

## 2025-2 웹퍼블리싱 기말 프로젝트 안내

12월 7일(일) 23:59 까지 LMS에 제출해 주세요!

제출해 주실 것은,

### 1. 웹페이지 결과물

- 완성된 페이지 폴더를 압축(zip)하거나
- GitHub Pages, Netlify, JSFiddle 등 퍼블리싱된 웹 주소 제출

### 2. 웹페이지 설명서(A4 1장 이내)

- 워드, 한글 등 자유 양식
- 아래 내용을 간단히 정리

#### <설명서에 포함될 내용>

---

1. 내 웹 페이지의 주제와 목적
2. 어떤 기술을 사용했나요?
3. 프로젝트 품질 기준을 활용한 코드 설명
4. AI 도구 활용 사례

의 내용과 프로젝트 품질 기준의 코드, 용어를 활용해서, 내가 만든 웹페이지를 설명해 주세요!

---

#### <기말과제 평가 기준>

---

##### [과제 점수 20점]

- 프로젝트 품질 기준 중 5가지 이상이 자연스럽게 구현되었는가? (20/18/16/14/12점)
- 단순히 코드 유무가 아니라, 실제 동작 여부와 구현의 자연스러움을 중심으로 평가합니다.
- **정량적으로 확인 가능한 항목(L1~A4)을 중심으로 평가**

##### [팀프로젝트 점수 10점]

- 완성된 웹페이지는 각 기능들을 일관된 목적으로 연결하여 완성도 높은 결과물을 보여주는가? (5/4/3/2/1점)
  - 웹페이지 설명서는 최종 결과물인 웹페이지의 기능과 특징을 정확히 설명하고 있는가? (5/4/3/2/1점)
  - **결과물의 완성도와 설명서에 담긴 정성 등을 포함해 평가**
- 

#### <드리고 싶은 말씀>

---

기말 프로젝트는 여러분이 한 학기 동안 학습한 HTML, CSS, JavaScript의 핵심 개념을 종합적으로 적용해 보는 과정입니다.

복잡하고 완벽한 결과물을 만드는 것이 목표가 아니라, 자신의 아이디어를 웹으로 표현하고, 수업에서 배운 기술들을 자연스럽게 활용하는 경험을 쌓는 것이 더 중요합니다.

이 과제는 여러분의 5-6주차 과제(나만의 힐링송) 수준 정도로도 충분히 만족할 만한 결과를 낼 수 있습니다.

기준을 모두 완벽히 충족하지 못하더라도 감점 폭은 크지 않으며, 노력과 성실한 시도가 분명히 반영될 것입니다.

기말고사 준비로 바쁜 시기인 만큼, 무리하지 말고 본인이 구현할 수 있는 수준에서 완성도 있게 마무리해 주세요.

진행 중 막히거나 궁금한 부분이 있다면 언제든지 편하게 연락 주시기 바랍니다.

**여러분이 한 학기 동안 쌓아온 감각과 노력이 결과물 속에서 자연스럽게 드러나길 기대합니다!**

---

프로젝트 품질 기준(체크리스트)

영역	평가 항목
	세부 조건
1. 디자인과 레이아웃 (Layout & Design)	L1. [Flexbox] 1차원 정렬 완성 “Flexbox를 이용해 한 줄에 있는 메뉴나 버튼들을 justify-content로 정확히 정렬했나요?” 💡 확인: 요소가 한 줄로 정렬되고 간격이 일정하게 유지되면 OK
	L2. [CSS Grid] 2차원 배치 완성 “Grid로 콘텐츠를 행과 열로 구분해 정렬했나요?” 💡 확인: display: grid; grid-template-columns: 속성이 적용됨
	L3. [반응형] 모든 화면 대응 “데스크톱, 태블릿, 모바일에서 깨지지 않고 자연스럽게 줄 수가 바뀌나요?” 💡 확인: @media로 3단 → 2단 → 1단 전환
	L4. [Sticky] 헤더/메뉴 고정 “스크롤해도 상단 메뉴가 고정되어 있나요?” 💡 확인: position: sticky; top:0; 적용 여부
	L5. [Figma 추출] 픽셀 완벽 재현 “Figma에서 색상, 폰트, 간격을 확인하고 CSS에 그대로 적용했나요?” 💡 확인: 디자인 시안과 실제 페이지의 시각적 차이가 거의 없음
	L6. [BEM/외부 파일] “CSS와 JS 파일을 분리하고, 클래스 이름을 일관된 규칙(BEM 등)으로 작성했나요?”
2. 상호작용과 UX 품질 (Interaction & Quality)	U1. [Transition] 부드러운 전환 “Hover 시 색상·그림자 등이 즉시 바뀌지 않고 0.3초 이상 부드럽게 전환되나요?” 💡 확인: transition: all 0.3s ease; 적용
	U2. [Transform] 입체적인 움직임 “요소에 transform: scale()이나 translateY()를 사용해 입체감이 느껴지나요?”
	U3. [JS Toggle] 상태 전환 기능 “버튼 클릭 시 특정 클래스가 토글되어, 메뉴가 열리거나 다크 모드가 바뀌나요?” 💡 확인: element.classList.toggle() 사용
	U4. [setTimeout] 시간 제한 동작 “페이지 로드 후 또는 클릭 후 일정 시간 뒤 자동 동작(setTimeout)이 실행되나요?”
	U5. [setInterval] 자동 반복 동작 “3초마다 자동으로 바뀌는 요소(이미지·텍스트 등)를 구현했나요?”
	U6. [Clear] 자동 동작 제어 “멈춤 버튼을 클릭하면 자동 반복이 중단(clearInterval)되나요?”
	U7. [Form] 유효성 검사 “입력창에 빈 값이면 경고가 뜨나요? required, <label for> 등이 제대로 연결되어 있나요?”
3. 동적 데이터와 API (Dynamic Data & API)	D1. [fetch API] 외부 데이터 가져오기 “fetch()로 외부 JSON이나 Mock API 데이터를 가져왔나요?”
	D2. [Async/Await] 비동기 코드 정리 “async/await으로 fetch 코드를 더 읽기 쉽게 바꿨나요?”
	D3. [JSON] 데이터 변환 “응답 데이터를 response.json()으로 JS 객체로 변환했나요?”
	D4. [DOM 생성] 데이터 출력 “가져온 데이터를 반복문으로 <div>나 <li>로 화면에 표시했나요?”
	D5. [Error] 오류 처리 “try...catch로 오류가 나면 사용자에게 안내 메시지를 보여주나요?”
4. 접근성 및 포트폴리오 (Accessibility & Process)	A1. [시맨틱 HTML] 의미 있는 태그 사용 “<div> 대신 <header>, <main>, <footer> 등 의미에 맞는 태그를 사용했나요?”
	A2. [H1 규칙] 제목 체계 준수 “<h1>은 페이지에 한 번만 쓰고, <h2>, <h3> 순서가 올바른가요?”
	A3. [Alt & Focus] 접근성 필수 준수 “이미지에 alt가 있고, Tab 키로 포커스 이동 시 윤곽선이 보이나요?”
	A4. [배포] 인터넷 URL 공개 “GitHub Pages 등에 실제 배포해서 URL이 접속 가능한가요?”

## 웹페이지 설명서 예시 - “나만의 힐링송 페이지”

---

### ① 프로젝트 주제·목적

2026년 한 해 동안 들을 힐링송을 기분(감정)별로 정리하는 개인 플레이리스트 페이지. 내년의 내가 하루의 기분을 선택하면, 그에 맞는 노래 카드들이 나타나고 이미지·가사 하이라이트가 자동으로 바뀌어 감정에 맞는 곡을 탐색할 수 있다.

### ② 사용 기술 & 품질 기준 매핑(요약)

디자인·레이아웃: Sticky 헤더, Flex/Grid 기반 반응형 카드 목록 (L4, L3)

상호작용·품질: Hover 전환(Transition+Transform) (U1, U2)

동적 데이터·시간 제어: 5초 간격 자동 전환(setInterval) (U5)

접근성·포트폴리오: 시맨틱 태그 구조, 실서비스 URL 배포 (A1, A4)

### ③ 기능 상세 설명

#### 감정 선택 헤더(Sticky)

상단 헤더에 감정 선택 UI를 배치했다. 스크롤 중에도 헤더가 화면 상단에 고정되어 언제든지 감정을 바꿀 수 있다. 고정된 헤더 덕분에 탐색 흐름이 끊기지 않고, 페이지 전반의 사용성이 안정적으로 유지된다.

#### 카드형 노래 목록의 반응형 배치

노래들은 카드형으로 보여 준다. 화면이 넓을 때는 여러 열로, 좁아질수록 적은 열로 자연스럽게 재배치되어 모바일·태블릿·데스크톱 모두에서 읽기 쉬운 구조를 유지한다. 카드 내부 구성(이미지/텍스트/버튼)은 일관된 규칙으로 정렬되어 정보 스캔이 빠르다.

#### 카드 Hover 전환으로 품질감 강화

마우스를 올리면 카드가 부드럽게 강조된다(예: 살짝 확대, 그림자 강화 등). 이 전환 효과는 즉각적인 피드백을 제공해 ‘누를 수 있는 요소’임을 명확히 하고, 페이지 전반의 시각적 완성도와 사용감을 끌어올린다.

#### 이미지·가사 하이라이트 자동 전환(5초 간격)

각 카드의 앨범 이미지와 가사 하이라이트가 일정 시간(5초)마다 자동으로 바뀐다. 사용자는 조작 없이도 콘텐츠가 주기적으로 갱신되는 경험을 하며, ‘일시정지/재시작’ 같은 제어 요소를 통해 원하는 속도로 감상할 수 있다.

#### 시맨틱 구조 & 실제 배포

페이지는 <header>-<main>-<section>-<footer>로 의미 단위가 분명하게 나뉘어 있고, 이미지에는 대체 텍스트가 제공된다. 결과물은 다음 주소로 접근할 수 있다.

### ④ AI 도구 활용 사례

이번 프로젝트를 진행하면서 AI를 단순히 코드 자동 생성기가 아니라 아이디어 보조·문법 교정·코드 검증 도구로 활용했다.

#### 기획 단계

처음엔 페이지 주제를 어떻게 정할지 막막했는데, “음악과 감정을 연결한 웹 인터랙션 아이디어”를 ChatGPT에 물어보며 몇 가지 콘셉트를 얻었다. 그중 “감정별 음악 추천” 아이디어가 흥미로워 이를 기말 프로젝트 주제로 발전시켰다.

#### 레이아웃 설계

Flexbox와 Grid를 혼합해 카드형 레이아웃을 만들고 싶었지만, 처음에는 구조가 어지럽게 나와 정렬이 되지 않았다. ChatGPT에게 “Grid 안에 Flex를 중첩할 때 구조가 꼬이지 않게 하는 방법”을 물었고, grid-template-columns 설정 순서와 gap 조정 팁을 얻어 해결했다.

#### 웹페이지 설명서 작성

최종적으로 설명서를 작성할 때, 기술 용어와 문장 표현이 어색할까 봐 ChatGPT에게 “기술 포트폴리오 문체로 자연스럽게 다듬어줘”라고 요청하여 문장의 어조를 정리했다.

---