

# 🌐 AI와 함께하는 무료 홈페이지 구축 완벽 가이드

## 스마트폰 하나로 나만의 프로페셔널 웹사이트 만들기

"코딩을 몰라도, 서버가 없어도, 돈을 쓰지 않아도 — AI와 스마트폰만으로 프로페셔널 홈페이지를 구축 할 수 있습니다."

### 📋 목차

1. [Spline으로 3D 애니메이션 만들기](#)
2. [AI로 랜딩 페이지 디자인 & 코드 생성](#)
3. [하위 페이지 구조 구축하기](#)
4. [스마트폰을 웹 서버로 만들기 \(Termux + Ubuntu\)](#)
5. [디자인 다듬기 — 실전 수정 과정](#)
6. [Cloudflare Tunnel로 안전하게 공개하기](#)
7. [실전 프로 팁 — 전문가가 알려주는 추가 노하우](#)

### 📖 이 교재에서 배우는 것

구분	내용	사용 도구
3D 디자인	인터랙티브 3D 애니메이션 제작	Spline
코드 생성	HTML/CSS/Javascript 자동 생성	Antigravity AI
UI 디자인	화면 디자인 & 프로토타이핑	Google Stitch
서버 구축	안드로이드 폰을 웹 서버로	Termux + Ubuntu
배포 자동화	원클릭 배포 시스템	SSH + Shell Script
보안 & 도메인	무료 HTTPS 터널	Cloudflare Tunnel

총 비용: ₩0 (완전 무료)

## 🔧 Chapter 1. Spline으로 3D 애니메이션 임베딩하기

### 1.1 Spline이란?

Spline(스플라인)은 **브라우저에서 바로 작동하는 3D 디자인 도구**입니다. 코딩 없이도 아름다운 3D 오브젝트, 애니메이션, 인터랙션을 만들 수 있습니다.

왜 Spline인가?

- 무료 사용 가능 (기본 플랜)

- 코딩 불필요 — 드래그 앤 드롭 방식
- 웹 임베딩 즉시 가능 — HTML에 바로 삽입
- 인터랙션 지원 — 클릭, 호버, 스크롤 이벤트 연결

## 1.2 Spline 시작하기

### Step 1: 회원 가입

1. [spline.design](#) 접속
2. Google 계정 또는 이메일로 가입
3. 무료 플랜 선택

### Step 2: 3D 씬(Scene) 만들기

1. "New Project" 클릭
2. 기본 도형(큐브, 구체 등)을 배치
3. 재질(Material) 설정: 유리, 금속, 플라스틱 등
4. 조명(Lighting) 추가
5. 카메라 앵글 조정

### Step 3: 인터랙션(Interaction) 추가

이번 프로젝트에서는 3개의 버튼을 만들었습니다:

버튼	기능	연결된 링크
Button 1 (편안한)	Instagram으로 이동	<a href="https://instagram.com/s.rae7400">instagram.com/s.rae7400</a>
Button 2 (24시간)	블로그로 이동	<a href="https://blog.naver.com/youngme777">blog.naver.com/youngme777</a>
Button 3 (전문)	상담 섹션으로 스크롤	#contact

#### Spline에서 버튼 인터랙션 설정:

1. 버튼 오브젝트 선택
2. 우측 패널에서 "Events" 탭 클릭
3. "Mouse Down" 이벤트 추가
4. 이름을 설정 (예: "button 1")

### Step 4: 웹으로 내보내기(Export)

1. 우측 상단 "Export" 클릭
2. "Web Content" 선택
3. "Public URL" 복사

예시 URL:

<https://prod.spline.design/gtg5bodbBqThkVGx/scene.splinicode>

## 1.3 Spline을 HTML에 임베딩하기

## 방법 1: Spline Runtime (우리 프로젝트에서 사용)

```

<!-- HTML에 캔버스 삽입 -->
<canvas id="canvas3d"></canvas>

<!-- Spline Runtime 로드 -->
<script type="module">
    import { Application } from
        'https://unpkg.com/@splinetool/runtime';

    const canvas = document.getElementById('canvas3d');
    const app = new Application(canvas);

    // Spline 씬 로드
    app.load('https://prod.spline.design/여러분의URL/scene.splinecode')
        .then(() => {
            // 클릭 이벤트 처리
            app.addEventListener('mouseDown', (e) => {
                const name = e.target.name;

                if (name.includes('button 1')) {
                    // Instagram으로 이동
                    window.open('https://www.instagram.com/계정', '_blank');
                }
                else if (name.includes('button 2')) {
                    // 블로그로 이동
                    window.open('https://blog.naver.com/계정', '_blank');
                }
            });
        });
</script>

```

## 방법 2: Spline Viewer (간단한 방법)

```

<script type="module"
    src="https://unpkg.com/@splinetool/viewer/build/spline-viewer.js">
</script>

<spline-viewer url="https://prod.spline.design/여러분의URL/scene.splinecode">
</spline-viewer>

```

## 1.4 Spline Built-in 로고 가리기 (CSS 트릭)

Spline 무료 버전은 하단에 "built in spline" 로고가 표시됩니다. 이를 **물결 오버레이**로 자연스럽게 가릴 수 있습니다:

```
<!-- 물결 오버레이 -->
<div class="hero-wave">
  <svg viewBox="0 0 1440 80" preserveAspectRatio="none">
    <path d="M0,70 C200,68 500,62 720,58 C900,54 1100,35 1300,18
      C1380,11 1420,9 1440,8 L1440,80 L0,80 Z"
      fill="#0c0f14" />
    <path d="M0,75 C300,73 600,67 800,60 C1000,52 1150,38 1320,22
      C1400,14 1430,12 1440,11 L1440,80 L0,80 Z"
      fill="#0c0f14" opacity="0.5" />
  </svg>
</div>
```

```
.hero-wave {
  position: absolute;
  bottom: -2px;
  left: 0;
  width: 100%;
  z-index: 10;           /* Spline 위에 표시 */
  pointer-events: none; /* 클릭이 Spline으로 통과 */
}

.hero-wave svg {
  width: 100%;
  height: 80px;         /* 로고만 가릴 최소 높이 */
}
```

💡 **포인트:** `pointer-events: none`을 설정하면 물결 영역을 클릭해도 아래의 Spline 3D가 정상 작동합니다.

## 1.5 모바일 최적화

Spline 3D는 모바일에서 리소스를 많이 소모합니다. 따라서 모바일에서는 가벼운 대체 화면을 보여줍니다:

```
// 모바일 감지 (768px 이하)
if (window.innerWidth > 768) {
  // PC: Spline 3D 로드
  const app = new Application(canvas);
  app.load('https://...');

} else {
  // 모바일: 3D 캔버스 제거, 경량 화면 표시
  canvas.remove();
  console.log('Mobile mode: Spline 3D disabled');
}
```

```
/* 모바일에서 경량 하이로 섹션 표시 */
@media screen and (max-width: 768px) {
```

```

.spline-container {
    display: none !important;
}
.mobile-hero {
    display: flex;
}
}

```

## Chapter 2. AI로 랜딩 페이지 디자인 & 코드 생성

### 2.1 Antigravity AI 코딩 어시스턴트

Antigravity란?

Google DeepMind가 개발한 **AI 페어 프로그래밍 도구**입니다. 자연어 명령만으로 전체 웹사이트의 코드를 생성 할 수 있습니다.

사용 방법

1. **예시 이미지 제공**: 원하는 디자인의 참고 이미지나 스크린샷을 전달
2. **프롬프트 작성**: 원하는 스타일, 색상, 기능을 자연어로 설명
3. **코드 생성**: AI가 HTML/CSS/JavaScript를 자동으로 생성
4. **반복 수정**: 결과물을 확인하고 추가 수정 요청

프롬프트 예시

```

"다크 모드 기반의 뷰티 살롱 랜딩 페이지를 만들어줘.
골드(#c5a059) 액센트 컬러를 사용하고,
히어로 섹션에 Spline 3D를 넣을 거야.
About, Gallery, Services, Contact 섹션이 있어야 해.
폰트는 Playfair Display와 Noto Sans KR을 사용해."

```

생성된 디자인 시스템 (CSS 변수):

```

:root {
    --bg-color: #0c0f14;           /* 메인 배경 */
    --bg-secondary: #141820;        /* 보조 배경 */
    --bg-card: rgba(255, 255, 255, 0.04); /* 카드 배경 */
    --text-color: #e0e0e0;          /* 본문 텍스트 */
    --text-muted: #8a8f9a;          /* 부제목 */
    --gold-color: #c5a059;          /* 골드 포인트 */
    --gold-light: #d4b06a;          /* 밝은 골드 */
    --glow-white: #ffffff;          /* 글로우 화이트 */
    --border-subtle: rgba(255, 255, 255, 0.06);
}

```

## 2.2 Google Stitch로 UI 디자인하기 (추가 방법)

예시 이미지 기반 방식이 만족스럽지 않을 때, **Google Stitch**를 활용하면 더 정교한 디자인이 가능합니다.

Stitch란?

Google의 AI 기반 **UI 디자인 & 프로토타이핑** 도구입니다.

Stitch 사용 워크플로우

### Step 1: 프로젝트 생성

Stitch에서 새 프로젝트 생성  
→ 타이틀: "Seorae Beauty Homepage"

### Step 2: 텍스트 프롬프트로 화면 생성

"뷰티 살롱 웹사이트의 하이로 셕션을 디자인해줘.  
다크 배경(#0c0f14)에 골드(#c5a059) 액센트.  
중앙에 3D 오브젝트 공간, 우아하고 고급스러운 느낌."

### Step 3: 변형(Variants) 생성

- 하나의 디자인에서 여러 변형을 AI가 자동 생성
- 색상, 레이아웃, 타이포그래피 등을 변경한 버전 비교

### Step 4: 화면 편집 & 수정

- 특정 화면을 선택하고 편집 프롬프트로 수정
- "이 버튼을 더 크게 만들고 골드 그라데이션을 적용해"

Stitch의 장점

항목	예시 이미지 방식	Stitch 방식
정확도	이미지에 의존	AI 생성으로 정교함
수정	재생성 필요	부분 수정 가능
변형	수동	자동 변형 생성
반응형	별도 작업	디바이스별 자동

💡 **실전 팁:** Stitch에서 디자인 → 스크린샷 → Antigravity에 전달하면 가장 정확한 코드를 얻을 수 있습니다.

## 2.3 AI가 생성한 주요 코드 구조

## 전체 파일 구조

```
website/
├── index.html      ← 메인 (HTML + CSS + JS 통합, 1260+ 라인)
├── logo.png        ← 로고 이미지
└── images/
    └── about-image.png ← About 섹션 이미지
```

## HTML 기본 구조 (Single Page Application)

```
<!DOCTYPE html>
<html lang="ko">
<head>
    <!-- 메타 태그 & 폰트 -->
    <style>
        /* 전체 CSS (600+ 라인) */
    </style>
</head>
<body>
    <header><!-- 네비게이션 --></header>

    <section id="home" class="hero">
        <!-- Spline 3D 히어로 -->
    </section>

    <section id="about">
        <!-- 회사 소개 -->
    </section>

    <section id="gallery">
        <!-- 갤러리 + 라이트박스 -->
    </section>

    <section id="services">
        <!-- 서비스 목록 -->
    </section>

    <section id="contact">
        <!-- 연락처 + 상담 품 -->
    </section>

    <footer><!-- 푸터 --></footer>

    <script>/* 인터랙션 JS */</script>
    <script type="module">/* Spline JS */</script>
</body>
</html>
```

# Chapter 3. 하위 페이지 구조 구축하기

## 3.1 싱글 페이지 아키텍처 (SPA)

이 프로젝트는 **싱글 페이지 애플리케이션(SPA)** 방식을 사용합니다. 모든 콘텐츠가 하나의 HTML 파일에 있고, 네비게이션은 **앵커 스크롤**로 처리합니다.

### 네비게이션 구조

```
<header>
    <!-- 로고 -->
    <div class="logo-container">
        <div class="logo-main">
            
        </div>
        <div class="logo-sub">SEORAE BEAUTY</div>
    </div>

    <!-- 메뉴 -->
    <ul class="nav-links">
        <li><a href="#home">Home</a></li>
        <li><a href="#about">About</a></li>
        <li><a href="#services">Services</a></li>
        <li><a href="#contact">Contact</a></li>
    </ul>

    <!-- 모바일 햄버거 -->
    <div class="burger">
        <div class="line1"></div>
        <div class="line2"></div>
        <div class="line3"></div>
    </div>
</header>
```

### 스무스 스크롤 구현

```
document.querySelectorAll('a[href^="#"]').forEach(anchor => {
    anchor.addEventListener('click', function (e) {
        e.preventDefault();
        nav.classList.remove('nav-active'); // 모바일 메뉴 닫기

        document.querySelector(this.getAttribute('href'))
            .scrollIntoView({ behavior: 'smooth' });
    });
});
```

## 3.2 각 섹션 상세 구현

## Gallery 섹션 — 탭 필터 + 라이트박스

```
<!-- 카테고리 탭 -->
<div class="gallery-tabs">
    <button class="gallery-tab active" data-filter="all">전체</button>
    <button class="gallery-tab" data-filter="education">교육 과정</button>
    <button class="gallery-tab" data-filter="studio">뷰티 스튜디오</button>
    <button class="gallery-tab" data-filter="seminar">세미나</button>
</div>

<!-- 갤러리 그리드 -->
<div class="gallery-grid">
    <div class="gallery-item" data-category="education">
        
        <div class="gallery-overlay"><h4>전문가 교육 과정</h4></div>
    </div>
    <!-- 더 많은 아이템... -->
</div>
```

### 탭 필터링 JavaScript:

```
const tabs = document.querySelectorAll('.gallery-tab');
const items = document.querySelectorAll('.gallery-item');

tabs.forEach(tab => {
    tab.addEventListener('click', () => {
        tabs.forEach(t => t.classList.remove('active'));
        tab.classList.add('active');

        const filter = tab.dataset.filter;
        items.forEach(item => {
            if (filter === 'all' || item.dataset.category === filter) {
                item.classList.remove('hidden');
            } else {
                item.classList.add('hidden');
            }
        });
    });
});
```

## Contact 섹션 — 상담 신청 폼

무료 이메일 전송 서비스 **FormSubmit**을 활용합니다:

```
<form action="https://formsubmit.co/이메일주소" method="POST">
    <input type="hidden" name="_subject"
           value="새 상담 신청">
    <input type="hidden" name="_captcha" value="false">
```

```
<input type="text" name="name" placeholder="이름" required>
<input type="tel" name="phone" placeholder="전화번호" required>
<input type="email" name="email" placeholder="이메일" required>

<select name="course" required>
    <option value="semi-permanent">반영구 화장</option>
    <option value="tattoo">타투 아트</option>
    <option value="startup">창업 컨설팅</option>
</select>

<button type="submit">상담 신청하기</button>
</form>
```

💡 **포인트:** FormSubmit.co는 완전 무료이며, 백엔드 서버 없이 이메일로 폼 데이터를 받을 수 있습니다.

### 3.3 반응형 디자인 (모바일 대응)

```
@media screen and (max-width: 768px) {
    /* 모바일 네비게이션 */
    .nav-links {
        position: absolute;
        right: 0;
        top: 70px;
        background-color: rgba(12, 15, 20, 0.98);
        flex-direction: column;
        width: 100%;
        transform: translateY(-150%);
        transition: transform 0.5s ease-in;
    }

    .nav-links.nav-active {
        transform: translateY(0%);
    }

    /* 햄버거 표시 */
    .burger { display: block; }

    /* Spline 대신 모바일 히어로 */
    .spline-container { display: none !important; }
    .mobile-hero { display: flex; }
}
```

---

## 🔨 Chapter 4. 스마트폰을 웹 서버로 만들기

---

### 4.1 왜 스마트폰 서버인가?

항목	호스팅 서비스	스마트폰 서버
비용	월 \$5~\$20+	완전 무료
성능	공유 서버	전용 자원
학습	관리 패널만	리눅스 실전 학습
자유도	제한적	완전한 제어
24시간	<input checked="" type="checkbox"/>	충전 필요

## 4.2 필요한 준비물

- 📱 안드로이드 스마트폰 (안 쓰는 구형 폰 OK)
- 📶 Wi-Fi 연결 (같은 공유기)
- 💻 PC (코드 작성용)
- ⚡ 충전기 (24시간 운영 시)

## 4.3 Termux 설치

### Step 1: Termux 앱 설치

Google Play가 아닌 **F-Droid**에서 설치합니다:

1. F-Droid 앱 설치 ([f-droid.org](https://f-droid.org))
2. F-Droid에서 "Termux" 검색
3. Termux 설치

**⚠ 주의:** Google Play의 Termux는 오래된 버전이므로 반드시 F-Droid에서 설치하세요.

### Step 2: 기본 패키지 업데이트

```
# Termux에서 실행
pkg update && pkg upgrade -y
```

### Step 3: 필수 패키지 설치

```
# 필수 도구 설치
pkg install openssh -y          # SSH 서버
pkg install proot-distro -y      # Linux 배포판 설치 도구
pkg install wget curl -y         # 파일 다운로드 도구
```

## 4.4 Ubuntu 설치 (proot-distro)

```
# Ubuntu 설치
proot-distro install ubuntu
```

```
# Ubuntu에 접속  
sudo su -  
apt distro login ubuntu
```

Ubuntu 내부에서 Nginx 설치:

```
# Ubuntu 내부에서 실행  
apt update && apt upgrade -y  
apt install nginx -y  
  
# 웹사이트 디렉토리 생성  
mkdir -p /var/www/website
```

## 4.5 SSH 서버 설정 (원격 접속)

PC에서 스마트폰에 원격으로 접속하기 위해 SSH를 설정합니다:

스마트폰(Termux)에서:

```
# SSH 서버 시작  
sshd  
  
# 비밀번호 설정  
passwd  
  
# 현재 IP 확인  
ifconfig  
# → 예: 172.30.1.84
```

PC에서 접속 테스트:

```
# PC에서 스마트폰에 SSH 접속  
ssh -p 8022 u0_a341@172.30.1.84
```

## 4.6 SSH 키 기반 인증 (비밀번호 없이 접속)

매번 비밀번호를 입력하지 않도록 SSH 키를 설정합니다:

PC에서:

```
# SSH 키 생성  
ssh-keygen -t rsa -b 4096 -f termux_key
```

```
# 공개키를 스마트폰에 복사  
scp -P 8022 termux_key.pub u0_a341@172.30.1.84:~/ssh/authorized_keys
```

이후 접속:

```
# 비밀번호 없이 접속 가능  
ssh -p 8022 -i termux_key u0_a341@172.30.1.84
```

## 4.7 Nginx 웹 서버 설정

사이트 설정 파일 생성:

```
# /etc/nginx/sites-available/website

server {
    listen 8083;
    server_name localhost;
    root /var/www/website;
    index index.html;

    # Gzip 압축 (속도 향상)
    gzip on;
    gzip_vary on;
    gzip_min_length 256;
    gzip_comp_level 6;
    gzip_types
        text/plain text/css text/javascript
        application/javascript application/json
        image/svg+xml font/woff font/woff2;

    # 이미지 캐싱 (30일)
    location ~* \.(jpg|jpeg|png|gif|ico|svg|webp)$ {
        expires 30d;
        add_header Cache-Control "public, immutable";
    }

    # CSS/JS 캐싱 (7일)
    location ~* \.(css|js|woff|woff2|ttf)$ {
        expires 7d;
        add_header Cache-Control "public";
    }

    location / {
        try_files $uri $uri/ =404;
    }
}
```

## Nginx 시작:

```
# 설정 테스트
nginx -t

# Nginx 시작
service nginx start

# 또는 재시작
service nginx reload
```

## 4.8 자동 배포 스크립트

PC에서 코드를 수정하고 한 줄 명령어로 배포하는 스크립트입니다:

deploy\_and\_optimize.sh:

```
#!/bin/bash
echo "==== Deploying index.html to /var/www/website ==="

# Termux → Ubuntu로 파일 복사
prroot-distro login ubuntu -- bash -c '
    cp /data/data/com.termux/files/home/website/index.html \
        /var/www/website/index.html
    cp /data/data/com.termux/files/home/website/logo.png \
        /var/www/website/logo.png
    chown www-data:www-data /var/www/website/*'
'

echo "==== Optimizing Nginx ==="
prroot-distro login ubuntu -- bash -c '
    nginx -t && service nginx reload
    echo "Nginx optimized successfully!"
'

echo "==== Done ==="
```

PC에서 원클릭 배포:

```
# Step 1: 파일 전송 (PC → 스마트폰)
scp -P 8022 index.html u0_a341@172.30.1.84:~/website/

# Step 2: 배포 스크립트 실행
ssh -p 8022 u0_a341@172.30.1.84 "./deploy_and_optimize.sh"
```

**💡 결과:** PC에서 코드 수정 → 2줄 명령어 → 실시간 배포 완료!

## 4.9 접속 테스트

접속 환경	URL
같은 Wi-Fi (내부)	<a href="http://172.30.1.84:8083">http://172.30.1.84:8083</a>
외부 (Cloudflare)	<a href="https://your-tunnel.trycloudflare.com">https://your-tunnel.trycloudflare.com</a>

# 🔨 Chapter 5. 디자인 다듬기 — 실전 수정 과정

## 5.1 헤더 디자인 반복 수정

디자인 작업에서 가장 중요한 것은 \*\*반복 수정(Iteration)\*\*입니다. 실전에서 겪은 수정 과정을 공유합니다.

### 수정 이력

단계	수정 내용	결과
1차	로고 + 텍스트 세로 배치	텍스트가 너무 멀리 떨어짐
2차	가로 배치로 변경	글자 크기가 안 맞음
3차	글자 크기 조절	대문자가 위로 빠져나옴
4차	로고 크기를 키움	여전히 baseline 불일치
최종	세로 배치 + <b>letter-spacing</b>	<input checked="" type="checkbox"/> 완벽한 균형

최종 결과 코드:

```
<div class="logo-container"
    style="display:flex; flex-direction:column;
        align-items:center; gap:5px;">

    <div class="logo-main">
        
    </div>

    <div class="logo-sub"
        style="font-size:1.0rem;
            letter-spacing:0.35em;
            color:#ffffff;
            text-shadow: 0 0 5px #FFD700;">
        SEORAE BEAUTY
    </div>
</div>
```

## 배운 점

1. 완벽을 처음부터 기대하지 말 것 — 디자인은 반복 수정의 과정
2. 작은 변화를 하나씩 — 한 번에 여러 속성을 바꾸면 원인을 찾기 어려움
3. AI에게 구체적으로 — "좀 더 예쁘게"보다 "font-size를 1rem으로, padding을 0으로"

## 5.2 인터랙티브 사운드 효과

버튼 클릭 시 맞춤 사운드를 재생하도록 **Web Audio API**를 사용했습니다:

```
function playTone(type) {
  const ctx = new AudioContext();
  const osc = ctx.createOscillator();
  const gain = ctx.createGain();

  osc.connect(gain);
  gain.connect(ctx.destination);
  const now = ctx.currentTime;

  if (type === 'bright') {
    // 맑고 영롱한 크리스탈 소리
    osc.type = 'sine';
    osc.frequency.setValueAtTime(1174.66, now); // D6
    gain.gain.linearRampToValueAtTime(0.2, now + 0.05);
    gain.gain.exponentialRampToValueAtTime(0.001, now + 2.0);
    osc.start(now);
    osc.stop(now + 2.0);
  }
  else if (type === 'water1') {
    // 맑은 물방울 '또롱~' 소리
    osc.type = 'sine';
    osc.frequency.setValueAtTime(800, now);
    osc.frequency.exponentialRampToValueAtTime(300, now + 0.2);
    // ... (피치 하강으로 물방울 효과)
  }

  setTimeout(() => ctx.close(), 2500); // 메모리 정리
}
```

## 5.3 지연 네비게이션 (Delayed Navigation)

버튼 클릭 → 사운드 재생 → **3.5초 후** 페이지 이동:

```
app.addEventListener('mousedown', (e) => {
  if (name.includes('button 1')) {
    playTone('bright'); // 사운드 먼저

    setTimeout(() => { // 3.5초 후 이동
      window.open('https://instagram.com/...', '_blank');
    }, 3500);
```

```

    }
});
```

## 5.4 스크롤 시 헤더 변환

```

window.addEventListener('scroll', function () {
    if (window.scrollY > 50) {
        header.classList.add('scrolled');
    } else {
        header.classList.remove('scrolled');
    }
});
```

```

/* 기본 상태: 투명 */
header {
    background: rgba(12, 15, 20, 0);
}

/* 스크롤 시: 배경 + 블러 + 그림자 */
header.scrolled {
    background: rgba(20, 24, 32, 0.95);
    backdrop-filter: blur(15px);
    box-shadow: 0 2px 10px rgba(0, 0, 0, 0.3);
}
```

## ❖ Chapter 6. Cloudflare Tunnel로 안전하게 공개하기

### 6.1 왜 포트 포워딩 대신 Cloudflare인가?

기존 방식: 공유기 포트 포워딩

```
인터넷 → 공유기(포트 오픈) → 스마트폰
```

#### ✖ 문제점:

- 보안 취약 (IP 노출)
- 공유기 설정이 복잡
- ISP가 차단할 수 있음
- HTTPS 인증서 별도 필요

새로운 방식: Cloudflare Tunnel

인터넷 → Cloudflare(보안) → 암호화 터널 → 스마트폰

장점:

- IP 숨김 (보안)
- 무료 **HTTPS** 자동 적용
- **DDoS 보호** 무료 제공
- 공유기 설정 불필요

## 6.2 Cloudflare Tunnel 설치

Termux에서:

```
# cloudfaired 설치 (ARM 버전)
wget
https://github.com.cloudflare/cloudfaired/releases/latest/download/cloudfaired-
linux-arm64
chmod +x cloudfaired-linux-arm64
mv cloudfaired-linux-arm64 $PREFIX/bin/cloudfaired
```

## 6.3 Quick Tunnel 사용 (가장 간단)

계정 없이 바로 사용할 수 있는 방법:

```
# Nginx가 8083 포트에서 실행 중이라면:
cloudfaired tunnel --url http://localhost:8083
```

출력 예시:

```
INF +-----+
INF | Your quick Tunnel URL:      |
INF | https://mysimon-fraser-romantic-
INF | joshua.trycloudflare.com      |
INF +-----+
```

이 URL을 전 세계 누구나 접속할 수 있습니다! 🌐

## 6.4 영구 도메인 연결 (Named Tunnel)

무료 도메인을 연결하려면:

Step 1: Cloudflare 계정 생성

- dash.cloudflare.com에서 무료 가입

## Step 2: 터널 생성

```
# 로그인
cloudflared tunnel login

# 터널 생성
cloudflared tunnel create my-beauty-site

# DNS 레코드 연결
cloudflared tunnel route dns my-beauty-site www.mydomain.com

# 터널 실행
cloudflared tunnel run my-beauty-site
```

## 6.5 24시간 운영 팁

```
# Termux에서 백그라운드 실행
nohup cloudflared tunnel --url http://localhost:8083 &

# Termux 알림에서 "Acquire wakelock" 활성화
# → 화면이 꺼져도 서버가 계속 실행됩니다
```

# 💡 Chapter 7. 실전 프로 팁 — 전문가 노하우

## 7.1 성능 최적화

### 이미지 최적화

```
<!-- Lazy Loading: 화면에 보일 때만 로드 -->
![설명](image.jpg)
```

### GPU 가속

```
/* 부드러운 스크롤을 위한 GPU 가속 */
section, .gallery-item, .service-card {
    will-change: auto;
    transform: translateZ(0);
}
```

### 폰트 최적화

```
<!-- DNS 사전 연결 -->
<link rel="preconnect" href="https://fonts.googleapis.com">
<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
<link rel="preconnect" href="https://unpkg.com">
```

## 7.2 SEO 기본 설정

```
<head>
  <title>서래뷰티 | SEORAE BEAUTY</title>
  <meta charset="UTF-8">
  <meta name="viewport"
        content="width=device-width, initial-scale=1.0">
  <meta name="description"
        content="서래뷰티 - 프리미엄 반영구 화장 & 타투 아카데미">
</head>
```

## 7.3 무료 리소스 총정리

도구	용도	URL
Spline	3D 디자인	spline.design
Unsplash	무료 이미지	unsplash.com
Google Fonts	무료 폰트	fonts.google.com
FormSubmit	무료 폼 전송	formsubmit.co
Cloudflare	무료 터널/CDN	cloudflare.com
Termux	안드로이드 터미널	f-droid.org
Let's Encrypt	무료 SSL	letsencrypt.org

## 7.4 보안 체크리스트

- SSH 키 기반 인증 사용 (비밀번호 비활성화)
- Cloudflare Tunnel 사용 (IP 숨김)
- Nginx 보안 헤더 설정
- 불필요한 포트 닫기
- 정기적 패키지 업데이트

## 7.5 문제 해결 가이드 (FAQ)

Q: Spline 3D가 모바일에서 너무 느려요

A: 모바일에서는 3D를 비활성화하고 경량 대체 화면을 보여주세요 (Chapter 1.5 참고)

Q: SSH 접속이 안 돼요

**A:** Termux에서 `sshd`가 실행 중인지 확인하세요. `whoami`로 사용자명, `ifconfig`로 IP를 확인하세요.

**Q:** Nginx 설정 변경이 반영이 안 돼요

**A:** `nginx -t`로 문법 확인 후 `service nginx reload`를 실행하세요.

**Q:** Cloudflare Tunnel URL이 바뀌어요

**A:** Quick Tunnel은 재시작마다 URL이 바뀝니다. 영구 URL이 필요하면 Named Tunnel을 사용하세요 (Chapter 6.4 참고).

---

## 마무리

### 우리가 구축한 것

이 교재를 통해 여러분은 완전 무료로 다음을 구축했습니다:

- 3D 인터랙티브 히어로 셕션** (Spline)
- AI 생성 프로페셔널 랜딩 페이지** (Antigravity)
- 갤러리/서비스/상담 폼** (HTML/CSS/JS)
- 스마트폰 웹 서버** (Termux + Ubuntu + Nginx)
- 자동 배포 파이프라인** (SCP + SSH + Shell Script)
- 글로벌 HTTPS 접속** (Cloudflare Tunnel)

### 다음 단계

-  **커스텀 도메인 연결** (예: seorae-beauty.com)
-  **PWA (Progressive Web App) 전환**
-  **Google Analytics 연동**
-  **결제 시스템 통합** (토스페이먼츠 등)
-  **뉴스레터 구독 기능**

"여러분도 할 수 있습니다. AI와 함께라면, 코딩 경험이 없어도 프로페셔널 웹사이트를 만들 수 있습니다."

---