

# 웹프로그래밍 활용 Term Project

2501110223 한모세

## 주제선정이유

지난 HTML 과제때 크게 총 3가지의  
페이지를 제작하였다

1. 퀴즈 기능이 있는 메모장
2. 드래그 앤 드랍 기능이 있는 페이지
3. 별똥별이 내리고 호버 기능이 있는 페이지

이 중에서 2번, 3번을 구현 하기로 결정했다  
이유는 저 두 페이지가 가장 상징성이 있는 페이지이고  
가장 시각적 재미가 있는 페이지이기 때문

# 구현페이지 화면캡처 및 설명(기술, UI편의성)



## 2. useState를 이용한 상태(state) 관리

사용된 State:

- droppedItems → 현재 항아리에 떨어진 재료 목록
- currentDragged → 현재 드래그 중인 재료 ID 추적
- opacityState → 각 재료 이미지의 투명도 관리
- jarImage → 선택된 조합에 따라 항아리 이미지 업데이트  
→ State 변경마다 UI 렌더링

## 1. 함수형 컴포넌트 구조 활용

- Jsx파일 전체가 React의 함수형 컴포넌트
- useState()를 사용하여 UI 상태 및 투명도, 드래그 상태 등을 관리
- 재사용 가능한 컴포넌트로 분리하기 위해
- IngredientItem.jsx를 분리하여 props 전달 구조 사용.

## 3. 이벤트 핸들러로 드래그 앤 드롭 구현

- onDragStart
  - onDragEnd
  - onDragOver
  - onDrop
- 드래그 시작 → 드롭 → 항아리 이미지 업데이트

## 구현페이지 화면캡처 및 설명(기술, UI편의성)



### 1. 함수형 컴포넌트 기반 구조

- Jsx 파일 전체가 React의 함수형 컴포넌트로 구성
- Constellation.jsx : 전체 배경/별똥별 로직 담당
  - ConstellationGroups.jsx : 그룹(별자리 버튼) 및 텍스트 UI 담당
- UI와 로직을 분리하여 구조로 유지보수성이 높음

### 2. useEffect를 활용한 반복 애니메이션 생성

HTML 때는 별똥별 생성을 위해 script 기반 방식을 사용했지만  
React 때는 useEffect() 훅을 사용하여 구현

- setInterval()을 useEffect 내부에서 실행
  - 컴포넌트가 unmount될 때 clearInterval() 자동 실행
- ```
useEffect(() => {  
  const interval = setInterval(createShootingStar, 100);  
  return () => clearInterval(interval);  
}, []);
```