

17. 실전 프로젝트 기반 연습 항목

카드 UI, 프로필 박스 스타일링

다양한 웹 프로젝트에서 **카드 UI**와 **프로필 박스**는 정보를 시각적으로 정리하고, 사용자 경험을 향상시키는 핵심 구성요소이다. 아래는 실전 예제와 함께 핵심 속성, 스타일링 방법을 자세히 설명한다.

✅ 1. 카드 UI (Card UI)

📌 핵심 구성 요소

- **Container:** 박스 전체
- 이미지 영역
- 텍스트(제목, 설명)
- 버튼 또는 링크

🏗 HTML 구조 예시

```
1 <div class="card">
2   
3   <div class="card-content">
4     <h3 class="card-title">제목</h3>
5     <p class="card-desc">설명 텍스트</p>
6     <a href="#" class="card-btn">자세히 보기</a>
7   </div>
8 </div>
```

🎨 CSS 스타일 예시

```
1 .card {
2   width: 300px;
3   border-radius: 12px;
4   overflow: hidden;
5   box-shadow: 0 4px 16px rgba(0, 0, 0, 0.1);
6   background-color: #fff;
7   transition: transform 0.2s ease;
8 }
9
10 .card:hover {
11   transform: translateY(-4px);
12 }
13
14 .card-img {
15   width: 100%;
16   height: 180px;
17   object-fit: cover;
18 }
19
```

```

20 .card-content {
21     padding: 16px;
22 }
23
24 .card-title {
25     font-size: 1.2rem;
26     margin: 0 0 8px;
27     color: #333;
28 }
29
30 .card-desc {
31     font-size: 0.95rem;
32     color: #666;
33     margin-bottom: 12px;
34 }
35
36 .card-btn {
37     display: inline-block;
38     padding: 8px 16px;
39     background-color: #007acc;
40     color: white;
41     text-decoration: none;
42     border-radius: 8px;
43     font-size: 0.9rem;
44 }

```

✓ 2. 프로필 박스 (Profile Box)

📌 핵심 구성 요소

- 아바타 이미지
- 사용자 이름
- 상태 메시지 또는 설명
- 추가 정보 (예: 팔로워 수)

🏗 HTML 구조 예시

```

1 <div class="profile-box">
2   
3   <div class="profile-info">
4     <h4 class="profile-name">홍길동</h4>
5     <p class="profile-desc">프론트엔드 개발자</p>
6   </div>
7 </div>

```

CSS 스타일 예시

```
1  .profile-box {
2    display: flex;
3    align-items: center;
4    gap: 16px;
5    padding: 16px;
6    background-color: #f8f9fa;
7    border-radius: 10px;
8    box-shadow: 0 2px 10px rgba(0, 0, 0, 0.05);
9  }
10
11 .profile-img {
12   width: 64px;
13   height: 64px;
14   border-radius: 50%;
15   object-fit: cover;
16 }
17
18 .profile-info {
19   flex: 1;
20 }
21
22 .profile-name {
23   font-size: 1.1rem;
24   margin: 0;
25   color: #222;
26 }
27
28 .profile-desc {
29   font-size: 0.9rem;
30   margin-top: 4px;
31   color: #666;
32 }
```

UI 요소 확장 팁

기능	적용 방법
반응형 카드	<code>max-width: 100%;</code> , <code>media query</code> 로 레이아웃 조절
프로필에 상태 표시	<code>::after</code> 가상 요소 + 색상 원(circle)
카드에 배지 표시	<code>position: absolute</code> 로 상단 라벨 삽입
호버 인터랙션	<code>transform</code> , <code>transition</code> , <code>box-shadow</code> 로 강조 효과

✖ 보너스: 카드 UI + 프로필 박스 통합 예시

```
1 <div class="profile-card">
2   
3   <h4>김코딩</h4>
4   <p>백엔드 개발자 & 커피 애호가</p>
5   <a href="#" class="follow-btn">팔로우</a>
6 </div>
```

```
1 .profile-card {
2   text-align: center;
3   background: #fff;
4   border-radius: 12px;
5   padding: 20px;
6   width: 240px;
7   box-shadow: 0 0 10px rgba(0, 0, 0, 0.08);
8 }
9
10 .profile-card .profile-img {
11   width: 80px;
12   height: 80px;
13   border-radius: 50%;
14   object-fit: cover;
15   margin-bottom: 12px;
16 }
17
18 .profile-card h4 {
19   margin: 0;
20   color: #333;
21   font-size: 1.1rem;
22 }
23
24 .profile-card p {
25   font-size: 0.9rem;
26   color: #666;
27   margin-bottom: 12px;
28 }
29
30 .follow-btn {
31   background: #007acc;
32   color: #fff;
33   padding: 6px 14px;
34   text-decoration: none;
35   border-radius: 20px;
36   font-size: 0.85rem;
37 }
```

다음 주제

- 다크 모드 대응 카드
- Grid/Flex 기반의 카드 레이아웃 배치
- 카드 슬라이더 구성
- Tailwind로 동일 스타일 구현

반응형 내비게이션 바

반응형 내비게이션 바는 브라우저의 가로 너비에 따라 구조와 동작이 달라지는 메뉴 구성 방식이다.

보통 데스크탑에서는 가로 정렬 메뉴, 모바일에서는 햄버거 아이콘 메뉴로 바뀌는 구조를 많이 사용한다.

✓ 1. HTML 기본 구조

```
1 <nav class="navbar">
2   <div class="navbar-logo">MySite</div>
3   <ul class="navbar-menu">
4     <li><a href="#">홈</a></li>
5     <li><a href="#">서비스</a></li>
6     <li><a href="#">포트폴리오</a></li>
7     <li><a href="#">연락처</a></li>
8   </ul>
9   <div class="navbar-toggle" id="toggleBtn">&#9776;</div>
10 </nav>
```

✓ 2. CSS 스타일링

```
1 /* 기본 설정 */
2 * {
3   margin: 0;
4   padding: 0;
5   box-sizing: border-box;
6 }
7
8 .navbar {
9   display: flex;
10  align-items: center;
11  justify-content: space-between;
12  background-color: #004080;
13  padding: 12px 20px;
14  color: white;
15 }
16
17 .navbar-logo {
18   font-size: 1.5rem;
19   font-weight: bold;
20 }
21
```

```
22 .navbar-menu {
23   list-style: none;
24   display: flex;
25   gap: 20px;
26 }
27
28 .navbar-menu li a {
29   color: white;
30   text-decoration: none;
31   font-size: 1rem;
32 }
33
34 .navbar-toggle {
35   display: none;
36   font-size: 24px;
37   cursor: pointer;
38 }
```

✅ 3. 미디어 쿼리: 모바일 반응형 처리

```
1 @media (max-width: 768px) {
2   .navbar-menu {
3     display: none;
4     flex-direction: column;
5     width: 100%;
6     background-color: #004080;
7     position: absolute;
8     top: 60px;
9     left: 0;
10  }
11
12  .navbar-menu.active {
13    display: flex;
14  }
15
16  .navbar-toggle {
17    display: block;
18  }
19 }
```

✓ 4. JavaScript (햄버거 메뉴 열기)

```
1 <script>
2   const toggleBtn = document.getElementById('toggleBtn');
3   const menu = document.querySelector('.navbar-menu');
4
5   toggleBtn.addEventListener('click', () => {
6     menu.classList.toggle('active');
7   });
8 </script>
```

✓ 5. 구조 요약

해상도	구성
데스크탑	로고 + 가로 메뉴
모바일 (768px 이하)	로고 + 햄버거 아이콘 → 클릭 시 세로 메뉴 표시

✓ 확장 아이디어

기능	설명
서브 메뉴	메뉴 내부에 dropdown 추가 가능
애니메이션	메뉴 열고 닫기 시 <code>transition</code> 활용
아이콘 사용	FontAwesome 또는 SVG로 햄버거 / X 아이콘 전환
Sticky Nav	스크롤해도 상단 고정하려면 <code>position: sticky; top: 0</code>
Accessibility	<code>aria-expanded</code> , <code>aria-controls</code> 속성 추가 고려

그리드 기반 포트폴리오 레이아웃

CSS Grid는 시각적 정렬이 중요한 **포트폴리오, 갤러리, 카드 목록** UI 구성에 가장 적합한 레이아웃 시스템이다. 아래는 반응형을 지원하며 다양한 콘텐츠를 정돈되게 보여주는 **실전형 그리드 포트폴리오 레이아웃 예제**이다.

✓ 1. HTML 구조

```
1 <section class="portfolio">
2   <div class="portfolio-item">
3     
4     <h3>Project Title 1</h3>
5     <p>설명 또는 태그</p>
6   </div>
7   <div class="portfolio-item">
8     
9     <h3>Project Title 2</h3>
10    <p>설명 또는 태그</p>
11  </div>
12  <!-- 반복 -->
13 </section>
```

✓ 2. CSS: 기본 Grid 레이아웃

```
1 .portfolio {
2   display: grid;
3   grid-template-columns: repeat(auto-fit, minmax(280px, 1fr));
4   gap: 24px;
5   padding: 40px;
6   background-color: #f5f5f5;
7 }
8
9 .portfolio-item {
10  background: white;
11  border-radius: 12px;
12  overflow: hidden;
13  box-shadow: 0 4px 12px rgba(0, 0, 0, 0.08);
14  transition: transform 0.2s ease;
15 }
16
17 .portfolio-item:hover {
18   transform: translateY(-6px);
19 }
20
21 .portfolio-item img {
22   width: 100%;
23   height: 180px;
24   object-fit: cover;
25 }
26
27 .portfolio-item h3 {
28   margin: 16px;
29   font-size: 1.1rem;
30   color: #333;
31 }
32
```



```
33 .portfolio-item p {
34   margin: 0 16px 16px;
35   font-size: 0.9rem;
36   color: #666;
37 }
```

✓ 3. 주요 스타일 포인트

항목	설명
<code>auto-fit</code> + <code>minmax()</code>	반응형 컬럼 자동 정렬
<code>gap</code>	카드 사이 여백
<code>object-fit: cover</code>	이미지 비율 유지하며 잘림 처리
<code>hover</code>	카드 강조 인터랙션
<code>box-shadow</code> , <code>border-radius</code>	카드 형태 미적 보완

✓ 4. 반응형 정렬 예시

- `grid-template-columns: repeat(auto-fit, minmax(280px, 1fr));`
 - 가로폭이 넓을 때는 여러 열
 - 좁을 땐 자동으로 열 수 줄어듦
 - 모바일에서도 잘 작동

✓ 5. 확장 기능 제안

기능	적용 방법
필터링 버튼	JS와 <code>data-category</code> 조합
다크 모드 대응	<code>prefers-color-scheme</code> 활용
<code>grid-auto-flow: dense;</code>	격자 밀집 정렬
카드 내부 버튼 추가	"자세히 보기", "Live Demo" 등 추가

✓ 6. 예시 카드에 버튼 추가 (선택)

```
1 | <a href="#" class="btn-demo">Live Demo</a>
```

```
1 .btn-demo {
2   display: inline-block;
3   margin: 8px 16px 16px;
4   padding: 6px 12px;
5   font-size: 0.85rem;
6   background-color: #007acc;
7   color: white;
8   text-decoration: none;
9   border-radius: 6px;
10 }
```

다음 주제

- 필터링 가능한 포트폴리오 그리드
- Masonry 스타일의 불규칙 그리드
- Grid vs Flex 비교
- Tailwind 기반 포트폴리오 템플릿

다크모드 전환 기능 구현

다크모드는 사용자 경험을 향상시키고 눈의 피로를 줄이는 중요한 UI 요소다.

CSS와 JavaScript를 활용해 **다크/라이트 모드 전환 기능**을 직접 구현할 수 있으며, 시스템 설정 연동도 가능하다.

✓ 1. 기본 구조: HTML

```
1 <button id="toggle-theme"> 🌙 다크모드 </button>
2
3 <div class="content">
4   <h1>포트폴리오 제목</h1>
5   <p>설명 텍스트</p>
6 </div>
```

✓ 2. CSS: 라이트/다크 테마 정의

```
1 :root {
2   --bg-color: #ffffff;
3   --text-color: #111111;
4 }
5
6 body {
7   background-color: var(--bg-color);
8   color: var(--text-color);
9   transition: background-color 0.3s, color 0.3s;
10 }
11
12 body.dark-mode {
```

```

13  --bg-color: #111111;
14  --text-color: #ffffff;
15  }

```

✓ 3. JavaScript: 다크모드 전환 토글

```

1  <script>
2    const btn = document.getElementById('toggle-theme');
3    const body = document.body;
4
5    btn.addEventListener('click', () => {
6      body.classList.toggle('dark-mode');
7
8      const isDark = body.classList.contains('dark-mode');
9      btn.textContent = isDark ? '☀ 라이트모드' : '🌙 다크모드';
10
11      localStorage.setItem('theme', isDark ? 'dark' : 'light');
12    });
13
14    // 페이지 로드시 저장된 테마 적용
15    window.addEventListener('DOMContentLoaded', () => {
16      const savedTheme = localStorage.getItem('theme');
17      if (savedTheme === 'dark') {
18        body.classList.add('dark-mode');
19        btn.textContent = '☀ 라이트모드';
20      }
21    });
22  </script>

```

✓ 4. 시스템 설정 감지 (선택)

```

1  @media (prefers-color-scheme: dark) {
2    :root {
3      --bg-color: #111111;
4      --text-color: #ffffff;
5    }
6  }

```

- 사용자가 시스템에서 다크모드를 기본 설정했을 경우 자동 적용됨
- 위 CSS는 JS보다 우선 적용되며, 사용자가 수동 전환할 수 있게 하려면 `localStorage`로 우선권을 줘야 함

✓ 5. 고급 팁

기능	설명
<code>localStorage</code>	사용자 설정 유지

기능	설명
CSS 변수(<code>--color</code>)	테마 확장에 유리
<code>data-theme</code> 속성	HTML 또는 <code>:root[data-theme="dark"]</code> 방식도 있음
<code>transition</code>	자연스러운 전환 효과 적용
아이콘 전환	햄버거/태양/달 아이콘 함께 교체 가능 (FontAwesome, SVG 등 활용)

✓ 결과 요약

- CSS 변수로 테마 색상 구성
- 클래스 토글로 다크모드 적용
- `localStorage`로 상태 저장
- 미디어 쿼리로 시스템 감지 가능

다음 주제

- Tailwind 기반 다크모드 전환
- `data-theme` 방식 적용
- `@supports`를 활용한 조건 처리

CSS-only 드롭다운/토글/모달

JavaScript 없이 **CSS만으로 동작하는 UI 컴포넌트**는 HTML의 `<input>` 이나 `<details>` 태그와 CSS의 상태 선택자 (`:checked`, `:target`)를 활용해 만들 수 있다.

아래는 **CSS-only** 방식의 드롭다운, 토글 스위치, 모달 창 구현 예제이다.

✓ 1. 드롭다운 메뉴 (CSS-only)

✦ HTML

```
1 <div class="dropdown">
2   <input type="checkbox" id="menu-toggle" hidden>
3   <label for="menu-toggle" class="menu-button">메뉴 ▼</label>
4   <ul class="menu-list">
5     <li><a href="#">홈</a></li>
6     <li><a href="#">소개</a></li>
7     <li><a href="#">연락처</a></li>
8   </ul>
9 </div>
```

CSS

```
1 .menu-list {
2   display: none;
3   list-style: none;
4   padding: 0;
5   margin: 8px 0 0;
6   background: #f2f2f2;
7   border: 1px solid #ccc;
8 }
9
10 #menu-toggle:checked + .menu-button + .menu-list {
11   display: block;
12 }
13
14 .menu-button {
15   cursor: pointer;
16   padding: 8px 12px;
17   background-color: #007acc;
18   color: white;
19   display: inline-block;
20 }
```

✓ 2. 토글 스위치 (CSS-only)

HTML

```
1 <label class="switch">
2   <input type="checkbox" id="theme-toggle">
3   <span class="slider"></span>
4 </label>
```

CSS

```
1 .switch {
2   position: relative;
3   display: inline-block;
4   width: 50px;
5   height: 26px;
6 }
7
8 .switch input {
9   display: none;
10 }
11
12 .slider {
13   position: absolute;
14   cursor: pointer;
15   background-color: #ccc;
16   transition: 0.4s;
```

```

17   border-radius: 26px;
18   height: 100%;
19   width: 100%;
20 }
21
22 .slider::before {
23   content: "";
24   position: absolute;
25   height: 20px;
26   width: 20px;
27   left: 4px;
28   bottom: 3px;
29   background-color: white;
30   transition: 0.4s;
31   border-radius: 50%;
32 }
33
34 input:checked + .slider {
35   background-color: #007acc;
36 }
37
38 input:checked + .slider::before {
39   transform: translateX(24px);
40 }

```

✅ 3. 모달 창 (CSS-only, `:target` 이용)

📌 HTML

```

1  <a href="#modal" class="open-modal">모달 열기</a>
2
3  <div id="modal" class="modal-overlay">
4    <div class="modal-box">
5      <a href="#" class="close-modal">X</a>
6      <h2>모달 제목</h2>
7      <p>모달 내용입니다.</p>
8    </div>
9  </div>

```

🎨 CSS

```

1  .modal-overlay {
2    display: none;
3    position: fixed;
4    top: 0; left: 0;
5    width: 100%; height: 100%;
6    background: rgba(0,0,0,0.6);
7    justify-content: center;
8    align-items: center;
9    z-index: 999;

```

```
10 }
11
12 .modal-box {
13     background: white;
14     padding: 20px;
15     border-radius: 10px;
16     width: 300px;
17     max-width: 90%;
18     position: relative;
19 }
20
21 .close-modal {
22     position: absolute;
23     right: 10px;
24     top: 10px;
25     text-decoration: none;
26     font-size: 20px;
27 }
28
29 #modal:target {
30     display: flex;
31 }
```

✅ 핵심 기술 요약

방식	핵심 속성/태그	특징
드롭다운	<code>:checked</code> , <code><input></code>	라벨 클릭으로 열고 닫기 가능
토글 스위치	<code>:checked</code> , <code><input></code>	스위치 형태 구성 가능
모달 창	<code>:target</code> , <code></code>	해시 이동으로 모달 열림

✅ 확장 가능 포인트

- `details/summary` 태그 활용한 네이티브 드롭다운도 가능
- `animation`, `transform` 으로 부드러운 인터랙션 추가
- `:has()` 선택자 지원 시 인터랙션 다양성 증가 (크로스 브라우징 고려 필요)

다음 주제

- 순수 CSS 아코디언 메뉴
- CSS-only 탭 메뉴
- 다크 모드와 연동한 스위치

인터랙션 효과 중심의 애니메이션

사용자의 클릭, 호버, 포커스 등 **UI 상호작용(interaction)**에 반응하여 부드러운 경험을 제공하는 **애니메이션 효과**는 웹에서 매우 중요한 역할을 한다.

아래는 실무에서 자주 활용되는 **CSS-only 인터랙션 애니메이션 예제**를 정리한 것이다.

✓ 1. 버튼 호버 애니메이션

```
1 <button class="btn-zoom">클릭하세요</button>
```

```
1 .btn-zoom {
2   background-color: #007acc;
3   color: white;
4   padding: 12px 20px;
5   border: none;
6   border-radius: 8px;
7   font-size: 1rem;
8   transition: transform 0.2s ease, background-color 0.3s;
9 }
10
11 .btn-zoom:hover {
12   transform: scale(1.05);
13   background-color: #005f99;
14 }
```

버튼에 `scale`을 주면 즉각적인 피드백을 줌

✓ 2. 카드 hover 애니메이션 (상승 + 그림자)

```
1 <div class="card">
2   <h3>카드 제목</h3>
3   <p>카드 설명입니다.</p>
4 </div>
```

```
1 .card {
2   background: white;
3   padding: 20px;
4   border-radius: 10px;
5   box-shadow: 0 0 0 rgba(0, 0, 0, 0);
6   transition: transform 0.3s ease, box-shadow 0.3s ease;
7 }
8
9 .card:hover {
10   transform: translateY(-8px);
11   box-shadow: 0 8px 20px rgba(0, 0, 0, 0.1);
12 }
```


✓ 3. 링크 밑줄 애니메이션

```
1 | <a href="#" class="link-underline">자세히 보기</a>
```

```
1 | .link-underline {  
2 |     position: relative;  
3 |     color: #007acc;  
4 |     text-decoration: none;  
5 | }  
6 |  
7 | .link-underline::after {  
8 |     content: '';  
9 |     position: absolute;  
10 |    bottom: 0;  
11 |    left: 0;  
12 |    height: 2px;  
13 |    width: 0%;  
14 |    background-color: #007acc;  
15 |    transition: width 0.3s ease;  
16 | }  
17 |  
18 | .link-underline:hover::after {  
19 |     width: 100%;  
20 | }
```

`::after`를 이용해 밑줄이 길어지는 효과 제공

✓ 4. 인풋 포커스 애니메이션

```
1 | <input type="text" class="input-animate" placeholder="입력하세요">
```

```
1 | .input-animate {  
2 |     padding: 10px;  
3 |     border: 2px solid #ccc;  
4 |     border-radius: 6px;  
5 |     transition: border-color 0.3s ease, box-shadow 0.3s ease;  
6 | }  
7 |  
8 | .input-animate:focus {  
9 |     border-color: #007acc;  
10 |    box-shadow: 0 0 5px rgba(0, 122, 204, 0.5);  
11 |    outline: none;  
12 | }
```

✓ 5. 메뉴 열릴 때 Fade + Slide 애니메이션

```
1 <ul class="dropdown-menu">
2   <li>Item 1</li>
3   <li>Item 2</li>
4 </ul>
```

```
1 .dropdown-menu {
2   opacity: 0;
3   transform: translateY(-10px);
4   transition: all 0.3s ease;
5   pointer-events: none;
6 }
7
8 .dropdown:hover .dropdown-menu {
9   opacity: 1;
10  transform: translateY(0);
11  pointer-events: auto;
12 }
```

`opacity` 와 `translateY` 를 동시에 사용하면 부드럽게 나타남

✓ 6. 커스텀 @keyframes 애니메이션 예시

```
1 <div class="pulse-box"></div>
```

```
1 .pulse-box {
2   width: 100px;
3   height: 100px;
4   background: #007acc;
5   border-radius: 50%;
6   animation: pulse 2s infinite;
7 }
8
9 @keyframes pulse {
10  0% { transform: scale(1); opacity: 1; }
11  50% { transform: scale(1.1); opacity: 0.7; }
12  100% { transform: scale(1); opacity: 1; }
13 }
```

✓ 인터랙션 애니메이션 설계 원칙

항목	설명
<code>transition</code>	단순 상태 전환 (hover, focus)
<code>transform</code>	성능이 뛰어남 (<code>translate</code> , <code>scale</code> 등)

항목	설명
<code>box-shadow</code> , <code>opacity</code>	자연스러운 강조 표현
<code>@keyframes</code>	반복 애니메이션 또는 커스텀 동작
<code>will-change</code>	성능 최적화에 도움됨 (주의해서 사용)

다음 주제

- 애니메이션 지연 (`transition-delay`)
- 카드 flip, 슬라이더, 로딩 애니메이션
- 인터랙션 UX 접근성 고려