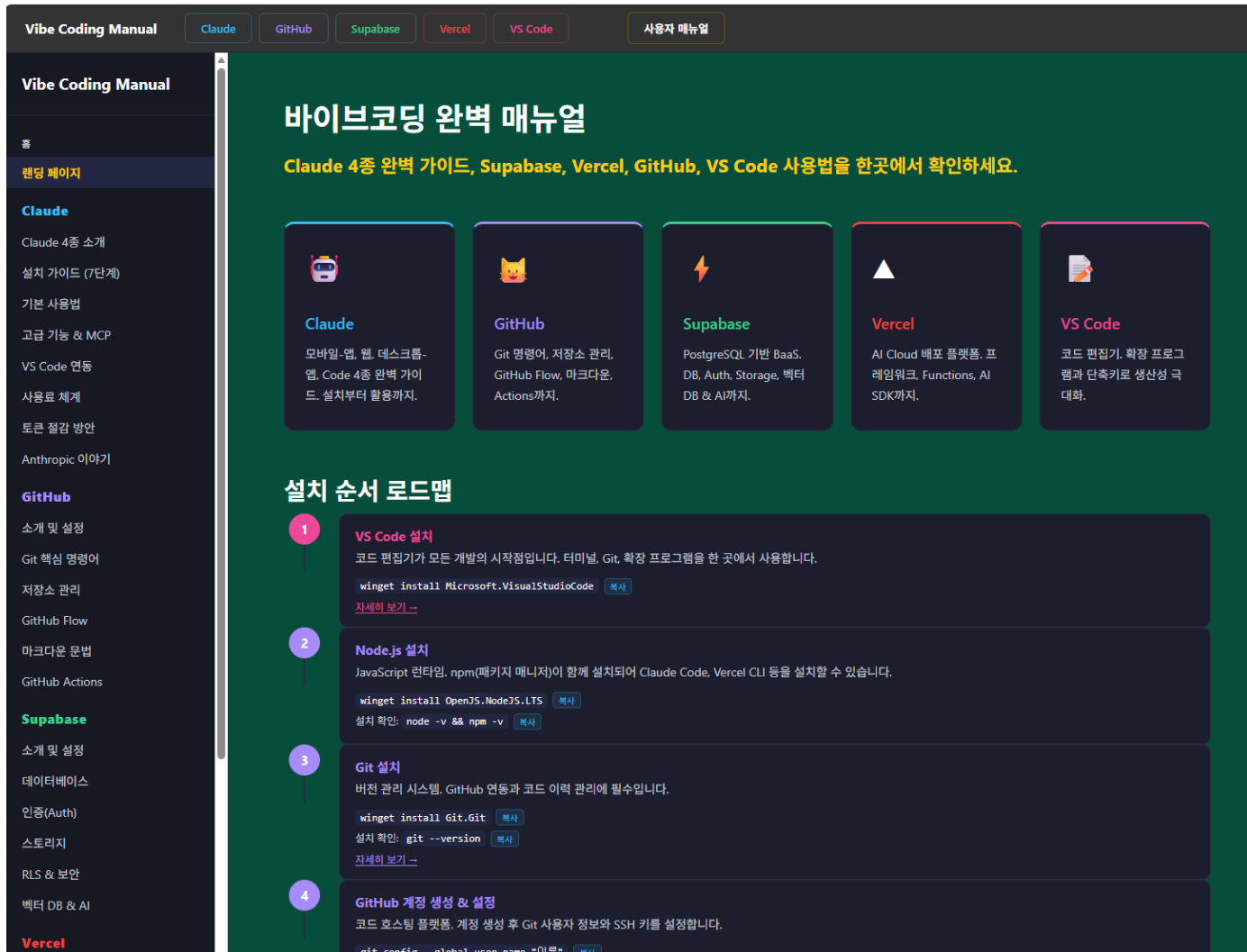
1주일 전까지 코드 한 줄 쓸 줄 몰랐던 사람이, 한 달 만에 30개의 웹사이트를 만들었다. 이것이 바이브 코딩의 힘이다. 그리고 이 책을 읽는 당신도, 이 챕터를 따라하고 나면 똑같은 경험을 하게 될 것이다.

## 이 챕터에서 배울 것

1. **바이브 코딩에 필요한 도구는 4개다** — VS Code, Node.js, Git, Claude Code. 이것만 설치하면 준비 끝이다.
2. **한 번에 하나씩 요청하라** — "탭도 만들고, 색상도 바꾸고, 푸터도 추가해줘"처럼 한꺼번에 요청하면 원하는 결과가 나오기 어렵다. 하나를 요청하고, 결과를 확인하고, 다음을 요청한다.
3. **큰 그림부터, 세부 사항은 나중에** — 뼈대 → 콘텐츠 → UI 순서로 진행한다. 처음부터 색상이나 글꼴을 고민하지 않는다.
4. **느낀 대로 말하면 된다** — "색상을 바꿔줘", "간격을 넓혀줘", "글씨가 잘 안 보여" 같은 일상적인 표현으로 충분하다. CSS를 몰라도, 색상 코드를 몰라도 상관없다.
5. **참고 자료를 주면 결과가 달라진다** — 공식 매뉴얼 URL이나 스크린샷을 함께 제공하면 AI가 훨씬 정확한 결과물을 만들어준다.
6. **완벽할 필요 없다** — 처음부터 완벽한 결과를 기대하지 말고, 만들고 → 보고 → 고치고를 반복한다. 22번의 명령 중 절반은 수정과 개선이다.

## 이 챕터에서 만들 것

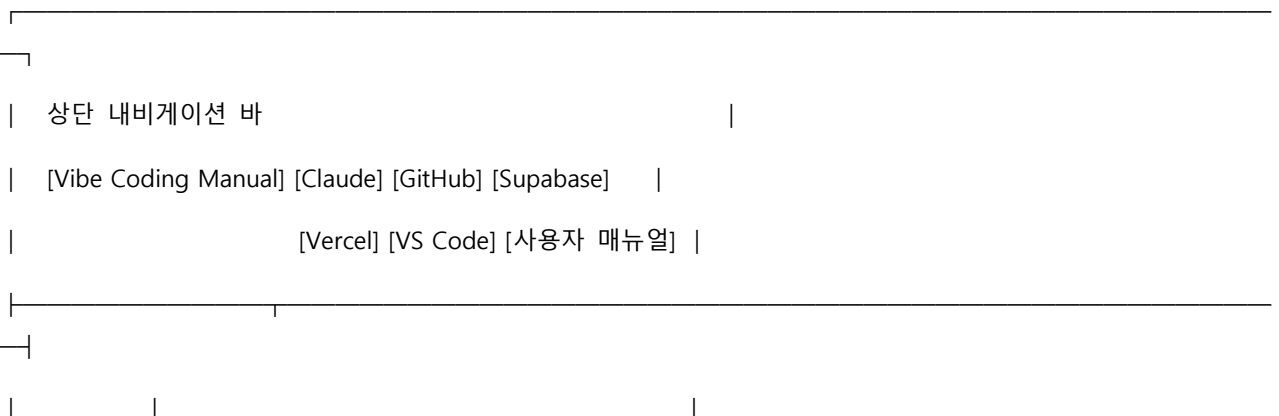
이 챕터를 따라하면, 아래와 같은 **바이브코딩 완벽 매뉴얼** 웹사이트를 직접 만들 수 있다. 코딩 경험이 전혀 없어도 괜찮다. AI에게 말만 하면 된다.



## 완성본 미리보기

### 전체 레이아웃

완성된 사이트는 하나의 HTML 파일(index.html)로 이루어져 있다. 화면은 크게 4개 영역으로 구성된다.





왼쪽	오른쪽 본문 영역	
사이드바		
메뉴	히어로 배너 / 카드 / 로드맵	
	또는 각 도구별 상세 매뉴얼	
<hr/>		
하단 푸터		
사이트 소개   가이드 목록   관련 링크		
<hr/>		
<hr/>		

## 상단 내비게이션

화면 최상단에 고정된 메뉴 바다. 6개의 탭 버튼이 있고, 각 버튼을 누르면 해당 섹션으로 이동한다.

버튼	색상	역할
Claude	하늘색	AI 코딩 어시스턴트 가이드
GitHub	보라색	버전 관리 & 협업 가이드
Supabase	초록색	백엔드 서비스 가이드
Vercel	빨간색	배포 플랫폼 가이드
VS Code	핑크색	코드 편집기 가이드
사용자 매뉴얼	노란색	이 사이트 사용법

## 왼쪽 사이드바

각 도구별 하위 메뉴가 카테고리별로 정리되어 있다. 총 36개 페이지다.

### Claude (8개 메뉴)

- Claude 4종 소개
- 설치 가이드 (7단계)
- 기본 사용법
- 고급 기능 & MCP

- VS Code 연동
- 사용료 체계
- 토큰 절감 방안
- Anthropic 이야기

#### **GitHub (6개 메뉴)**

- 소개 및 설정
- Git 핵심 명령어
- 저장소 관리
- GitHub Flow
- 마크다운 문법
- GitHub Actions

#### **Supabase (6개 메뉴)**

- 소개 및 설정
- 데이터베이스
- 인증(Auth)
- 스토리지
- RLS & 보안
- 벡터 DB & AI

#### **Vercel (6개 메뉴)**

- 소개 및 배포
- 프레임워크
- Functions & API
- 설정 및 도메인
- 환경변수 관리
- AI 인프라

#### **VS Code (6개 메뉴)**

- 소개 및 설치
- UI 구조 & 편집
- 확장 프로그램
- Git & 소스 제어
- 디버깅

- 설정 및 단축키

## 사용자 매뉴얼 (4개 메뉴)

- 시작하기
- 화면 구성
- 주요 기능
- 자주 묻는 질문

## 오른쪽 본문 영역

홈 화면에는 3가지 요소가 있다.

1. **히어로 배너** — "바이브코딩 완벽 매뉴얼" 제목과 설명
2. **5개 카드** — Claude, GitHub, Supabase, Vercel, VS Code 각각을 한 줄로 소개. 클릭하면 해당 섹션으로 이동
3. **설치 순서 로드맵** — 8단계의 세로 타임라인. VS Code 설치부터 개발 시작까지의 순서를 안내

각 도구의 상세 페이지에는 설명, 코드 블록, 명령어 복사 버튼이 포함되어 있다.

## 하단 푸터

3컬럼 구성이다.

- 왼쪽: 사이트 소개 문구
- 가운데: 포함된 가이드 목록 (Claude, GitHub, Supabase, Vercel, VS Code)
- 오른쪽: 각 도구의 공식 사이트 링크

---

이제부터 이 완성본을 바이브 코딩으로 직접 만들어 볼 것이다. 아래 단계를 순서대로 따라가면 된다.

## Step 2. 프로젝트 시작하기

### 2-1. 프로젝트 폴더 만들기

VS Code를 열고 터미널을 연다 (단축키: Ctrl + `).

원하는 위치에 프로젝트 폴더를 만든다.

```
mkdir my-vibe-manual
```

```
cd my-vibe-manual
```

### 2-2. Claude Code 실행

같은 터미널에서 Claude Code를 실행한다.

```
claude
```

실행되면 프롬프트가 나타난다. 이제 AI와 대화할 준비가 끝났다.

Claude Code	
/my-vibe-manual	

> 프롬프트에 우리가 원하는 것을 말하면 된다.

### Step 3. 22번의 명령으로 웹사이트 완성하기

이제부터 Claude Code에게 명령을 내려서 웹사이트를 만든다. 코딩을 한 줄도 모르는 사람이 웹사이트를 만들 수 있는 이유가 여기에 있다. **만들고 싶은 것을 말하면, AI가 코드를 작성해준다.**

핵심 원칙은 **한 번에 하나씩**이다. 여러 가지를 한꺼번에 요청하면 원하는 결과가 나오기 어렵다. 하나를 요청하고, 결과를 확인하고, 다음을 요청한다.

**결과 확인하기:** 명령을 내릴 때마다 브라우저에서 결과를 확인한다. VS Code에서 index.html을 열고 마우스 오른쪽 버튼 → "Open with Live Server"를 클릭하면 된다. 또는 index.html 파일을 브라우저로 직접 드래그해도 된다.

### Phase 1. 기본 구조 만들기 (명령 1~6)

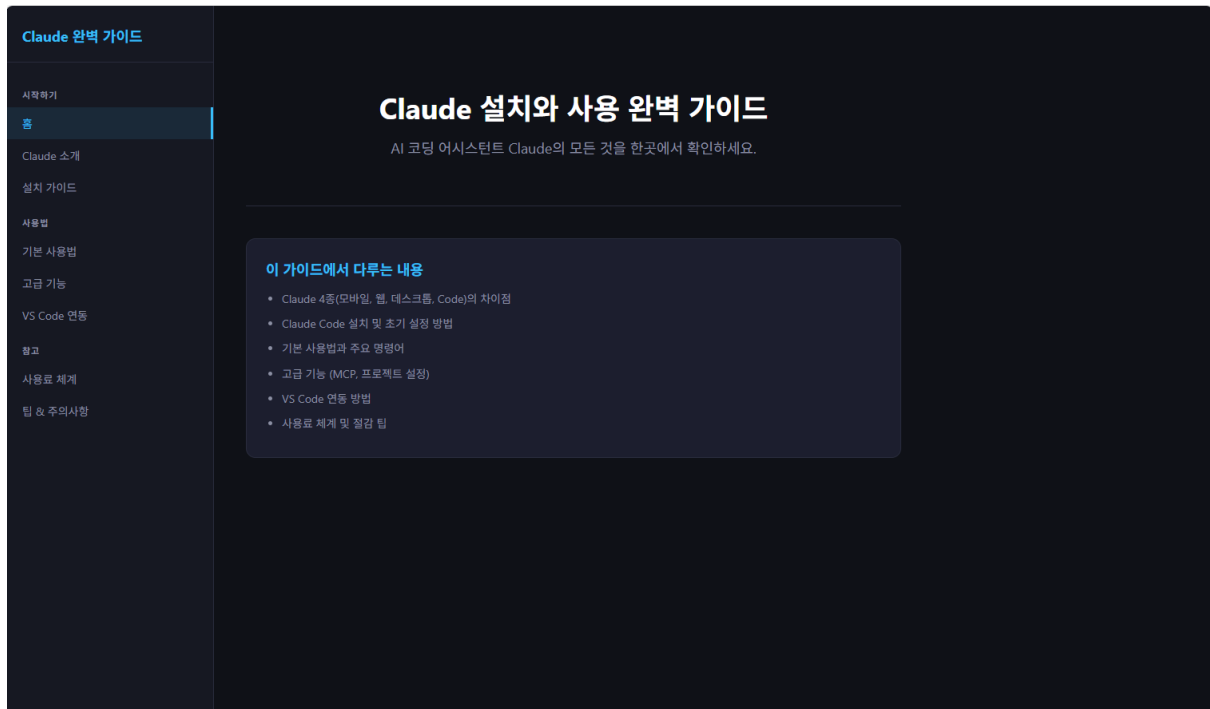
사이트의 뼈대를 세우는 단계다. 레이아웃, 콘텐츠, 내비게이션, 사이드바가 만들어진다.

바로 시작한다. 여러분도 아래 명령어를 클로드코드에게 말하면 된다.

#### ● 1번째 명령 (기본 뼈대 만들기):

```
> Claude 설치와 사용에 대한 완벽 가이드 웹페이지를 만들어줘.
index.html 단일 파일로 만들고, 왼쪽에 사이드바 메뉴, 오른쪽에 본문이 있는
레이아웃으로 해줘. 다크 모드 테마로 해줘.
```

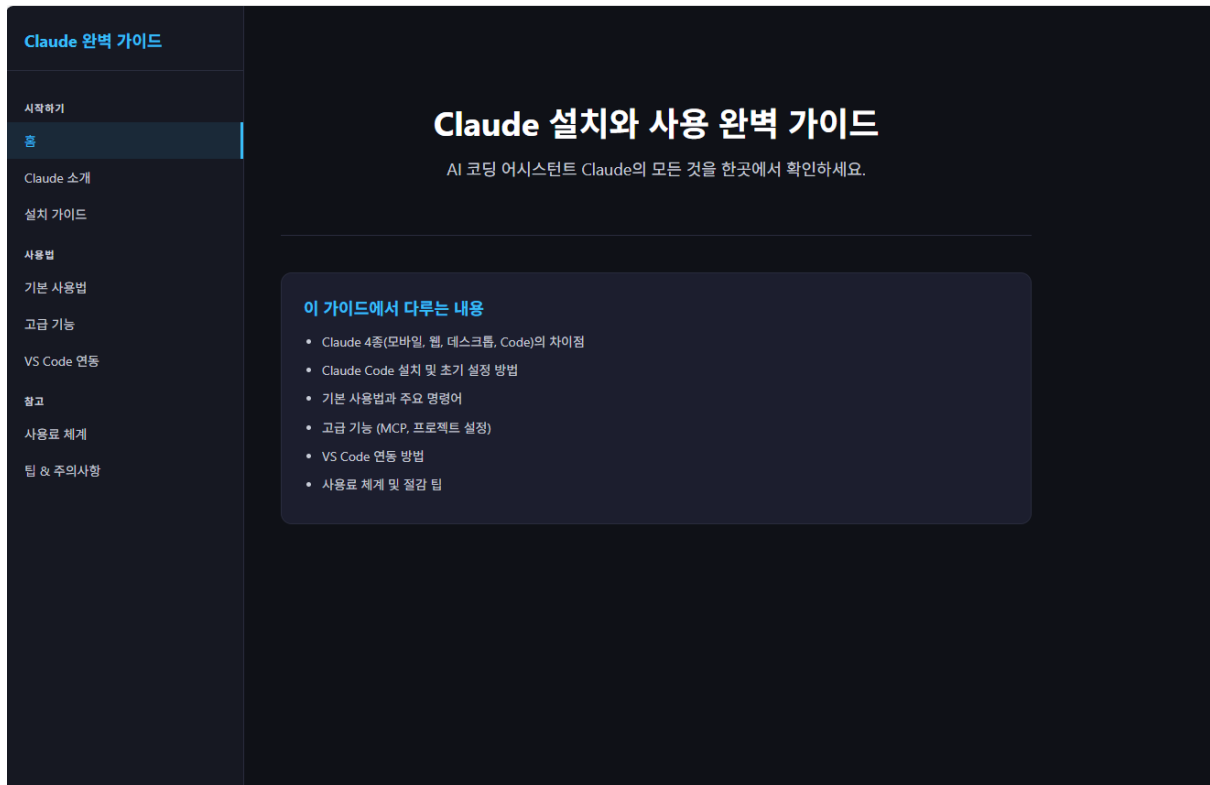
이 한마디에 AI가 약 770줄의 HTML/CSS/JavaScript 코드를 생성한다. 아래 이미지처럼 다크 모드 배경에 사이드바와 본문이 있는 웹페이지가 약 30초~1분 만에 만들어진다



● 2번째 명령 (수정):

> 본문 글씨가 어두워서 잘 안 보여. 글씨를 충분히 밝게 해서 잘 읽히도록 해줘.

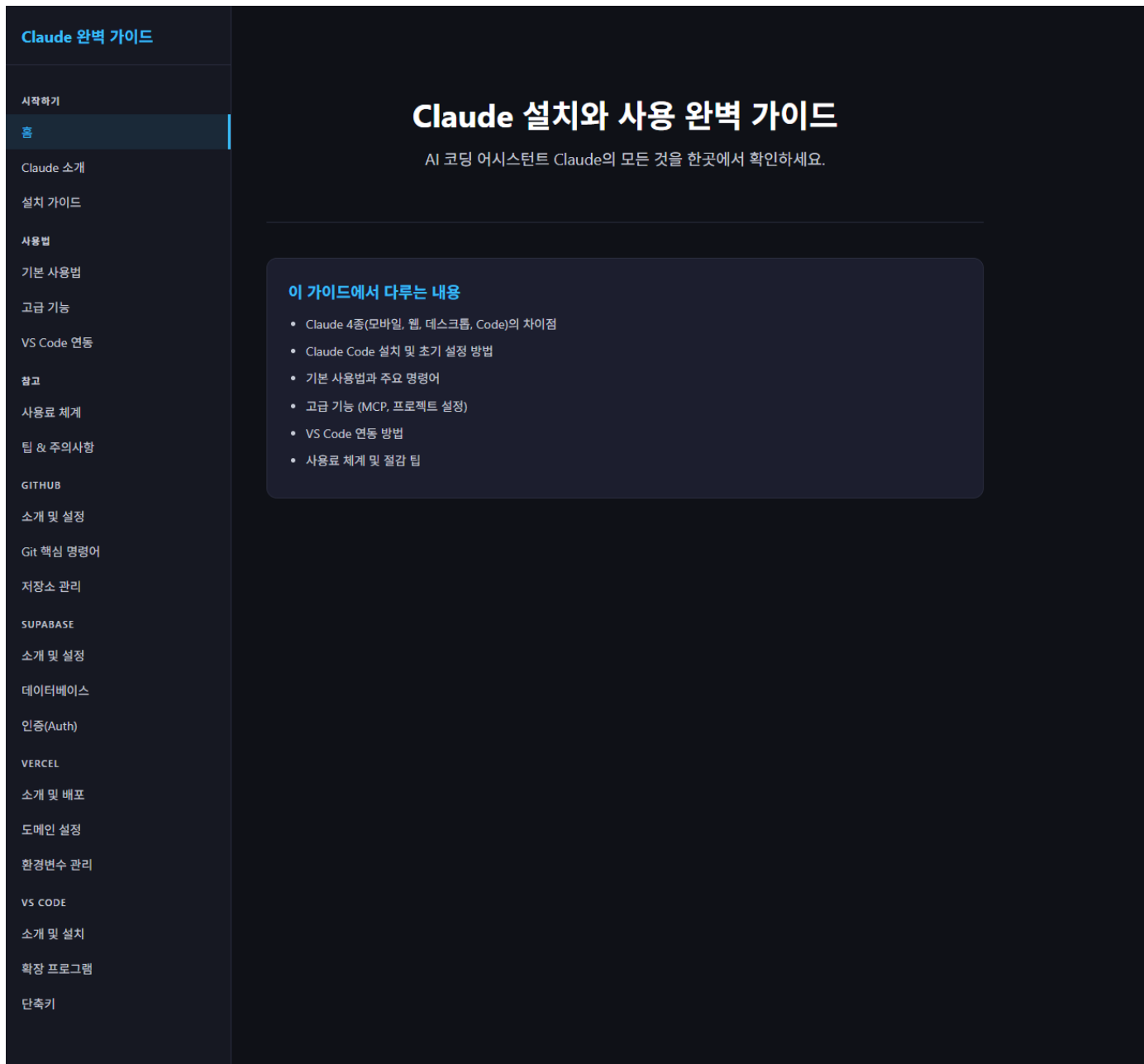
다크 모드에서 글씨가 어두우면 이렇게 느낀 그대로 말하면 된다. 아래처럼 AI가 CSS 색상을 조정해준다.



- 3번째 명령 (섹션 추가):

```
> GitHub, Supabase, Vercel, VS Code 섹션도 추가해줘.  
각 도구별로 소개와 핵심 기능을 설명해줘.
```

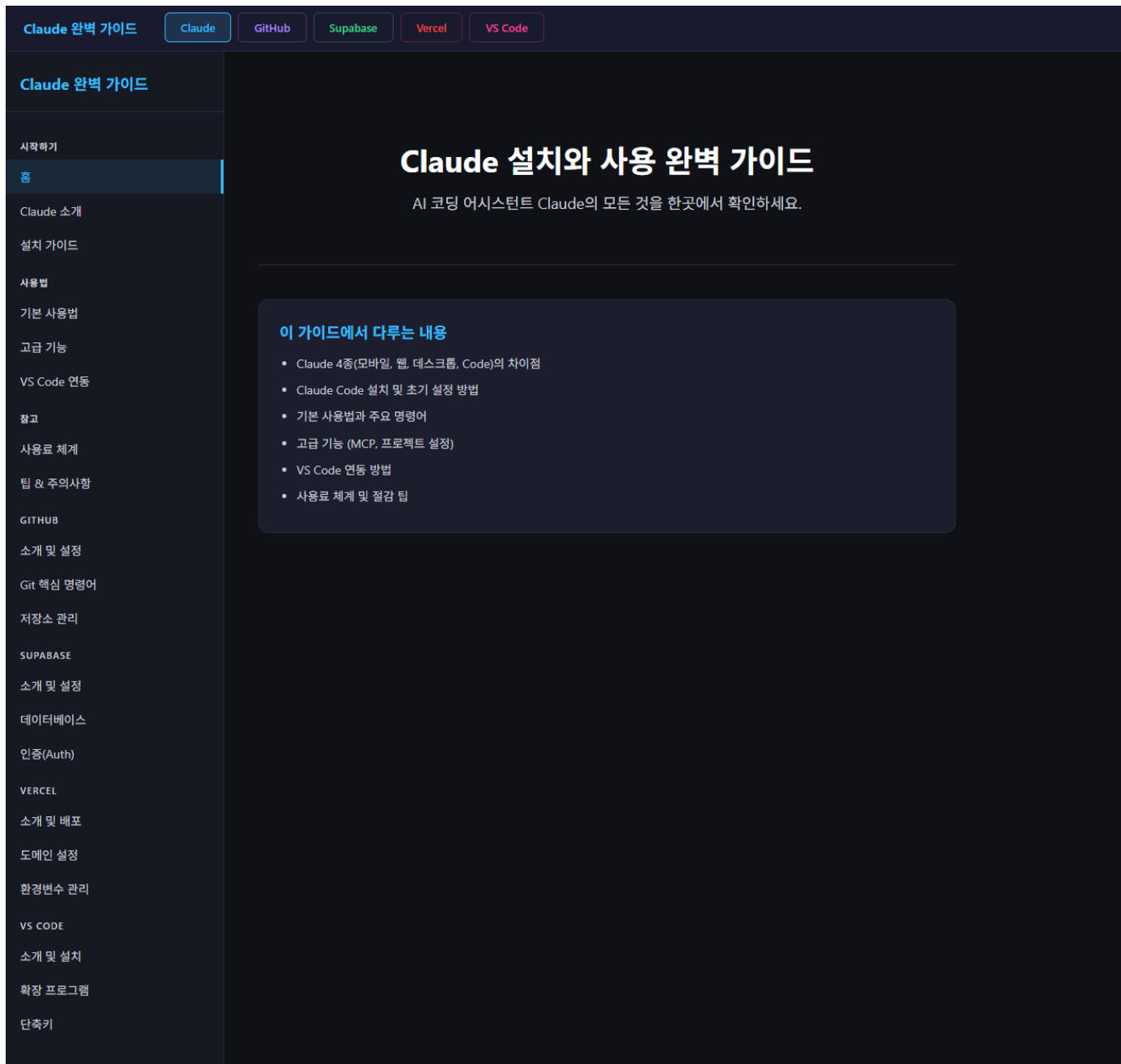
Claude만 있던 사이트에 4개 도구가 추가된다. 각 도구별 콘텐츠가 자동으로 생성된다.



● 4번째 명령 (상단 탭 추가):

> 상단에 내비게이션 탭 버튼을 만들어서 Claude, GitHub, Supabase, Vercel, VS Code를 선택할 수 있게 해줘.

화면 상단에 탭 버튼이 생긴다. 각 버튼을 클릭하면 해당 도구의 섹션으로 이동한다.

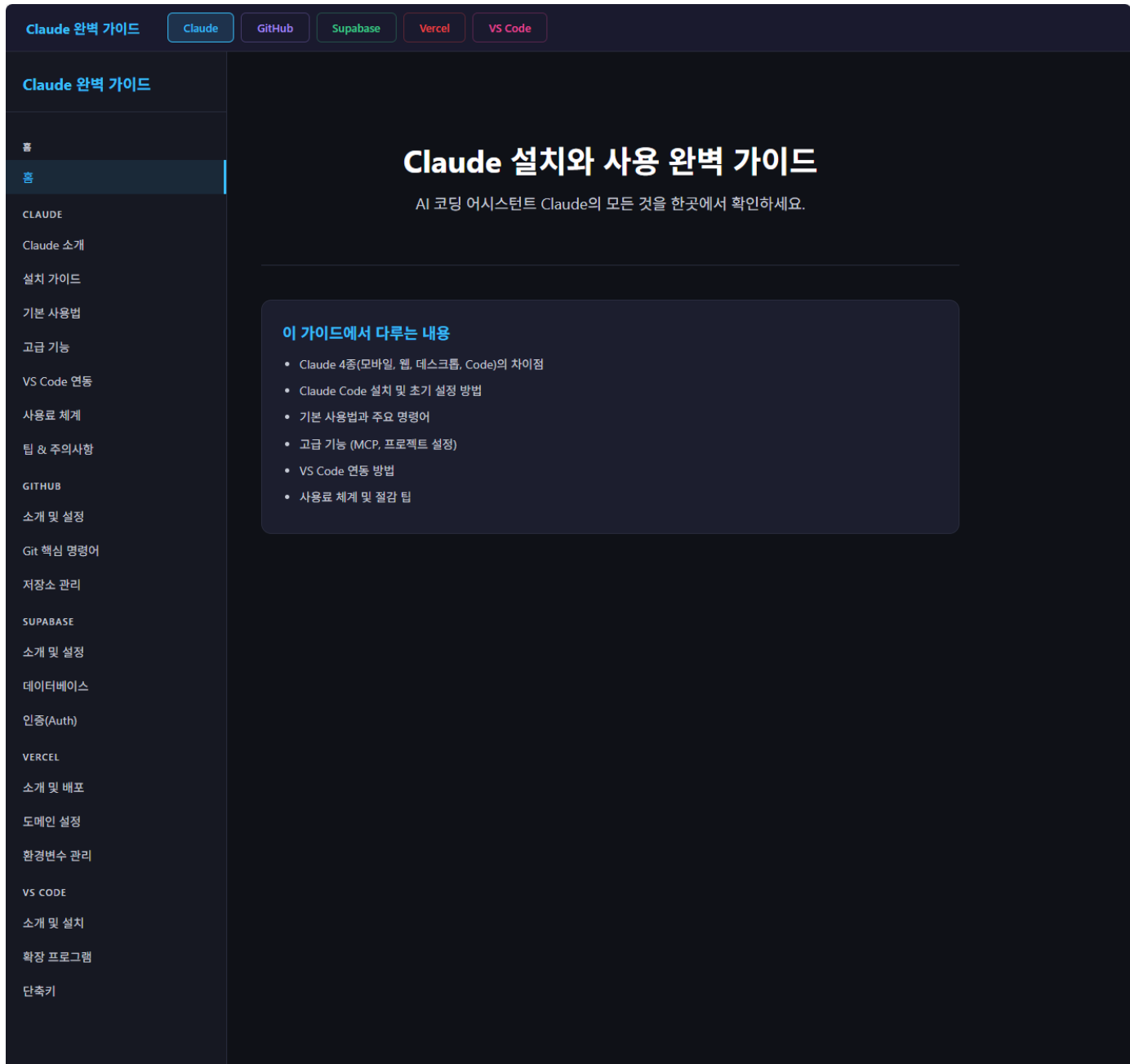


● 5번째 명령 (사이드바 정리):

> 왼쪽 사이드바에 각 도구별 하위 메뉴를 카테고리별로 나눠서 정리해줘.

뒤섞여 있던 사이드바 메뉴가 도구별로 깔끔하게 분류된다.

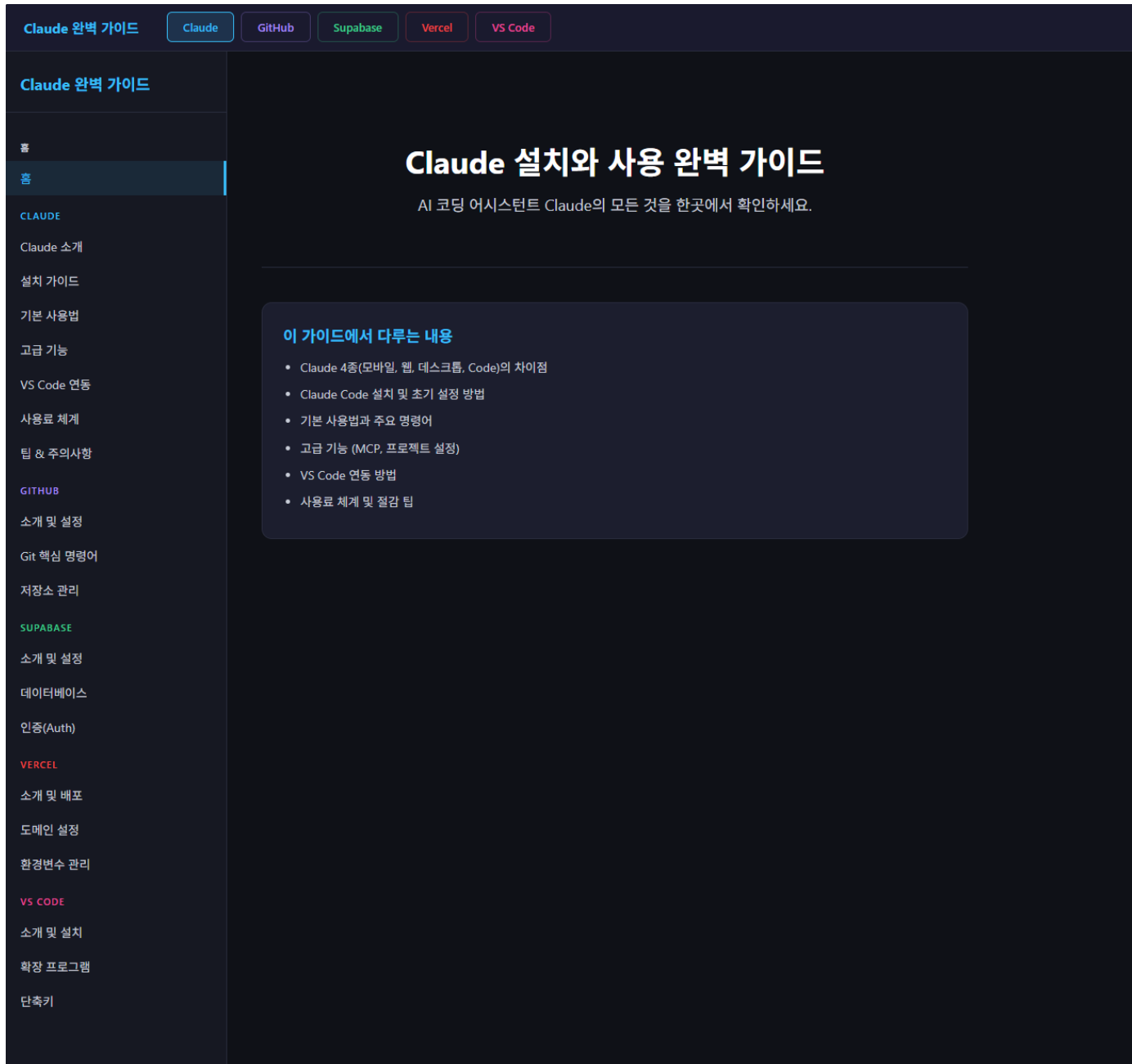




● 6번째 명령 (사이드바 색상):

> 사이드바의 카테고리 제목에 상단 탭 버튼과 같은 색상을 적용해줘.

사이드바에 브랜드 색상이 적용된다. Claude는 하늘색, GitHub는 보라색, Supabase는 초록색, Vercel은 빨간색, VS Code는 핑크색.



여기까지가 바이브 코딩의 핵심이다. 6번의 명령으로 사이트의 기본 구조가 완성됐다. 코드를 한 줄도 직접 쓰지 않았다. 만들고 싶은 것을 말했고, 마음에 안 드는 것을 말했을 뿐이다.

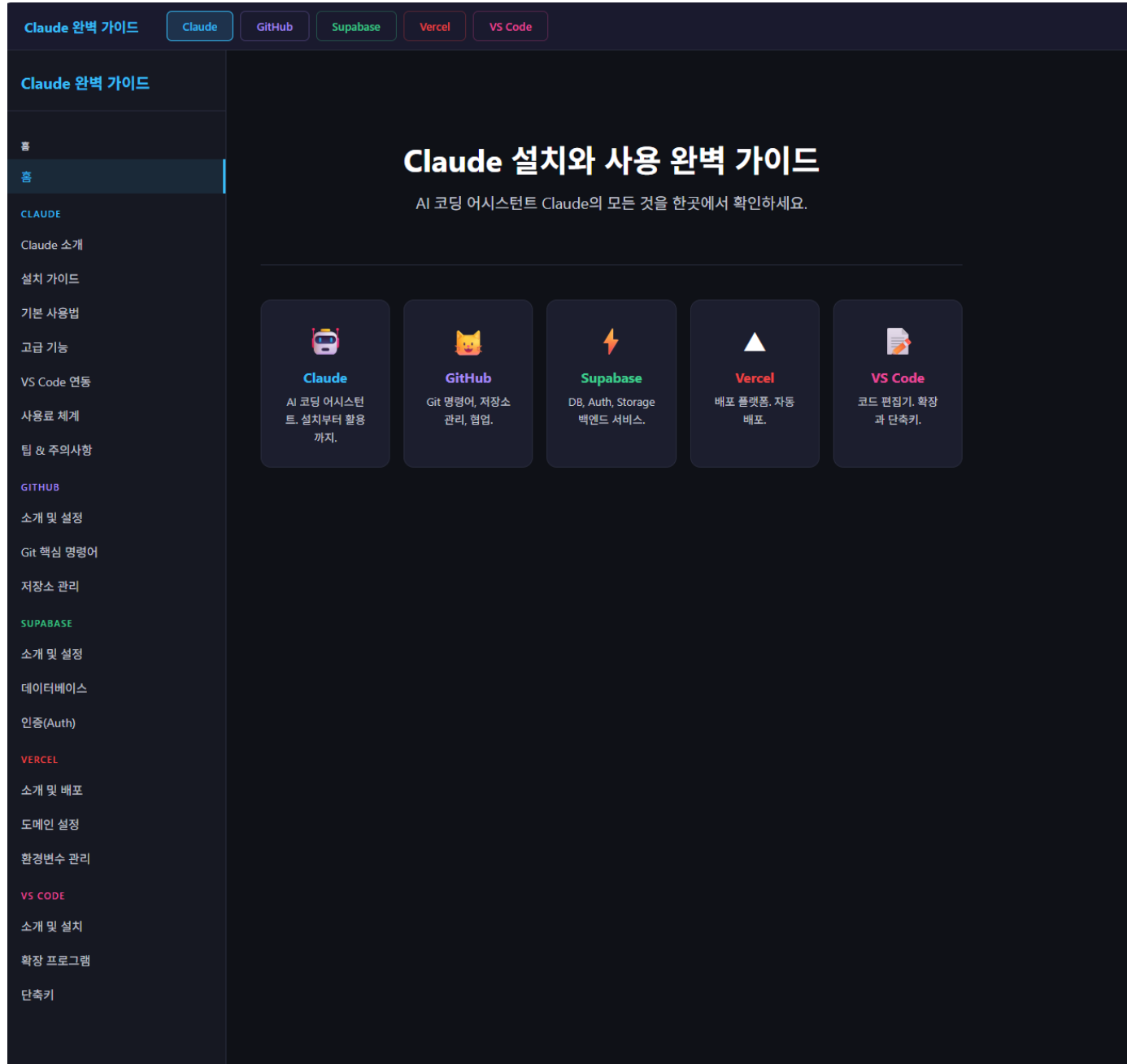
## Phase 2. 홈 화면 & 레이아웃 완성 (명령 7~10)

홈 화면을 풍성하게 만들고, 전체 레이아웃을 다듬는 단계다.

- 7번째 명령 (홈 화면 카드 추가):

> 홈 화면에 각 도구(Claude, GitHub, Supabase, Vercel, VS Code)를 소개하는 카드를 추가해줘.

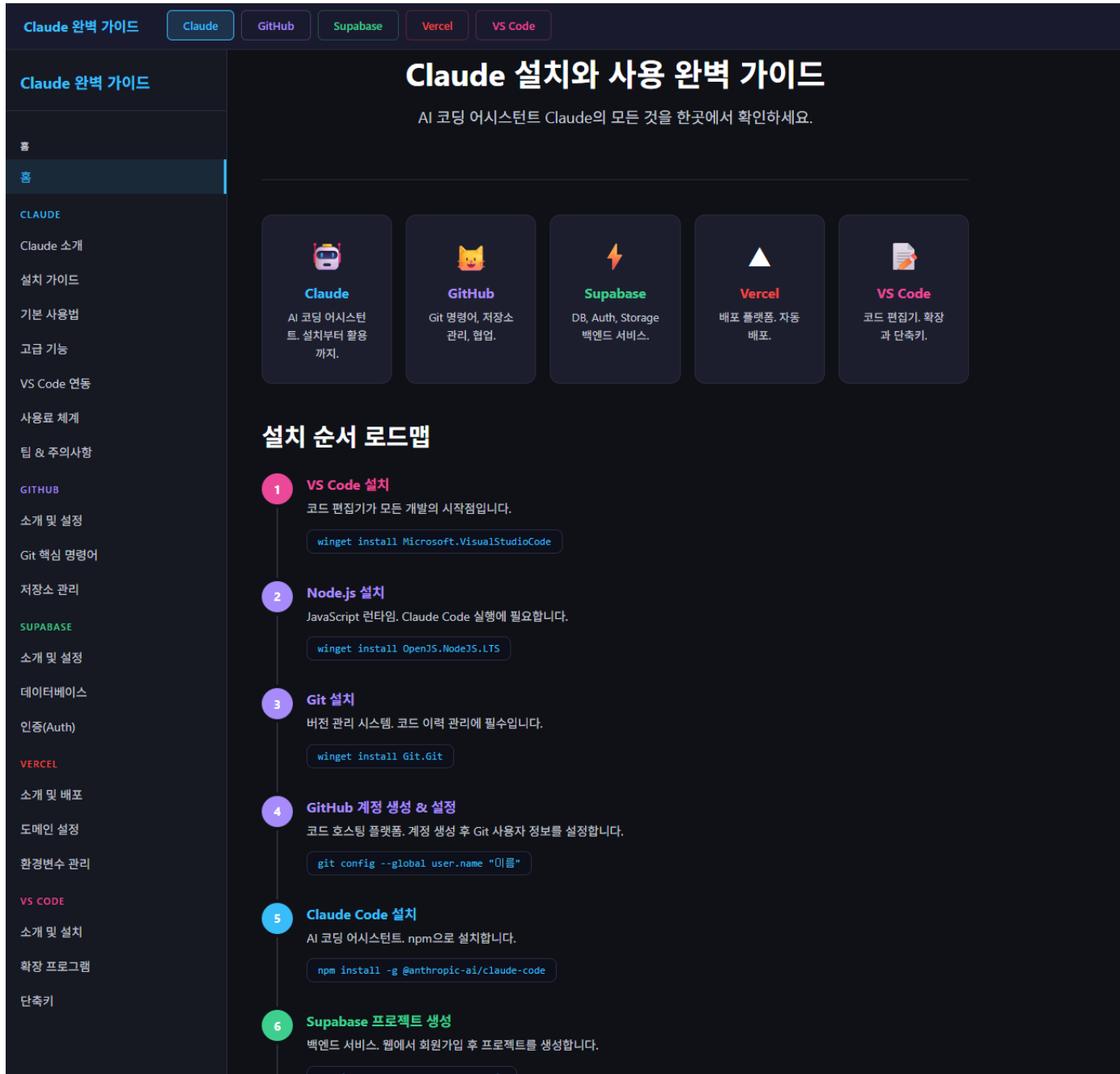
홈 화면에 5개의 카드가 나타난다. 각 도구를 아이콘+제목+설명으로 한눈에 보여준다.



● 8번째 명령 (설치 로드맵 추가):

> 홈 화면 카드 아래에 설치 순서 로드맵을 세로 타임라인 형태로 추가해줘.

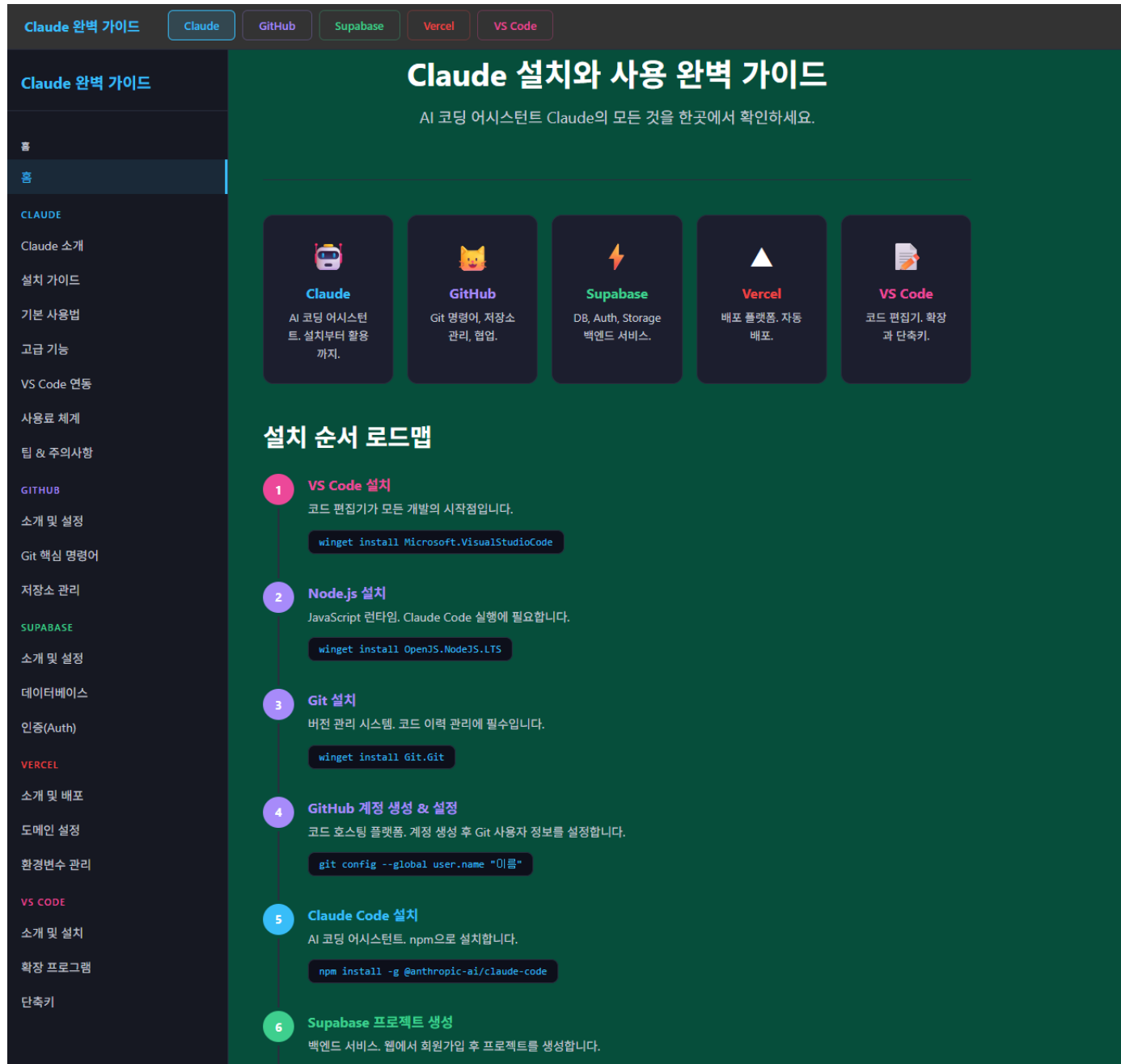
카드 아래에 8단계 세로 타임라인이 추가된다. VS Code 설치부터 개발 시작까지의 순서를 안내한다.



● 9번째 명령 (배경색 변경):

> 상단 내비게이션 바 배경색을 진한 회색으로, 페이지 배경색을 진한 녹색으로 바꿔줘.

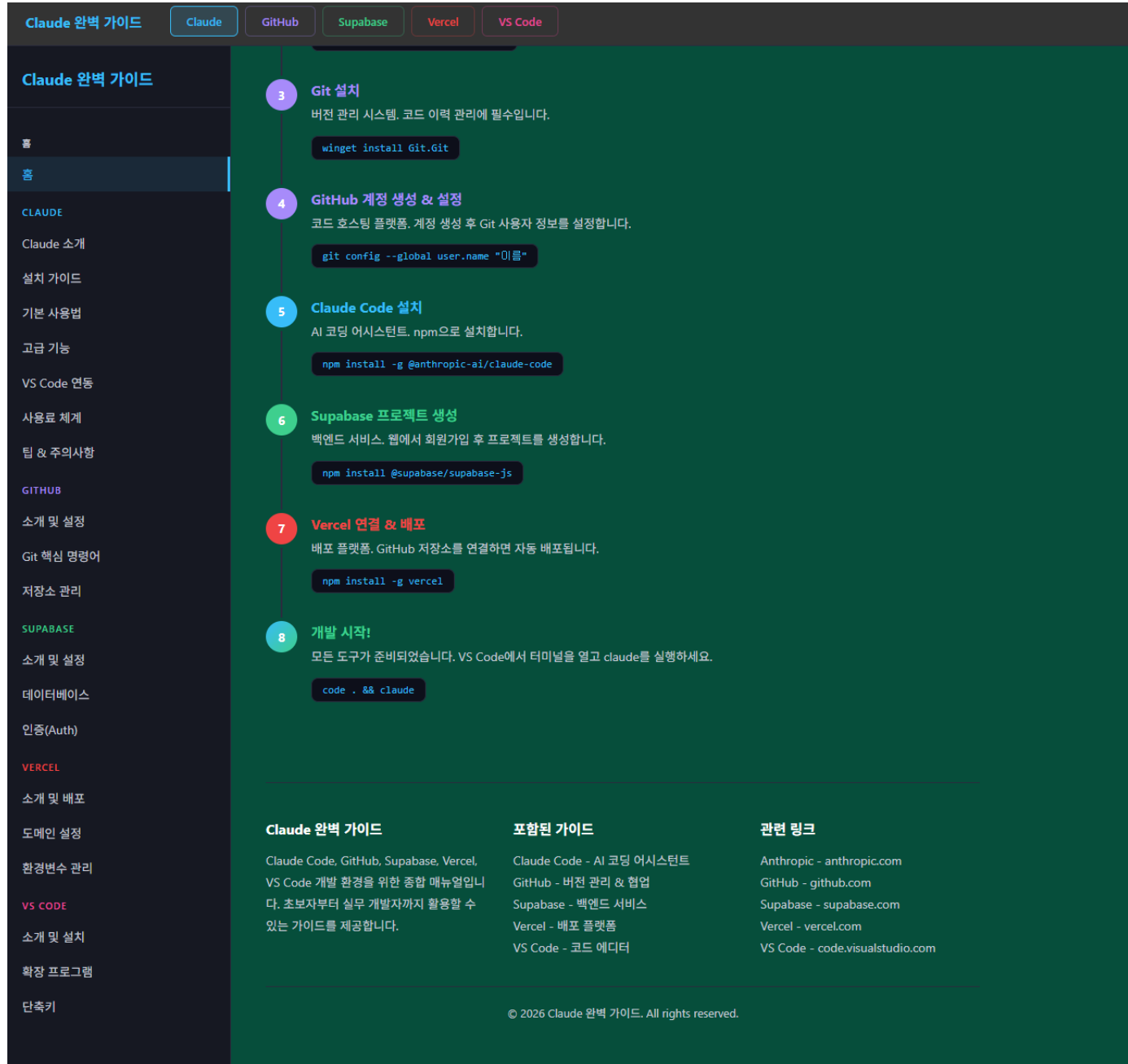
전체 색감이 바뀐다. 이처럼 색상을 바꾸고 싶으면 느낀 대로 말하면 된다.



● 10번째 명령 (푸터 추가):

> 페이지 맨 아래에 푸터를 추가해줘. 사이트 소개, 포함된 가이드 목록, 관련 링크를 3컬럼으로 보여줘.

페이지 하단에 3컬럼 푸터가 생긴다.

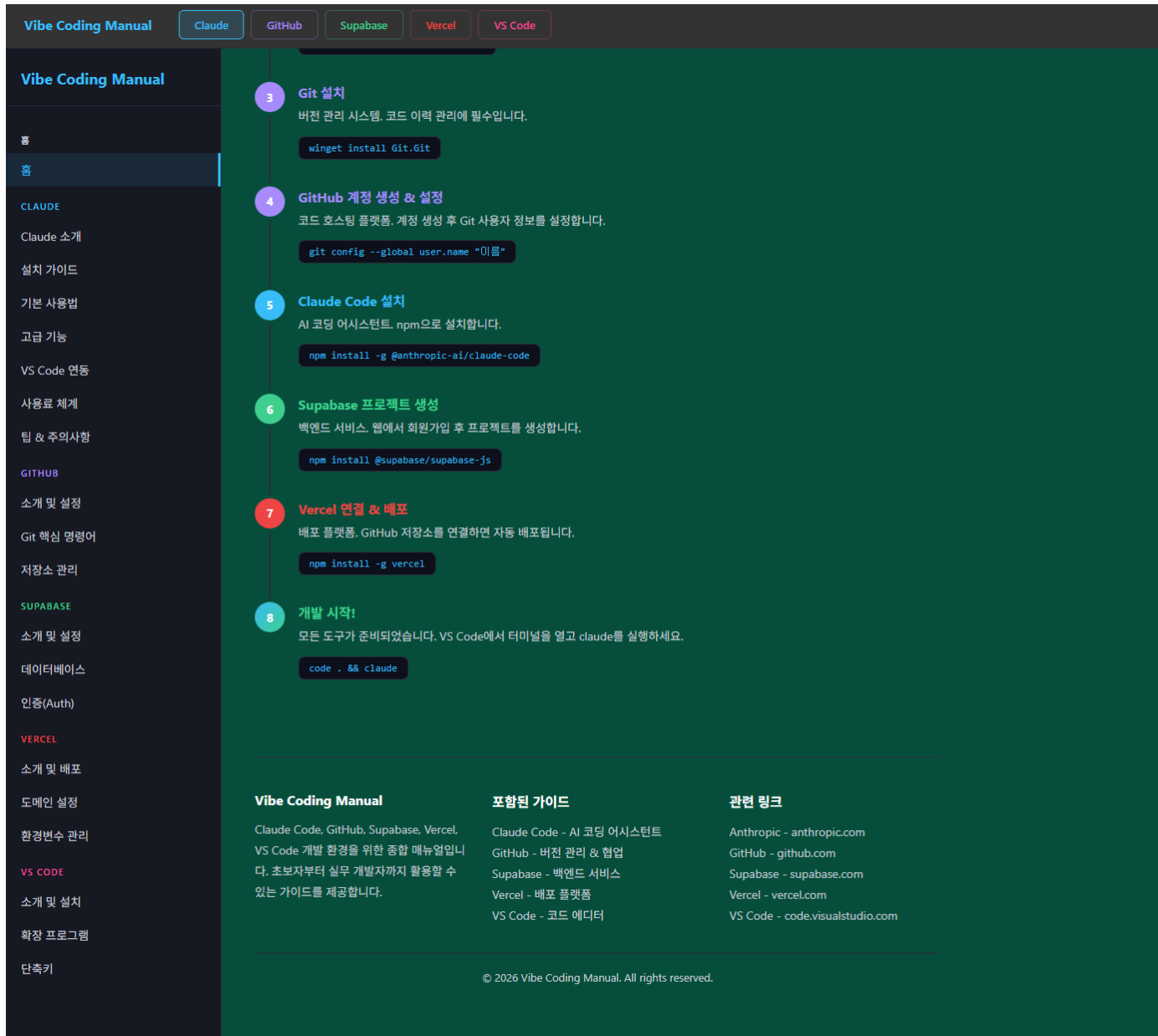


### Phase 3. 디자인 & 품질 향상 (명령 11~16)

브랜딩, 색상, 콘텐츠 품질을 높이는 단계다. 사이트의 완성도가 확 달라진다.

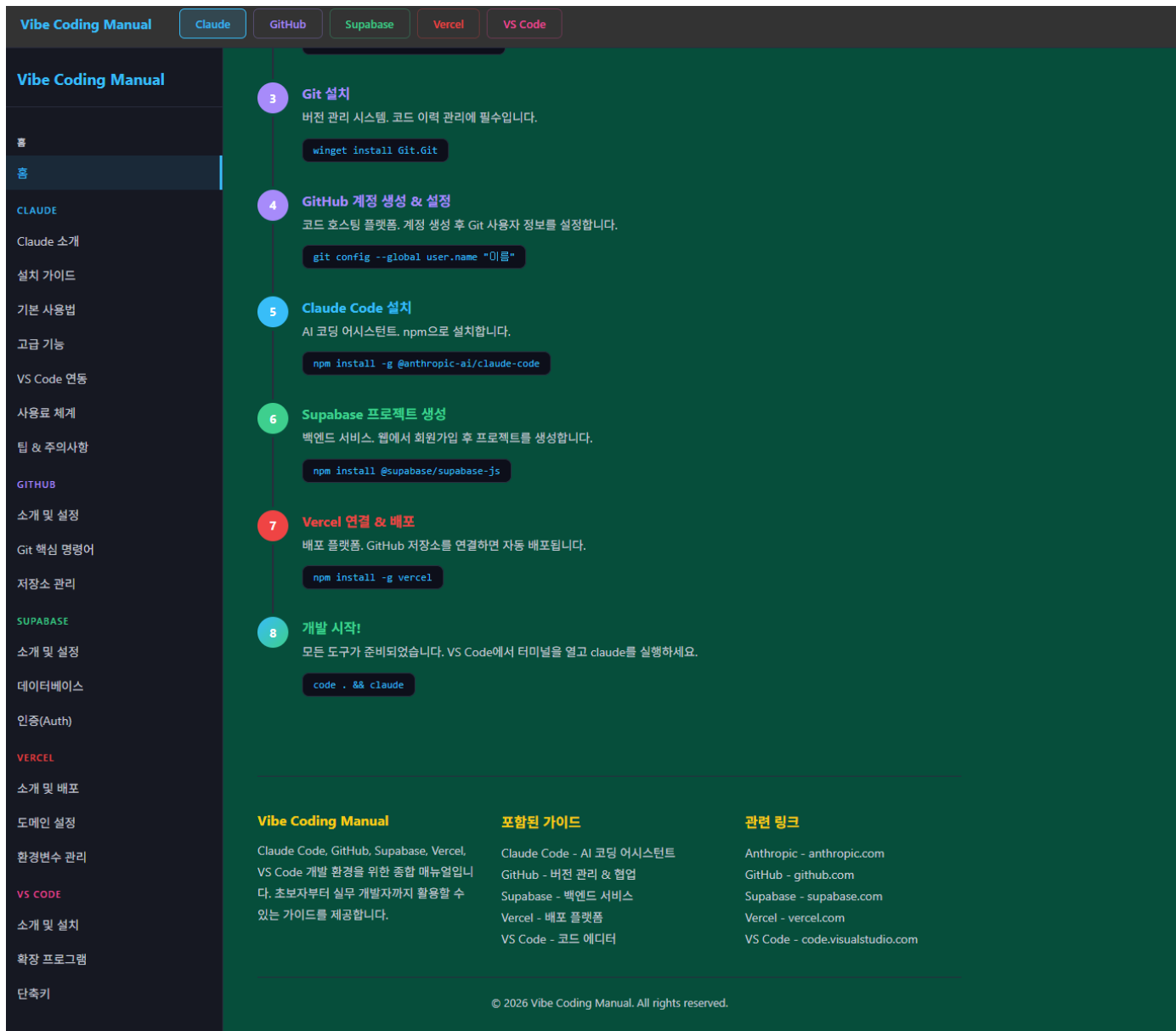
● 11번째 명령 (브랜딩 변경):

> 사이트 이름을 'Vibe Coding Manual'로 바꿔줘. 상단 로고, 사이드바 로고, 푸터 제목 모두 변경해줘.



● 12번째 명령 (푸터 제목 색상 변경):

> 푸터의 제목들을 노란색으로 바꿔줘.



"지금까지는 Claude Code가 알아서 만든 결과이다. 하지만 공식 매뉴얼 링크를 함께 제공하면 훨씬 정확하고 풍부한 콘텐츠를 만들어준다."

바이브 코딩 팁: AI에게 참고 자료를 주면 결과가 달라진다

지금까지 만든 콘텐츠는 Claude Code가 자체 지식으로 작성한 것이다.

충분히 쓸 만하지만, 공식 매뉴얼 링크를 함께 제공하면 훨씬 정확하고 상세한 콘텐츠가 나온다.



이것은 사람에게 "보고서 써줘"라고만 하는 것과 "이 자료 참고해서 보고서 써줘"라고 하는 것의 차이와 같다.

● 13번째 명령:

```
> GitHub 공식 매뉴얼(https://docs.github.com)을 참고해서
  GitHub 소개 섹션을 더 정확하고 상세하게 업데이트해줘.
```

참고 자료 URL을 함께 제공하면 AI가 훨씬 정확하고 풍부한 콘텐츠를 만들어준다. 사람에게 "보고서 써줘"라고만 하는 것과 "이 자료 참고해서 보고서 써줘"라고 하는 것의 차이와 같다.

The screenshot displays the 'Vibe Coding Manual' website with a dark green theme. The top navigation bar includes tabs for 'Claude', 'GitHub', 'Supabase', 'Vercel', and 'VS Code'. The left sidebar lists various manual sections, with 'GitHub' and its sub-section '소개 및 설정' (Introduction and Setup) highlighted. The main content area is titled 'GitHub 소개 및 설정' and provides a comprehensive overview of GitHub, including its definition as a code hosting platform, a list of key features like Repositories, Issues, Pull Requests, Actions, and Pages, and a step-by-step guide for account creation and initial Git configuration. The setup guide includes terminal commands for setting the global user name and email, and a command to list the configured settings.

**Vibe Coding Manual** Claude GitHub Supabase Vercel VS Code

**Vibe Coding Manual**

홈

CLAUDE

Claude 소개

설치 가이드

기본 사용법

고급 기능

VS Code 연동

사용료 체계

팁 & 주의사항

GITHUB

소개 및 설정

Git 핵심 명령어

저장소 관리

SUPABASE

소개 및 설정

데이터베이스

인증(Auth)

VERCEL

소개 및 배포

도메인 설정

환경변수 관리

VS CODE

소개 및 설치

확장 프로그램

단축키

**GitHub 소개 및 설정**

GitHub는 1억 명 이상의 개발자가 사용하는 세계 최대의 코드 호스팅 플랫폼입니다.

**GitHub란?**

GitHub는 Git 기반의 코드 호스팅 및 협업 플랫폼입니다. 2008년 설립되어 현재 Microsoft 소속입니다.

- **Repositories(저장소)** — 프로젝트의 코드, 파일, 이력을 보관하는 공간
- **Issues** — 버그 신고, 기능 요청, 작업 관리를 위한 트래커
- **Pull Requests** — 코드 변경을 제안하고 리뷰받는 협업 도구
- **Actions** — 테스트, 빌드, 배포를 자동화하는 CI/CD
- **Pages** — 저장소에서 직접 정적 웹사이트 호스팅

**계정 생성**

- github.com 접속 → "Sign up" 클릭
- 이메일, 비밀번호, 사용자명 입력
- Free 플랜으로 충분 (무제한 공개/비공개 저장소)

**Git 초기 설정**

GitHub 가입 후 로컬 Git에 사용자 정보를 등록합니다. 이 정보가 커밋 기록에 표시됩니다.

```
git config --global user.name "내 이름"
```

```
git config --global user.email "GitHub 가입 이메일"
```

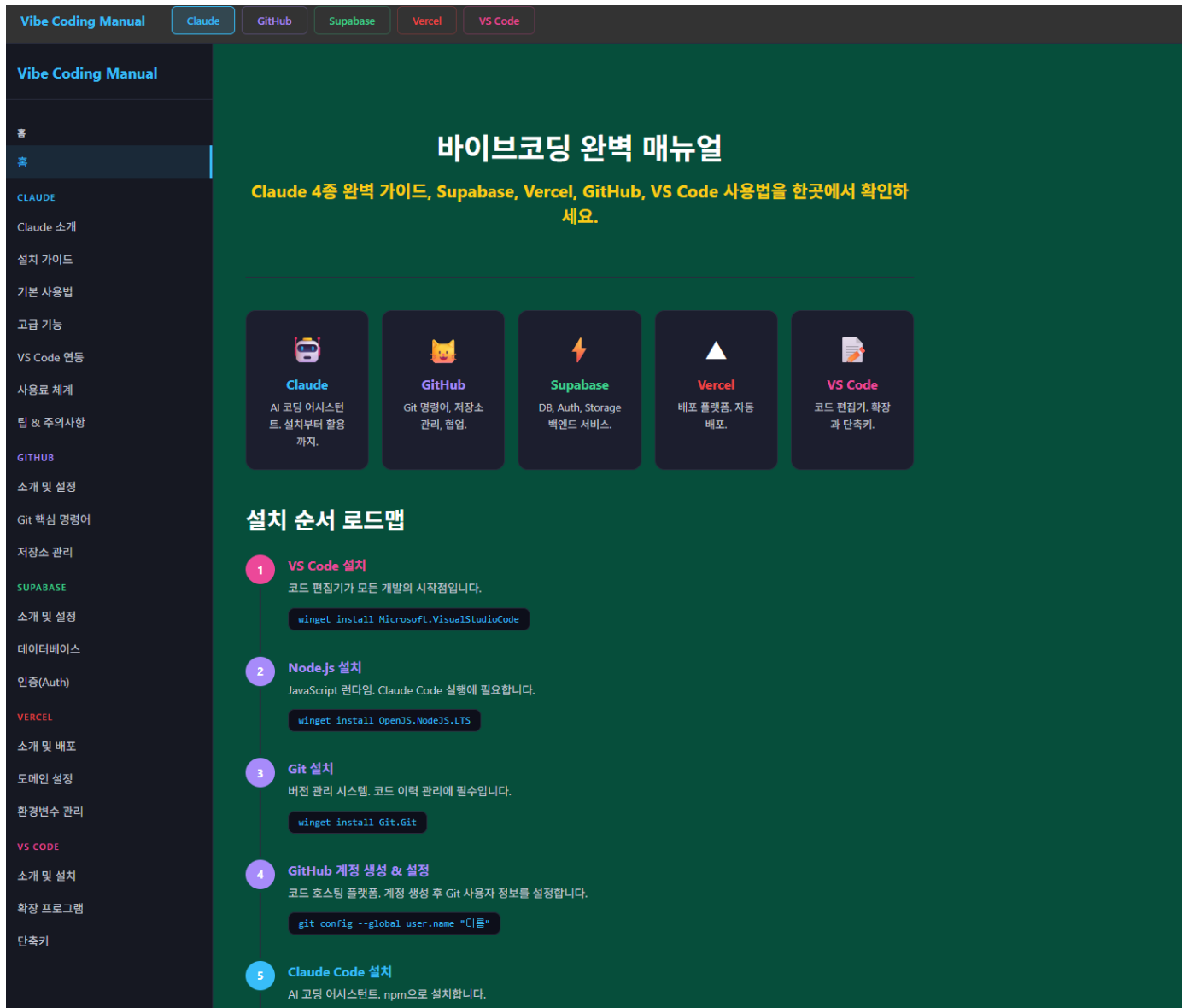
설정 확인:

```
git config --list
```

**SSH 키 설정 (선택):** HTTPS 대신 SSH로 연결하면 매번 비밀번호를 입력하지 않아도 됩니다. GitHub 공식 문서의 "Connecting to GitHub with SSH" 가이드를 참고하세요.

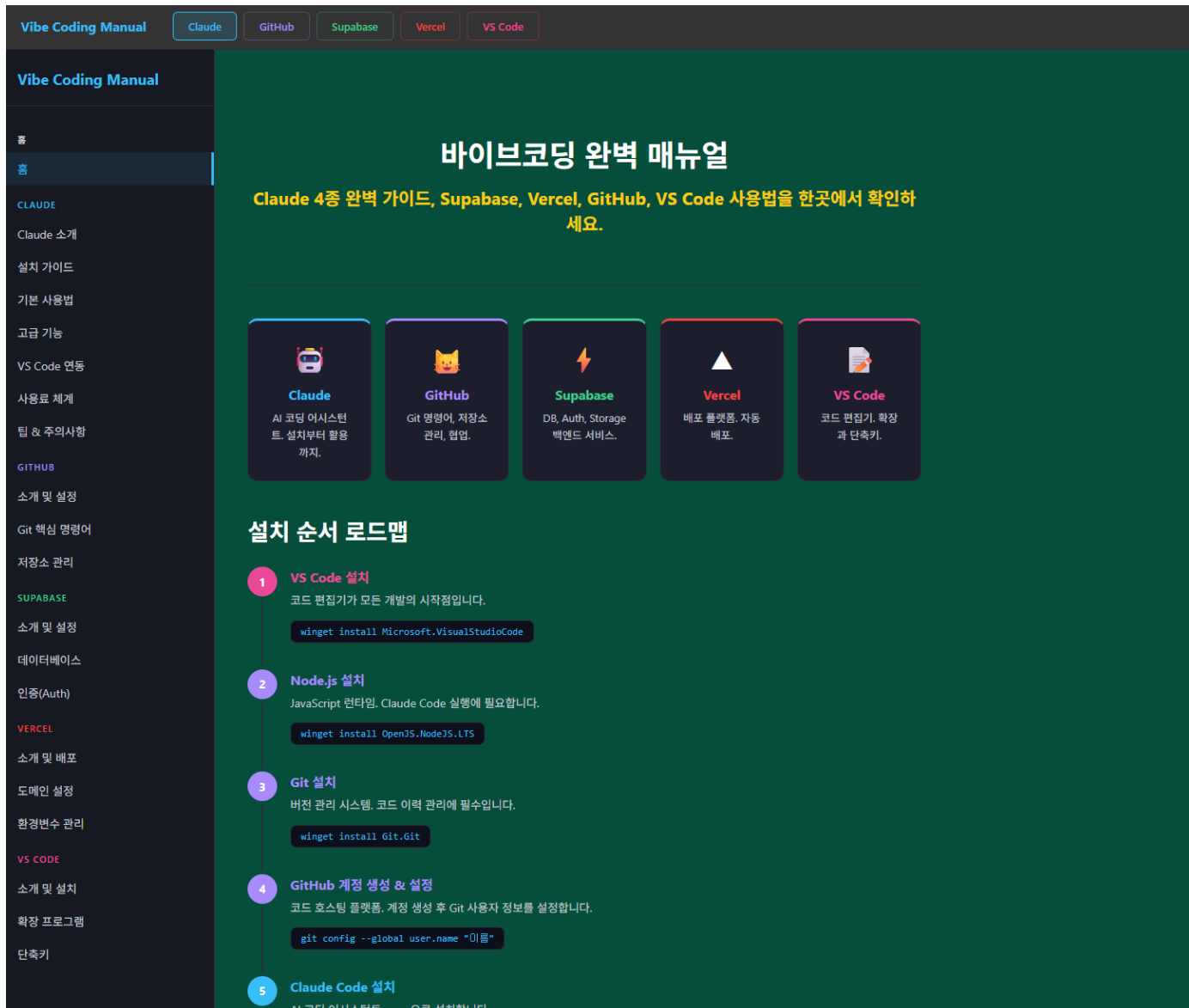
- 14번째 명령 (홈 화면 제목 변경):

> 홈 화면 제목을 '바이브코딩 완벽 매뉴얼'로 바꾸고,  
부제목을 노란색 볼드로 만들어줘.



- 15번째 명령 (카드 색상 테두리):

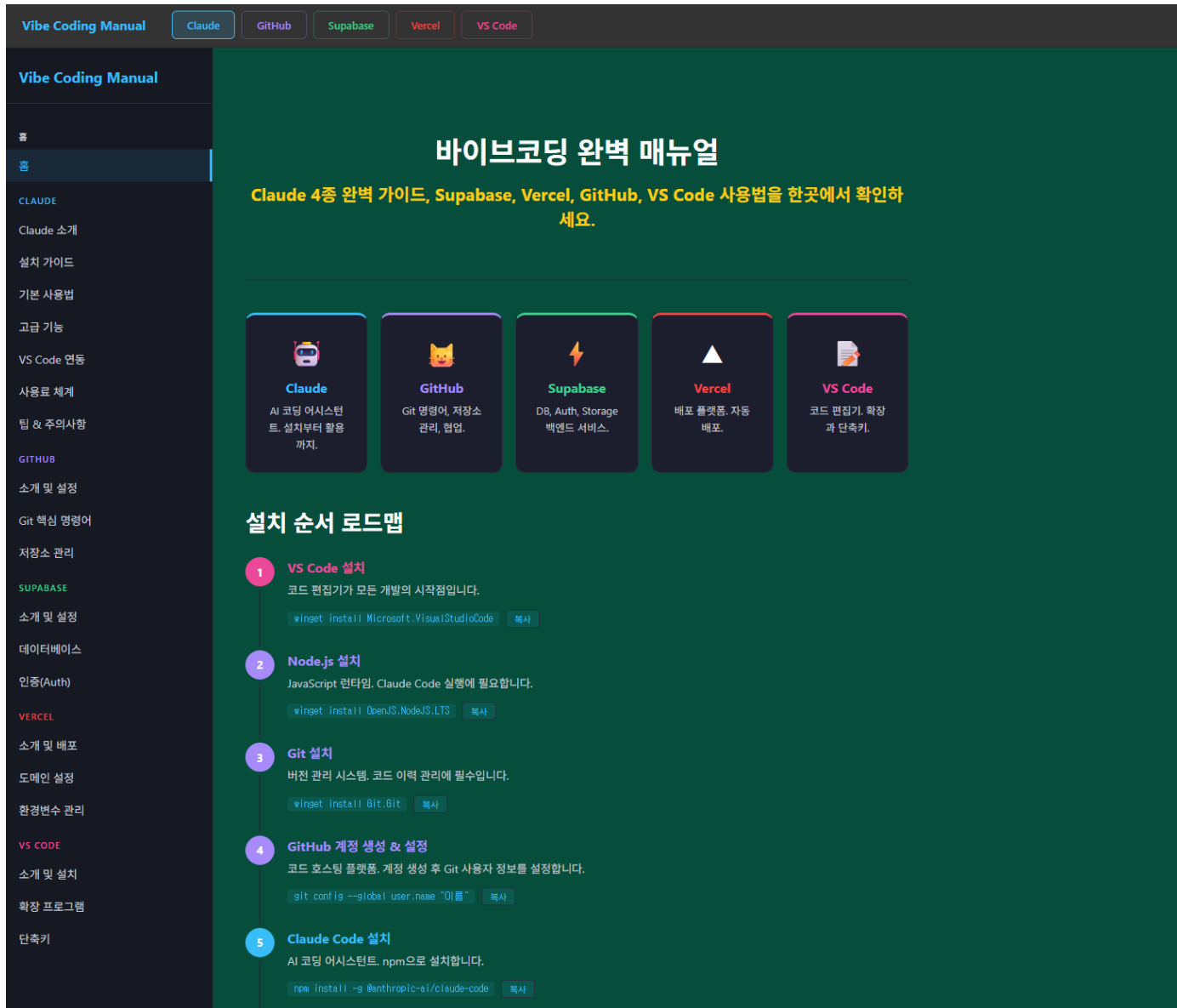
> 홈 화면의 각 카드 상단에 도구별 브랜드 색상 테두리를 추가해줘.



- 16번째 명령 (복사 버튼 추가):

> 로드맵의 각 명령어 옆에 복사 버튼을 추가해줘.

클릭하면 명령어가 클립보드에 복사되는 기능이 추가된다. JavaScript를 몰라도 "복사 버튼 넣어줘"라고 말하면 된다.



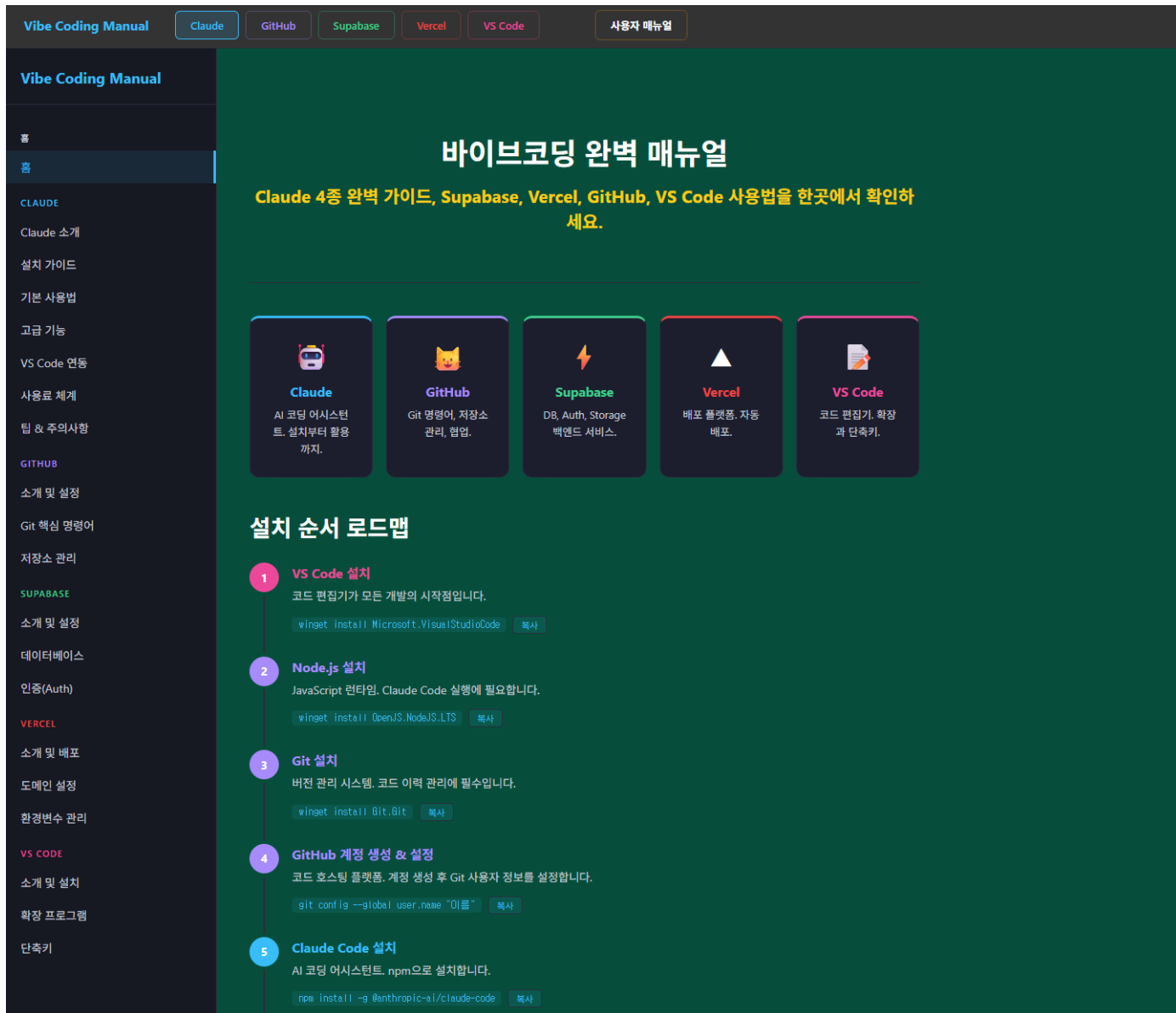
바이브 코딩에서 디자인 수정은 느낀 대로 말하면 된다. CSS를 몰라도, 색상 코드를 몰라도 상관없다. "좀 더 진하게", "좀 더 밝게", "노란색으로" — 이 정도면 충분하다.

#### Phase 4. 사용자 매뉴얼 & 최종 마무리 (명령 17~22)

사이트 사용법을 안내하는 매뉴얼 페이지를 추가하고, 콘텐츠를 확장하여 마무리하는 단계다.

- 17번째 명령 (사용자 매뉴얼 탭 추가):

> 상단 내비게이션에 '사용자 매뉴얼' 버튼을 추가해줘.  
오른쪽에 간격을 두고 배치해줘.



● 18번째 명령 (사용자 매뉴얼 섹션 추가):

> 사용자 매뉴얼 페이지를 만들어줘. 시작하기, 화면 구성, 주요 기능,  
자주 묻는 질문 4개 메뉴로 구성해줘.

아래는 필자가 만든 사용자 매뉴얼 페이지이다.

Vibe Coding ManualClaudeGitHubSupabaseVercelVS Code사용자 매뉴얼

Vibe Coding Manual

홈

랜딩 페이지

Claude

Claude 4종 소개

설치 가이드 (7단계)

기본 사용법

고급 기능 & MCP

VS Code 연동

사용료 체계

토론 절감 방안

Anthropic 이야기

GitHub

소개 및 설정

Git 핵심 명령어

저장소 관리

GitHub Flow

마크다운 문법

GitHub Actions

Supabase

소개 및 설정

데이터베이스

인증(Auth)

스토리지

RLS & 보안

벡터 DB & AI

사용자 매뉴얼

시작하기

Vibe Coding Manual 사이트의 사용 방법을 안내합니다.

Vibe Coding Manual이란?

Vibe Coding Manual은 현대 웹 개발에 필요한 5가지 핵심 도구(Claude Code, GitHub, Supabase, Vercel, VS Code)의 사용법을 한곳에 모은 종합 바이브코딩 완벽 매뉴얼입니다. 초보자부터 실무 개발자까지 단계별로 학습할 수 있도록 구성되어 있습니다.

사용 방법

- 상단 내비게이션 - 원하는 도구(Claude, GitHub, Supabase, Vercel, VS Code)를 클릭하면 해당 가이드의 첫 페이지로 이동합니다.
- 사이드바 메뉴 - 좌측 사이드바에서 각 도구의 세부 항목을 선택하여 원하는 내용으로 바로 이동할 수 있습니다.
- 코드 복사 - 코드 블록 우측 상단의 "복사" 버튼을 클릭하면 명령어를 클립보드에 복사합니다.
- 랜딩 페이지 - 좌측 상단의 "Vibe Coding Manual" 로고를 클릭하면 홈 화면으로 돌아갑니다.

권장 학습 순서

- VS Code 설치 - 코드 편집기가 모든 개발의 시작점입니다. `winget install Microsoft.VisualStudioCode` 복사
- Node.js 설치 - JavaScript 런타임과 npm을 설치합니다. `winget install OpenJS.NodeJS.LTS` 복사
- Git & GitHub 설정 - 버전 관리 환경을 구축합니다. `winget install Git.Git` 복사
- Claude Code 설치 - AI 코딩 어시스턴트를 활용합니다. `npm install -g @anthropic-ai/claude-code` 복사
- Supabase & Vercel - 백엔드와 배포 환경을 설정합니다. `npm install -g vercel` 복사

VIBE CODING MANUAL

Claude Code, GitHub, Supabase, Vercel, VS Code 개발 환경을 위한 종합 매뉴얼입니다. 초보자부터 실무 개발자까지 활용할 수 있는 가이드를 제공합니다.

포함된 가이드

Claude Code - AI 코딩 어시스턴트

GitHub - 버전 관리 & 협업

Supabase - 백엔드 서비스

Vercel - 배포 플랫폼

VS Code - 코드 에디터

관련 링크

Anthropic - anthropic.com

GitHub - github.com

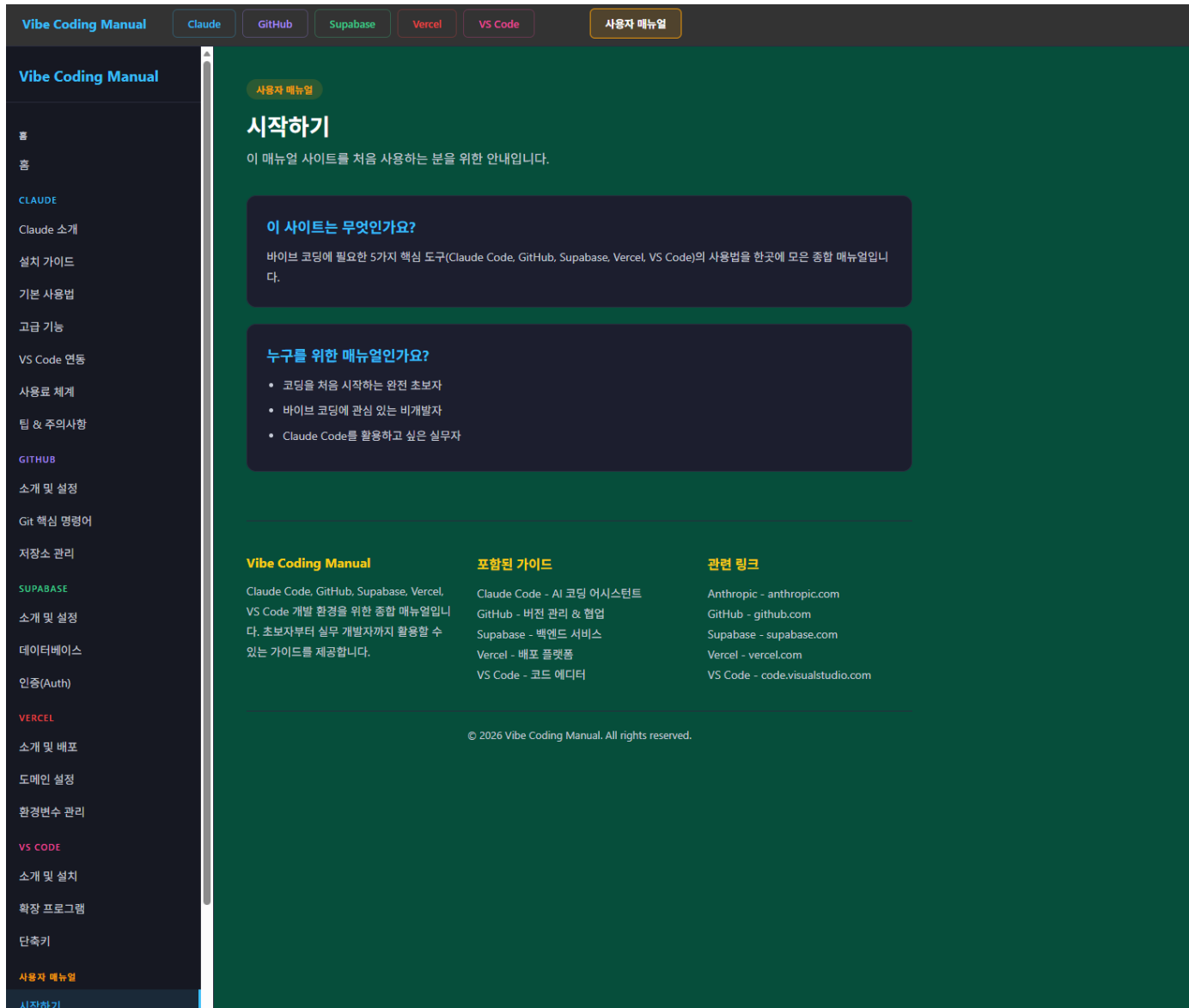
Supabase - supabase.com

Vercel - vercel.com

VS Code - code.visualstudio.com

© 2026 Vibe Coding Manual by Jaiwoo Shim. All rights reserved. | Claude Code / Supabase / Vercel 바이브코딩 완벽 매뉴얼 (010-2397-5734)

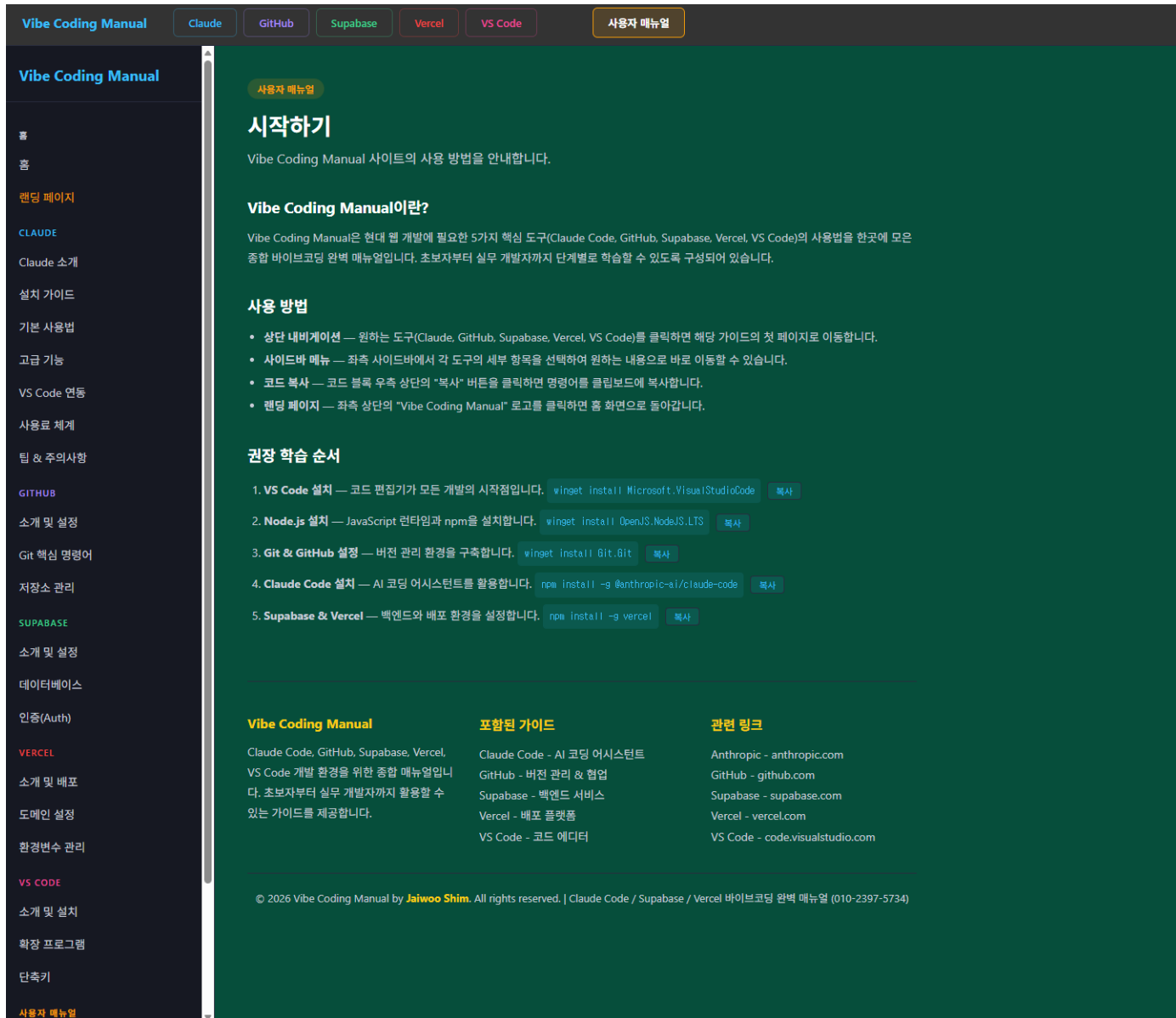
30



● 19번째 명령 (사용자 매뉴얼 내용 수정):

> 사용자 매뉴얼의 '시작하기' 페이지를 수정해줘. Vibe Coding Manual 소개, 사용 방법, 권장 학습 순서를 넣어줘. 권장 학습 순서에는 명령어와 복사 버튼을 포함해줘.

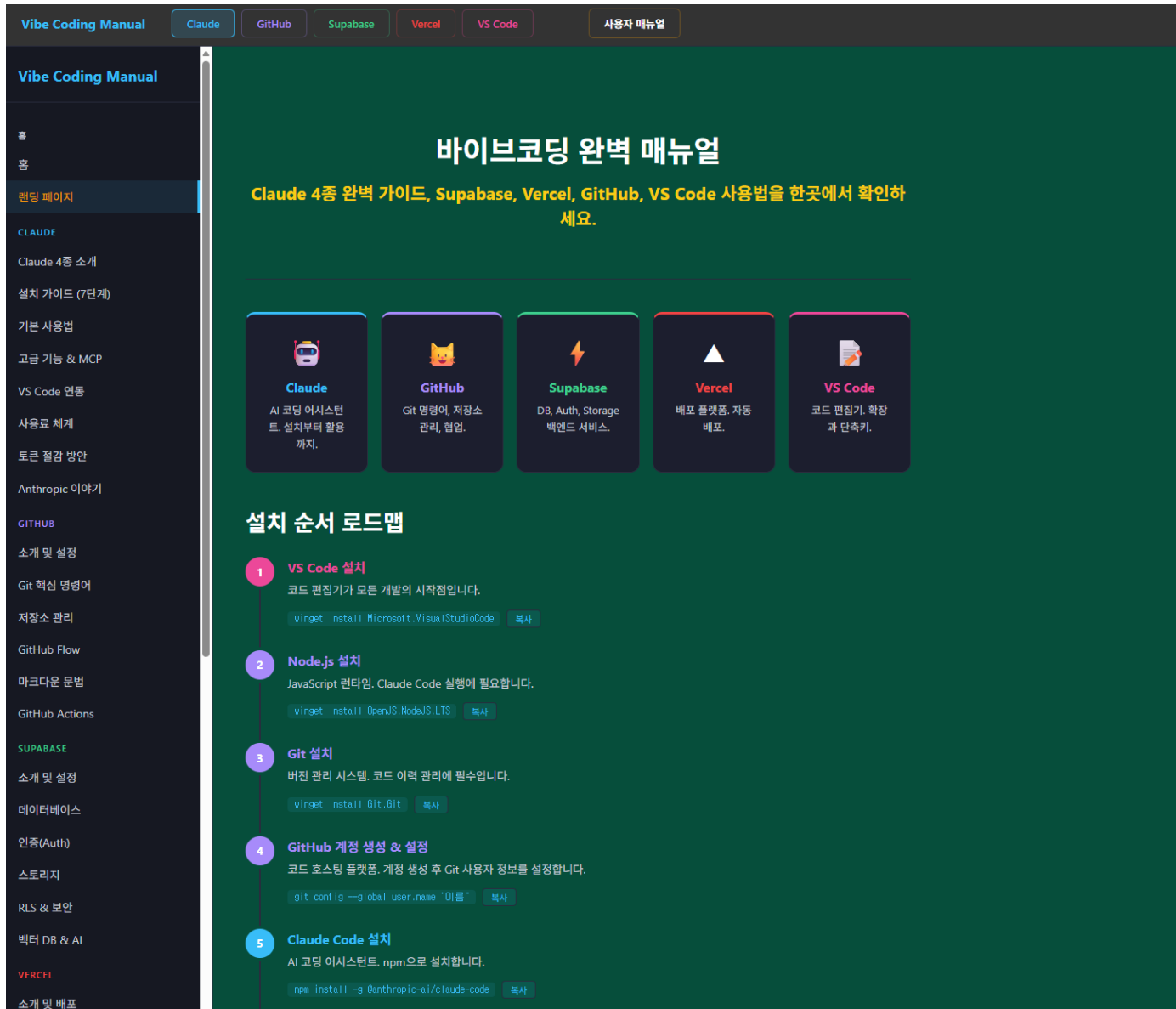
한 번에 완벽하게 나오지 않을 때는 이렇게 구체적으로 수정을 요청한다. "만들고 → 보고 → 고치고"를 반복하는 것이 바이브 코딩의 기본 흐름이다.



## ● 20번째 명령 (사이드바 메뉴 확장):

- > 각 도구의 사이드바 하위 메뉴를 더 추가해줘.
- Claude에 토큰 절감 방안, Anthropic 이야기를 추가하고,
- GitHub에 GitHub Flow, 마크다운 문법, GitHub Actions를 추가하고,
- Supabase에 스토리지, RLS & 보안, 벡터 DB & AI를 추가해줘.



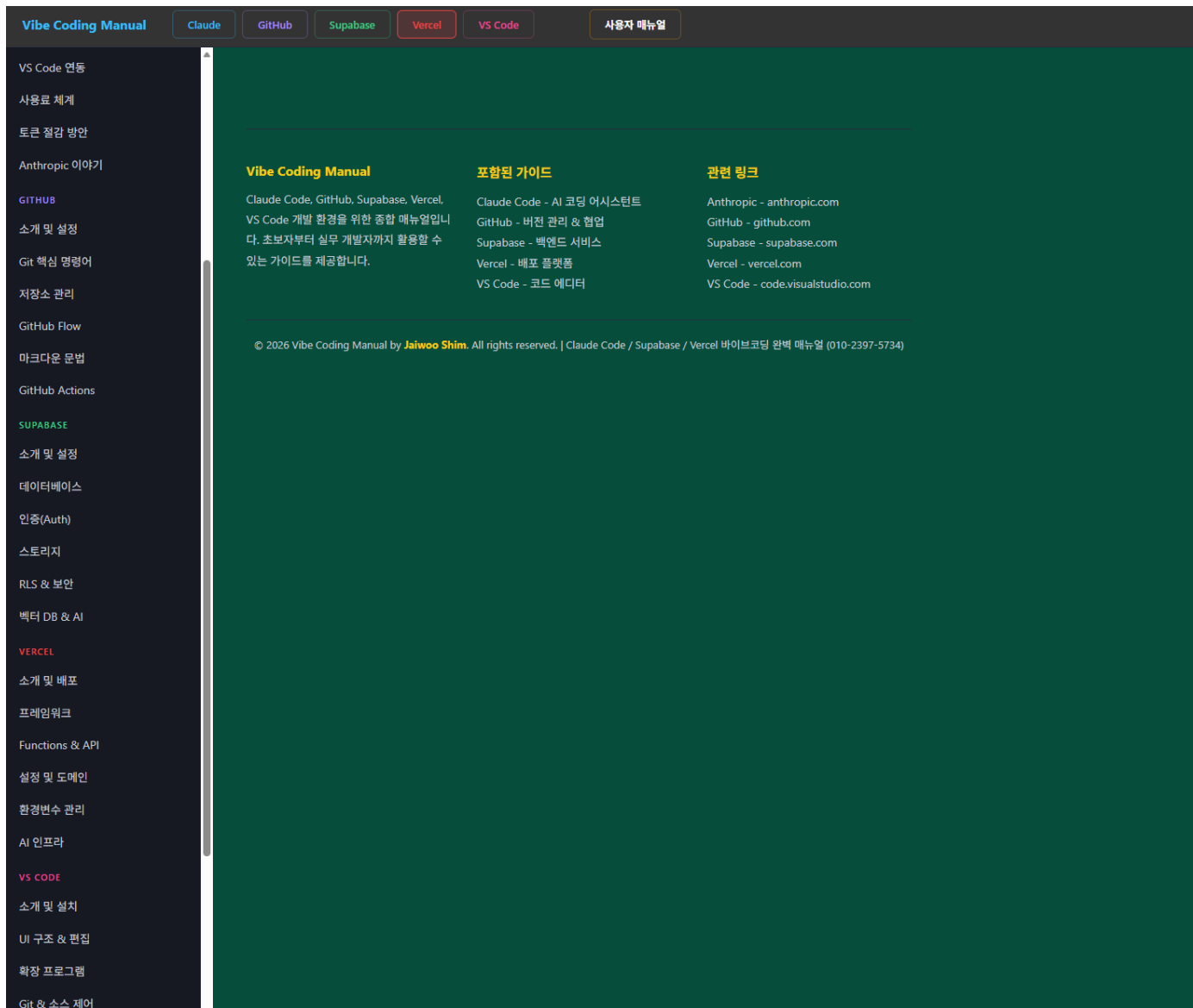


- 21번째 명령 (새 메뉴 콘텐츠 추가):

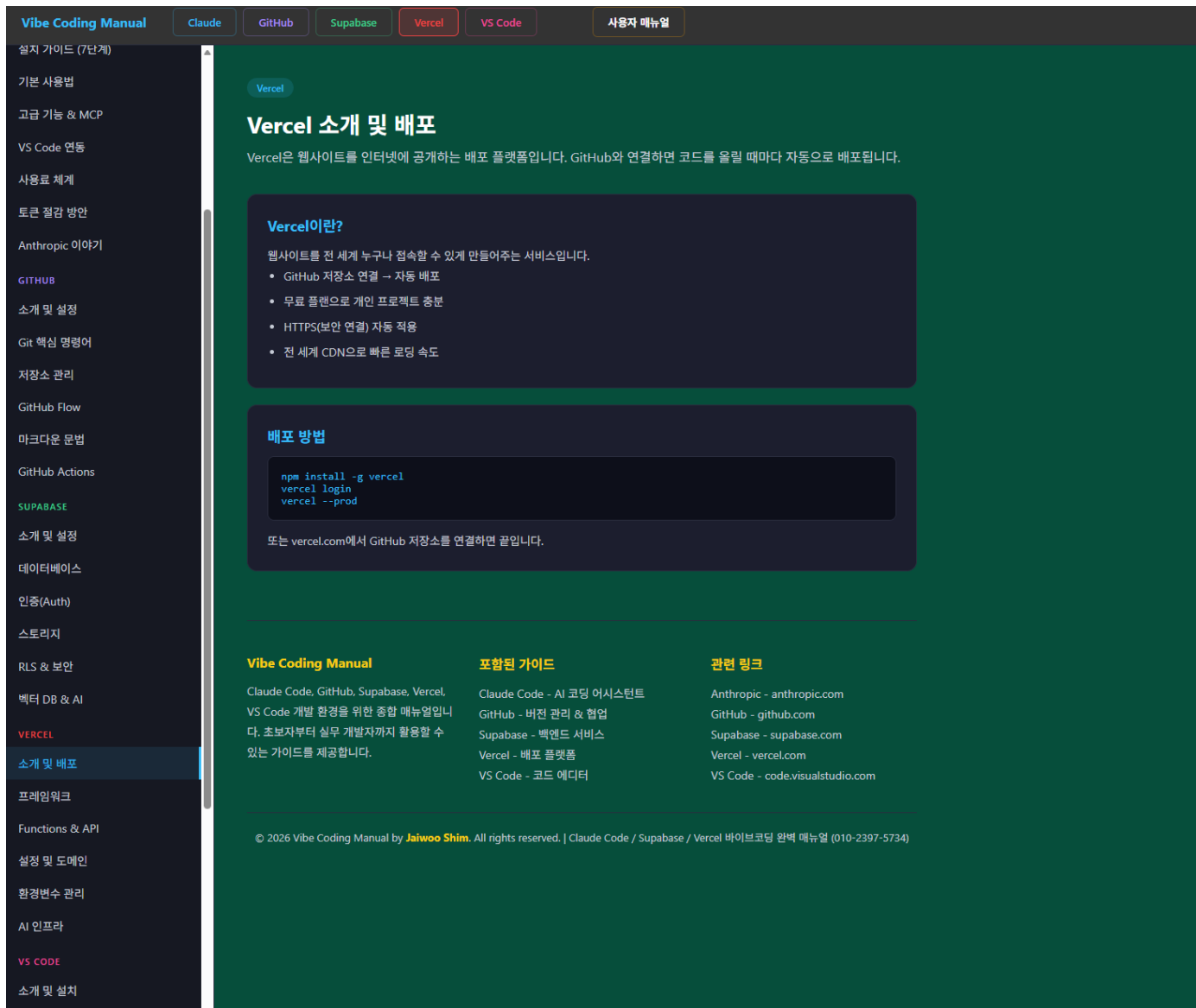
> 추가한 사이드바 메뉴에 맞는 콘텐츠 페이지를 모두 만들어줘.

이 한마디에 AI가 수십 개의 콘텐츠 페이지를 자동으로 생성한다.

이 명령어를 내리기 전에는 지금의 메뉴(Vercel – 소개 및 배포)를 클릭하면 아래처럼 푸터만 있는 빈 페이지가 보인다.

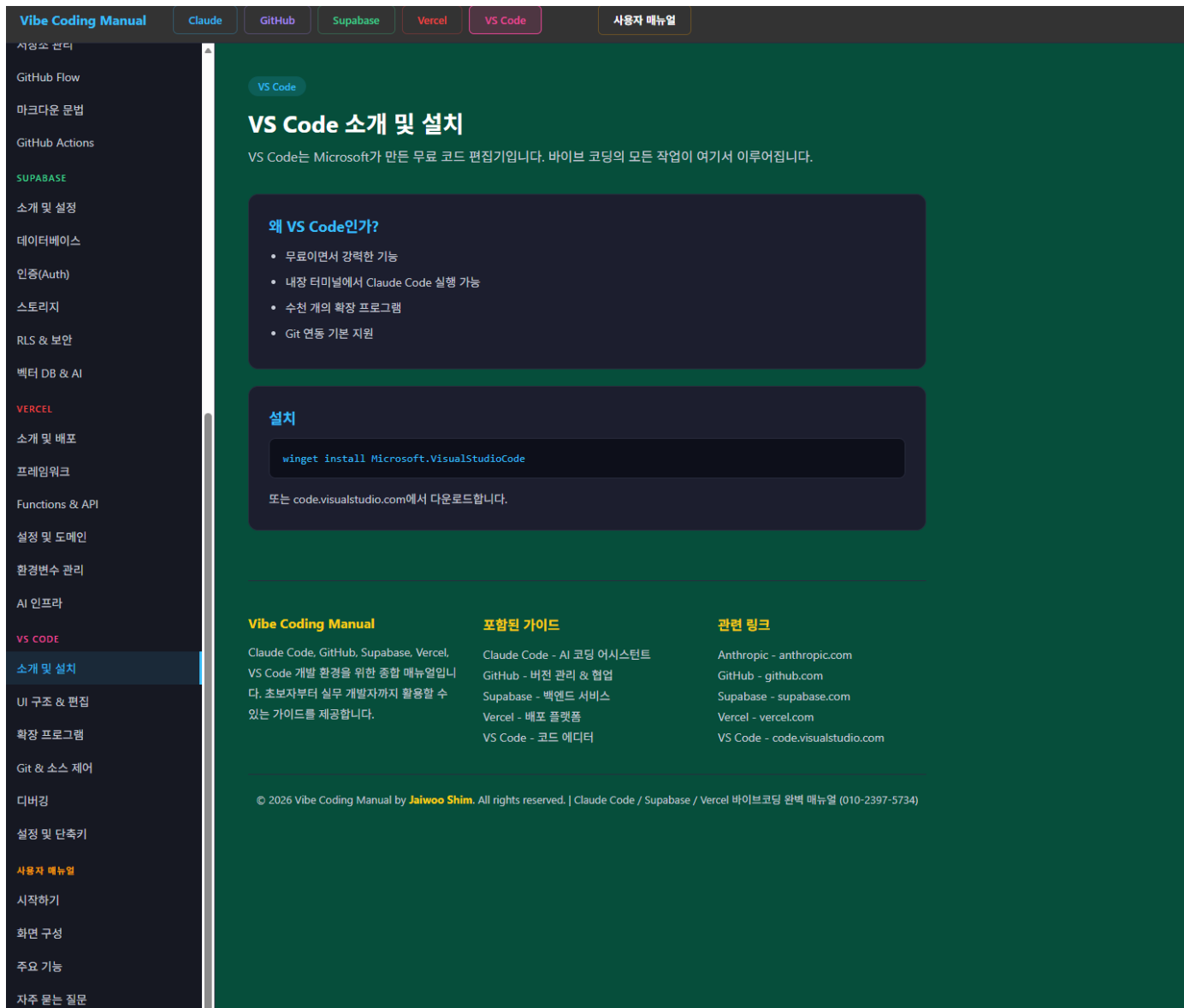


그런데 21번 명령어를 진행하면 아래처럼 새로운 페이지 콘텐츠들이 만들어진다.



아래는 VS Code의 메뉴를 클릭한 화면이다.

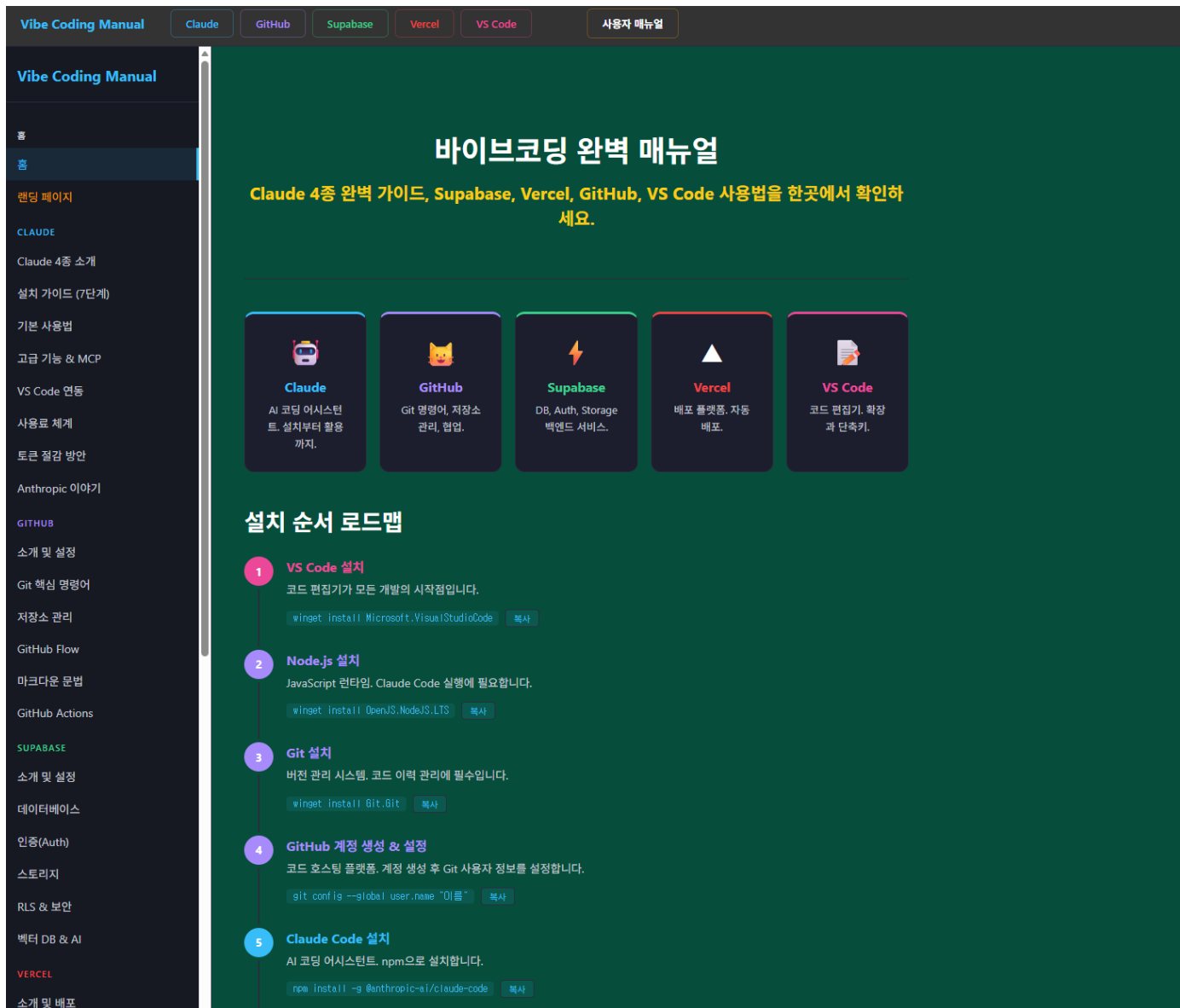
좌측 메뉴에 있는 모든 서브 메뉴의 페이지에 관련 콘텐츠들이 만들어졌다. 여러분이 직접 서브 메뉴를 하나씩 선택하여 확인해 보기 바란다.



지금까지 단지 21번의 명령(말로 내리는)만으로 바이브코딩을 위한 완벽한 매뉴얼이 거의 개발된 셈이다.

● 22번째 명령 (최종 마무리):

> 사이드바 로고와 상단 로고를 클릭하면 홈 화면으로 돌아가게 해줘.  
그리고 푸터 제목을 대문자로 바꿔줘.



22번째 명령을 통해서 다음이 완성됐다.

- 상단 로고 & 사이드바 로고 클릭 → 홈 화면으로 이동
- 푸터 제목 "VIBE CODING MANUAL" 대문자로 변경
- 푸터 "포함된 가이드", "관련 링크" 밑줄 추가

완성! 22번의 명령, 1시간 22분

22번의 명령으로 3,938줄짜리 웹사이트가 완성됐다.

## 실제 소요 시간

단계	내용	소요 시간
Step 1	개발 환경 준비 (VS Code, Node.js, Git, Claude Code 설치)	약 30분
Step 2	프로젝트 폴더 생성 & Claude Code 실행	약 2분
명령 1~6	기본 구조 만들기 (뼈대, 섹션, 탭, 사이드바)	약 15분
명령 7~10	홈 화면 & 레이아웃 (카드, 로드맵, 배경, 푸터)	약 10분
명령 11~16	디자인 & 품질 향상 (브랜딩, 색상, 테두리, 복사 버튼)	약 10분
명령 17~22	사용자 매뉴얼 & 최종 마무리	약 15분
합계		약 1시간 22분

## 우리가 직접 작성한 코드

0줄이다.

3,938줄의 HTML, CSS, JavaScript 코드를 단 한 줄도 직접 타이핑하지 않았다. 우리가 한 것은:

1. 만들고 싶은 것을 **말했다**
2. 결과를 **확인했다**
3. 마음에 안 드는 부분을 **말했다**
4. 다시 결과를 **확인했다**

이것이 바이브 코딩이다.

지금까지 22번의 명령으로 완성된 바이브코딩 완벽 매뉴얼 홈페이지를 요약하면 다음과 같다.

# 챕터 1. 바이브 코딩 명령어 요약표

22번의 명령으로 완성하는 바이브코딩 매뉴얼 사이트

순서	명령 목적	Claude Code에게 내린 명령
1	만들기	Claude 설치와 사용에 대한 완벽 가이드 웹페이지를 만들어줘. index.html 단일 파일로 만들고, 왼쪽에 사이드바 메뉴, 오른쪽에 본문이 있는 레이아웃으로 해줘. 다크 모드 테마로 해줘.
2	수정	본문 글씨가 어두워서 잘 안 보여. 글씨를 충분히 밝게 해서 잘 읽히도록 해줘.
3	섹션 추가	GitHub, Supabase, Vercel, VS Code 섹션도 추가해줘. 각 도구별로 소개와 핵심 기능을 설명해줘.
4	상단 탭	상단에 내비게이션 탭 버튼을 만들어서 Claude, GitHub, Supabase, Vercel, VS Code를 선택할 수 있게 해줘.
5	사이드바 정리	왼쪽 사이드바에 각 도구별 하위 메뉴를 카테고리 나뉘서 정리해줘.
6	사이드바 색상	사이드바의 카테고리 제목에 상단 탭 버튼과 같은 색상을 적용해줘.
7	카드 추가	홈 화면에 각 도구(Claude, GitHub, Supabase, Vercel, VS Code)를 소개하는 카드를 추가해줘.
8	로드맵	홈 화면 카드 아래에 설치 순서 로드맵을 세로 타임라인 형태로 추가해줘.
9	배경색 변경	상단 내비게이션 바 배경색을 진한 회색으로, 페이지 배경색을 진한 녹색으로 바꿔줘.
10	푸터 추가	페이지 맨 아래에 푸터를 추가해줘. 사이트 소개, 포함된 가이드 목록, 관련 링크를 3컬럼으로 보여줘.
11	브랜딩	사이트 이름을 'Vibe Coding Manual'로 바꿔줘. 상단 로고, 사이드바 로고, 푸터 제목 모두 변경해줘.
12	푸터 색상	푸터의 제목들을 노란색으로 바꿔줘.
13	품질 향상	GitHub 공식 매뉴얼( <a href="https://docs.github.com">https://docs.github.com</a> )을 참고해서 GitHub 소개 섹션을 더 정확하고 상세하게 업데이트해줘.
14	제목 변경	홈 화면 제목을 '바이브코딩 완벽 매뉴얼'로 바꾸고, 부제목을 노란색 볼드로 만들어줘.
15	카드 테두리	홈 화면의 각 카드 상단에 도구별 브랜드 색상 테두리를 추가해줘.
16	복사 버튼	로드맵의 각 명령어 옆에 복사 버튼을 추가해줘.
17	매뉴얼 탭	상단 내비게이션에 '사용자 매뉴얼' 버튼을 추가해줘. 오른쪽에 간격을 두고 배치해줘.
18	매뉴얼 내용	사용자 매뉴얼 페이지를 만들어줘. 시작하기, 화면 구성, 주요 기능, 자주 묻는 질문 4개 메뉴로 구성해줘.
19	매뉴얼 수정	사용자 매뉴얼의 '시작하기' 페이지를 수정해줘. Vibe Coding Manual 소개, 사용 방법, 권장 학습 순서를 넣어줘. 권장 학습 순서에는 명령어와 복사 버튼을 포함해줘.
20	메뉴 확장	각 도구의 사이드바 하위 메뉴를 더 추가해줘. Claude에 토큰 절감 방안, Anthropic 이야기를 추가하고, GitHub에 GitHub Flow, 마크다운 문법, GitHub Actions를 추가하고, Supabase에 스토리지, RLS & 보안, 벡터 DB & AI를 추가해줘.
21	콘텐츠	추가한 사이드바 메뉴에 맞는 콘텐츠 페이지를 모두 만들어줘.
22	최종 마무리	사이드바 로고와 상단 로고를 클릭하면 홈 화면으로 돌아가게 해줘. 그리고 푸터 제목을 대문자로 바꿔줘.



## 챕터 1을 마치며 독자들에게 전하는 말

### 1. 코딩을 몰라도 된다

우리는 1시간 동안 코드를 단 한 줄도 직접 타이핑하지 않았다. 한 것이라고는 "이렇게 만들어줘", "이걸 바꿔줘"라고 말한 것뿐이다. 그 결과 1,500줄이 넘는 웹사이트가 완성됐다.

### 2. 한 번에 완벽할 필요 없다

첫 번째 명령으로 만든 결과물은 글씨가 어두워서 잘 보이지 않았다. 푸터가 엉뚱한 위치에 붙기도 했다. 그래도 괜찮다. 바이브 코딩은 "만들고 → 보고 → 고치고"를 반복하는 과정이다. 처음부터 완벽한 결과를 기대하지 마라.

### 3. 작게 말하라

"상단 탭도 만들고, 사이드바도 정리하고, 색상도 바꿔줘"라고 한꺼번에 요청하면 원하는 결과가 나오기 어렵다. **한 번에 하나씩**. 이것이 바이브 코딩에서 가장 중요한 습관이다.

나쁜 예	좋은 예
탭 만들고 색상도 바꾸고 푸터도 추가해줘	상단에 탭 버튼을 만들어줘
사이트 전체를 예쁘게 만들어줘	푸터 제목을 노란색으로 바꿔줘

### 4. 눈으로 확인하고, 마음에 들면 저장하라

명령을 내린 후에는 반드시 브라우저에서 결과를 확인한다. 마음에 들면 바로 "git 커밋해줘"라고 말해서 저장한다. 저장해두면 언제든 과거 상태로 되돌릴 수 있다. 저장하지 않으면 되돌릴 수 없다.

### 5. AI에게 참고 자료를 주면 결과가 달라진다

13번째 명령에서 경험했듯이, 공식 매뉴얼 링크를 함께 제공하면 AI가 훨씬 정확하고 풍부한 콘텐츠를 만들어준다. 사람에게 "보고서 써줘"라고만 하는 것과 "이 자료 참고해서 보고서 써줘"라고 하는 것의 차이와 같다.

### 6. 22번의 명령, 1시간

이 챕터에서 우리가 사용한 시간과 명령 횟수다.

- 22번의 명령 (프롬프트)
- 1시간 소요

- 코드 직접 작성: 0줄
- 완성된 결과: 1,500줄+ 웹사이트

같은 결과물을 몇 년의 경력과 경험을 가진 개발자가 코딩으로 직접 만든다면 며칠은 걸릴 양이다. 바이브 코딩은 "무엇을 만들고 싶은지 아는 사람"이라면 누구나 할 수 있다.

## 7. 바이브 코딩의 본질

바이브 코딩은 코딩 기술이 아니다. **소통 기술**이다.

내가 원하는 것을 명확하게 전달할 수 있다면, AI가 코드를 만들어준다. 중요한 것은 프로그래밍 언어를 아는 것이 아니라, **내가 무엇을 원하는지 아는 것**이다.

## 비개발자가 바이브 코딩 시작 단계에서 멈추는 이유와 해결법

많은 비개발자가 바이브 코딩에 도전하지만 시작 단계에서 포기한다. 그 이유와 해결법을 정리했다.

### 1. 개발 환경 설치에서 막힌다

Node.js, Git, Claude Code 설치 과정에서 에러가 나면 그 에러 메시지 자체를 이해하지 못한다. "PATH가 설정되지 않았습니다", "permission denied" 같은 메시지를 보면 공포감이 먼저 온다.

**해결법:** 설치 과정을 스크린샷과 함께 1단계씩 보여주는 가이드를 따라한다. 에러가 나면 그 에러 메시지를 그대로 복사해서 Claude(웹)에게 물어본다. "이 에러가 뭐야? 어떻게 해결해?"라고 말하면 된다.

### 2. 터미널(명령 프롬프트)이 낯설다

평생 마우스 클릭으로만 컴퓨터를 사용한 사람에게 검은 화면에 글자를 입력하는 것 자체가 두렵다. "내가 뭘 잘못 입력하면 컴퓨터가 망가지지 않을까"라는 걱정이 있다.

**해결법:** 터미널은 "글자로 대화하는 윈도우 탐색기"일 뿐이다. 컴퓨터가 망가지지 않는다. 처음에는 딱 3개만 기억하면 된다.

- cd 폴더명 — 폴더 이동
- ls — 파일 목록 보기
- claude — AI 시작

### 3. 첫 프롬프트를 뭐라고 써야 할지 모른다

"뭘 만들어달라고 해야 하지?" 빈 프롬프트 앞에서 멍해진다. 너무 거창한 것을 요청해야 할 것 같고, 그렇다고 너무 사소한 것을 요청하기엔 아까운 것 같다.

**해결법:** 첫 프롬프트는 단순할수록 좋다. "Hello World 웹페이지 만들어줘"부터 시작한다. 결과가 나오면 "배경색을 파란색으로 바꿔줘"라고 말해본다. 이 두 번의 경험만으로 바이브 코딩의 원리를 체득할 수 있다.

#### 4. 결과물을 확인하는 방법을 모른다

index.html 파일이 만들어졌는데, 이걸 어떻게 브라우저에서 열어보는지 모른다. VS Code의 Live Server 같은 도구가 있다는 것도 모른다.

**해결법:** 가장 쉬운 방법은 index.html 파일을 브라우저 창으로 드래그하는 것이다. 그것만으로 결과를 볼 수 있다. Live Server는 나중에 알아도 된다.

#### 5. 에러가 나면 포기한다

AI가 만든 코드가 한 번에 완벽하게 동작하지 않을 때가 있다. 화면이 깨지거나, 버튼이 작동하지 않거나, 콘솔에 빨간 글씨가 뜬다. 비개발자는 이 순간 "나는 역시 안 되나 보다"라고 생각하고 멈춘다.

**해결법:** 에러는 실패가 아니다. 에러는 "다음 명령"의 재료다. 화면이 깨지면 "화면이 깨졌어. 고쳐줘"라고 말하면 된다. 빨간 에러 메시지가 뜨면 그걸 복사해서 "이 에러 해결해줘"라고 말하면 된다. AI는 에러를 고치는 데 매우 능숙하다.

#### 6. Git을 이해하지 못한다

commit, push, pull, branch 같은 용어가 낯설다. "왜 저장 버튼 하나면 되는 걸 이렇게 복잡하게 하는 거지?"라고 느낀다.

**해결법:** 처음에는 Git의 전체를 이해할 필요가 없다. 딱 하나만 기억한다. **"마음에 들면 커밋해줘라고 말한다."** 그게 전부다. branch, merge, rebase는 나중에 필요할 때 배우면 된다.

#### 7. 혼자 한다

주변에 물어볼 사람이 없다. 유튜브나 블로그를 찾아봐도 자기 상황과 정확히 맞는 답을 찾기 어렵다. 막히는 순간 이 오면 해결할 방법이 없어서 멈춘다.

**해결법:** Claude 웹(claude.ai)이 24시간 질문을 받아주는 선생님이다. "나는 코딩을 전혀 모르는 초보자인데, 지금 이런 상황이야"라고 말하면 눈높이에 맞춰서 설명해준다. 부끄러워할 필요도 없고, 같은 질문을 열 번 해도 된다.

#### 8. 목표가 너무 크다

처음부터 "쇼핑몰 만들어야지", "SNS 만들어야지"라는 거대한 목표를 세운다. 당연히 한 번에 안 되고, 복잡해지면서 포기한다.

**해결법:** 첫 프로젝트는 **한 페이지짜리 웹사이트**로 충분하다. 이 책의 챕터 1처럼 매뉴얼 사이트, 자기소개 페이지, 포트폴리오 페이지부터 시작한다. 한 페이지를 완성하는 경험이 다음 프로젝트의 자신감이 된다.

## 핵심 정리

멈추는 이유	해결 한마디
1. 설치에서 막힘	에러 메시지를 Claude에게 물어본다
2. 터미널이 무섭다	cd, ls, claude 3개만 기억한다
3. 뭘 써야 할지 모름	"Hello World 만들어줘"부터 시작한다
4. 결과 확인 방법 모름	파일을 브라우저에 드래그한다
5. 에러나면 포기	에러 메시지를 복사해서 "고쳐줘"라고 말한다
6. Git이 어렵다	"커밋해줘"만 기억한다
7. 혼자라서 막힘	Claude가 24시간 선생님이다
8. 목표가 너무 크다	한 페이지부터 시작한다
9. 영어 에러 메시지가 두렵다	에러 메시지를 그대로 복사해서 "한국어로 설명해줘"라고 말한다
10. AI가 만든 코드를 믿을 수 없다	브라우저에서 직접 눈으로 확인한다. 코드를 읽을 필요 없다
11. 파일 구조를 이해 못 한다	처음엔 index.html 하나만 쓴다. 폴더 구조는 나중에 배운다
12. 요청이 길어지면 AI가 엉뚱한 결과를 낸다	한 번에 하나만 요청한다. 짧게 말할수록 정확하다
13. 이전에 잘 되던 게 갑자기 안 된다	"방금 전까지 잘 됐는데 안 돼. 되돌려줘"라고 말한다
14. 비용이 얼마나 나올지 불안하다	/cost로 수시로 확인한다. 보통 1시간에 1~3달러 수준이다
15. 디자인 감각이 없다고 느낀다	"깔끔하게 만들어줘", "다크 모드로 해줘" 한마디면 AI가 디자인한다

멈추는 이유	해결 한마디
16. 배포 방법을 모른다	Vercel에 GitHub 연결하면 자동 배포된다. 클릭 3번이면 끝이다
17. 대화가 길어지면 AI가 느려진다	/compact로 대화를 요약하거나, /clear로 새로 시작한다
18. 원하는 걸 말로 표현하기 어렵다	참고할 사이트 URL이나 스크린샷을 함께 보여준다
19. 다른 사람 결과물과 비교하며 좌절한다	모든 결과물은 수십 번의 수정을 거친 것이다. 처음부터 완벽한 건 없다
20. 완성 후 뭘 해야 할지 모른다	GitHub에 올리고, Vercel로 배포하고, 링크를 공유한다. 그게 포트폴리오다
21. CSS가 뭔지 모른다	몰라도 된다. "글씨를 크게 해줘"라고 말하면 AI가 CSS를 작성한다
22. JavaScript가 뭔지 모른다	몰라도 된다. "버튼 누르면 메뉴가 열리게 해줘"라고 말하면 된다
23. 화면이 모바일에서 깨진다	"모바일에서도 잘 보이게 반응형으로 만들어줘"라고 말한다
24. 색상 코드를 모른다	"파란색으로 해줘", "좀 더 진하게"라고 느낌으로 말하면 된다
25. 폰트를 바꾸고 싶은데 방법을 모른다	"깔끔한 고딕 폰트로 바꿔줘"라고 말한다. 폰트 이름을 몰라도 된다
26. 이미지를 넣고 싶은데 방법을 모른다	"여기에 이미지를 넣을 수 있게 해줘"라고 말한다. 경로만 알려주면 된다
27. 수정을 너무 많이 해서 원본을 잃었다	git log로 이전 커밋을 확인하고, "이전 버전으로 되돌려줘"라고 말한다
28. AI가 파일을 잘못 수정했다	"방금 수정한 거 취소해줘"라고 말한다. Ctrl+Z처럼 되돌릴 수 있다
29. 여러 파일이 생기니 헷갈린다	"프로젝트 파일 구조를 설명해줘"라고 말하면 AI가 정리해준다
30. npm install이 뭔지 모른다	앱스토어에서 앱 설치하는 것과 같다. 필요한 도구를 자동으로 가져온다

멈추는 이유	해결 한마디
31. API 키가 뭔지 모른다	서비스를 이용하기 위한 비밀번호 같은 것이다. 절대 남에게 공유하지 않는다
32. .env 파일이 뭔지 모른다	API 키 같은 비밀 정보를 보관하는 금고다. AI에게 "env 파일 만들어 줘"라고 말한다
33. 데이터베이스가 뭔지 모른다	엑셀 표와 같다. 행과 열에 데이터를 저장하는 것이다. Supabase가 대신 관리해준다
34. 서버가 뭔지 모른다	24시간 켜져 있는 컴퓨터다. Vercel이 대신 운영해준다. 직접 관리할 필요 없다
35. 도메인 연결이 어렵다	Vercel 대시보드에서 도메인 입력하고, 도메인 업체에서 DNS 설정만 바꾸면 된다
36. 로그인 기능을 만들고 싶다	Supabase Auth를 쓴다. "Supabase로 이메일 로그인 기능 만들어줘"라고 말한다
37. 결제 기능은 어떻게 만드나	바이브 코딩 초기에는 결제 기능보다 화면과 기능 완성에 집중한다. 결제는 나중에 추가한다
38. 내가 만든 게 보안에 안전한지 모른다	"이 코드에 보안 문제 있는지 확인해줘"라고 AI에게 물어본다
39. 코드가 너무 길어져서 무섭다	코드를 읽을 필요 없다. AI가 관리한다. 결과 화면만 확인하면 된다
40. AI가 매번 다른 결과를 낸다	마음에 드는 결과가 나오면 바로 커밋한다. 저장이 곧 보험이다
41. 다른 AI 도구(ChatGPT 등)와 뭐가 다른지 모른다	Claude Code는 대화만 하는 게 아니라 직접 파일을 만들고 수정한다. 실행력이 다르다
42. 윈도우와 맥에서 명령어가 다를까 걱정된다	Claude Code 명령어는 동일하다. 설치 방법만 약간 다르고, AI에게 물어보면 알려준다
43. 인터넷이 느리면 안 되나	Claude Code는 인터넷이 필요하지만 대용량 다운로드는 없다. 일반 인터넷이면 충분하다
44. AI에게 한국어로 말해도 되나	된다. 한국어로 말하면 한국어로 만들어준다. 코드는 자동으로 영어로

멈추는 이유	해결 한마디
	작성된다
45. 만든 사이트를 수정하고 싶은데 AI 없이는 못 하나	다시 Claude Code를 실행하고 "이 부분 바꿔줘"라고 말하면 된다. 언제든지 수정 가능하다
46. 프로젝트를 중간에 멈췄다가 다시 시작할 수 있나	된다. 같은 폴더에서 claude를 다시 실행하면 이어서 작업할 수 있다
47. 내 코드를 다른 사람이 볼 수 있나	GitHub에서 Private(비공개)로 설정하면 나만 볼 수 있다
48. 저작권 문제는 없나	AI가 생성한 코드의 저작권은 사용자에게 있다. 자유롭게 사용할 수 있다
49. 바이브 코딩으로 돈을 벌 수 있나	웹사이트 제작, 자동화 도구, SaaS 등을 만들어 수익화할 수 있다. 기술보다 아이디어가 중요하다
50. 개발자가 될 필요는 없다	바이브 코딩의 목적은 개발자가 되는 것이 아니라, 내 아이디어를 직접 실현하는 것이다

바이브 코딩에서 가장 어려운 것은 코딩이 아니라, 시작하는 것이다. 그리고 시작한 후에는 멈추지 않는 것이다.