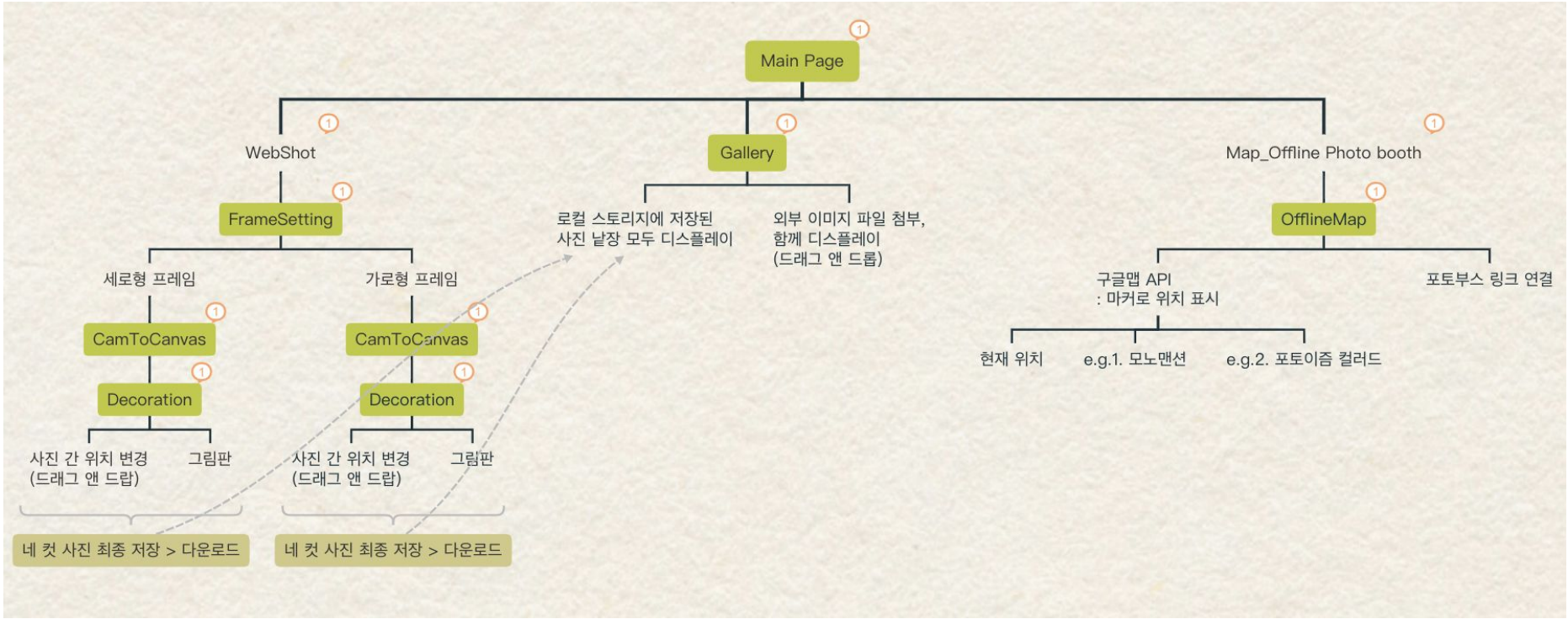
2023.09.18.

[HTML5/JavaScript] 웹 애플리케이션 제작 프로젝트

웹 포토 부스

KOSTA 265기 최혜린

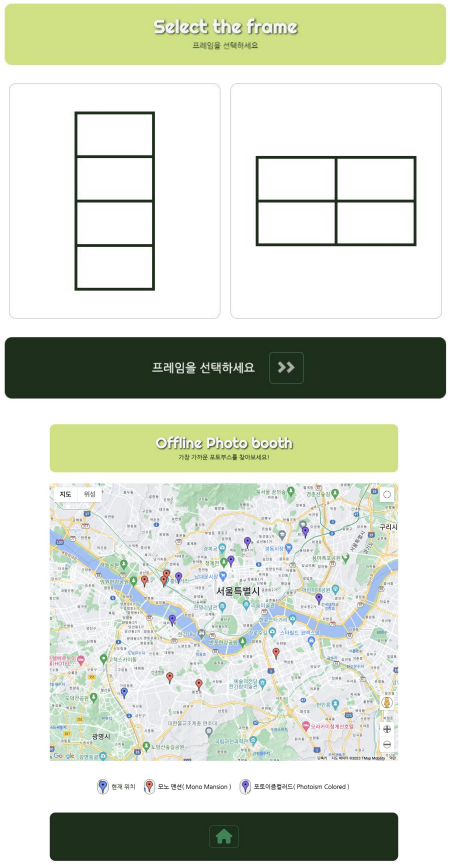
1. 아이디어 스케치



https://gitmind.com/app/docs/m2mrt1ur

2. 구현 화면


WebShot
시작하기
Offline Photo booth
위치 보기
Show All
이미지를 추가하세요
Copyright 2023 Hyerin Choi : Project_BrainStorming


Select the frame
프레임을 선택하세요
프레임을 선택하세요
Offline Photo booth
가장 가까운 포토부스를 찾아보세요!
지도
위성
서울특별시
구리시
광명시
현재 위치
모노 맨션(Mono Mansion)
포토이즘컬러드(Photoism Colored)

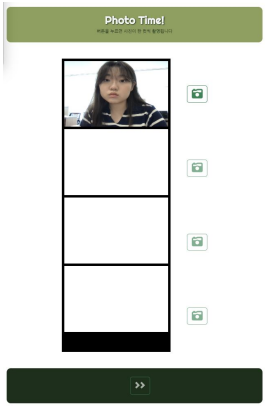


Photo Time!

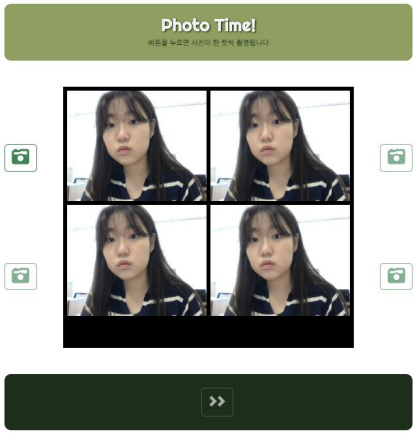
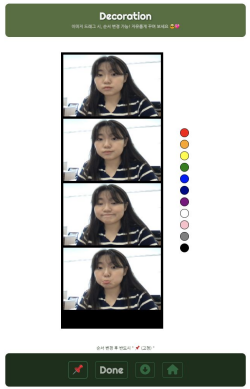
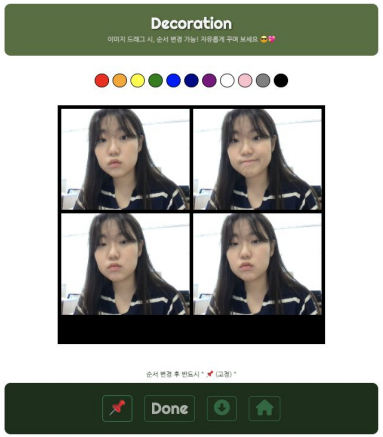


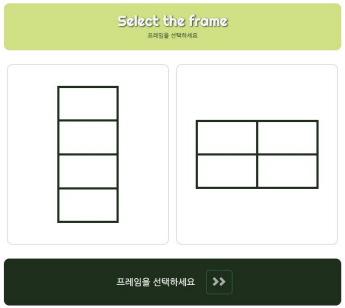
Photo Time!
버튼을 누르면 사진이 한 컷씩 촬영됩니다


Decoration
Done


Decoration
이미지 드래그 시, 순서 변경 가능! 자유롭게 꾸며 보세요
Done

3. FrameSetting

CSS 이용한 효과 연습



1. 마우스 호버(hover) 상태에 따라 요소 크기 변환 효과(CSS)

```css
#left, #right{
    transition: transform 0.3s; /* 변환(크기 변경)에 대한 전환 효과 추가 */
}

.hover:hover{
    transform: scale(1.05); /* 요소에 마우스 올라갈 때 크기 확대 */
    cursor: pointer;
}
```

2. 클릭 시, 요소 색 변화 효과(CSS, JS)

CSS 복사 캡션

```css
.framebox{
    display: inline-flex;
    width: 45%;
    height: 500px;
    flex-grow: 1;
    margin: 10px;
    padding: 10px;
    border: 1px solid #ccc;
    justify-content: center; /*아이템 중앙 정렬*/
    align-items: center;  /*아이템 중앙 정렬*/
}

.framebox.clicked{
    background-color: #cccccc; /* 클릭 시 어둡게 색상 변경 */
}
```

css 코드

- framebox를 클래스로 갖는 각 div 요소에는 CSS 클래스(.clicked)가 없으므로 클릭 이벤트 전에는 frambox에 설정한 배경색 로드

- framebox를 클래스로 갖는 각 div 요소에는 CSS 클래스(.clicked)가 없으므로 클릭 이벤트 전에는 frambox에 설정한 배경색 로드

```js
//선택에 따라 배경색, 메시지 업데이트하는 함수
function update(){
    if(isLeftselected){
        left.classList.add("clicked");
        right.classList.remove("clicked");
        displayMessage("세로형을 선택하셨습니다");
    }else{
        left.classList.remove("clicked");
        right.classList.add("clicked");
        displayMessage("가로형을 선택하셨습니다");
    }
}
```

js 코드

- 클릭 이벤트 발생 시, .clicked 클래스가 추가되면서 요소 배경색 변경
  - 왼쪽 옵션을 선택한 경우 왼쪽 div 요소의 배경색이 회색으로 변경
  - 오른쪽 옵션을 선택한 경우 오른쪽 div 요소의 배경색이 회색으로 변경

4. CamToCanvas

웹캠 스트림 캔버스 별 개별 제어

1. 웹캠으로 찍는 화면을 동영상 형태로 각 캔버스에 출력

   [문제 상황] 동영상 형태의 출력이 되지 X
   [해결] requestAnimationFrame 메소드 사용

2. 버튼 클릭 시 출력되던 영상을 이미지 스크린샷 출력으로 바꾸기
   => 해당 화면을 png URL 로 변환
   => URL을 동일 캔버스에 이미지 스크린샷으로 출력

   [문제 상황] 버튼을 눌러도 웹캠 스트림이 멈추지 X & 스크린샷이 캔버스에 그려지지 X
   [해결] 웹캠 스트림과 비디오 엘리먼트를 비활성화 시킨 뒤, 캔버스 초기화, 이미지 URL 그려지게 설정

3. 이때, 각 캔버스와 버튼의 개별 제어가 가능해야한다.

   [문제 상황] 첫 번째 버튼 눌렀을 때 첫 번째 캔버스에 그려진 이미지 URL -> 나머지 캔버스에도 동시에 출력
   [해결] 각 캔버스에 맞춰서 video 1, 2, 3, 4 분리

5. Decoration

1. [드래그앤드롭] 캔버스 별 위치 이동 - 캔버스 재배치 가능

1) DOM 요소 위치 변경: parent.insertBefore(draggedCanvas, this);

2) 드롭 대상 캔버스에 css 효과 주기 - 드롭 대상에서 벗어났을 시 효과 제거

2. [캔버스] 간단 그림판 - 이미지 위에 드로잉 가능

1) 캔버스 위치 '고정 후', 그림판 활성화

2) 색상 버튼 클릭 시 펜 색상 변경

3. [로컬 스토리지] 네 컷의 이미지를 하나의 이미지로 합친 뒤 저장 - 다운로드

1) canvas-container 영역의 내용(네 컷 이미지)을 하나의 이미지로 합치기:

- canvas-container 영역을 새로운 빈 캔버스에 그리기
- 그려진 새 캔버스의 내용을 이미지로 추출

2) 추출한 이미지를 다운로드 가능한 형식으로 제공하기:

- 이미지를 데이터 URL로 변환
- prompt 통해 데이터 URL을 링크로 제공

6. Gallery

[개선 필요한 부분]

1. 네 컷 이미지 합본 정교화
2. 이미지 사이즈 차이에 따른 갤러리 영역 공백 처리
3. 네이게이션 바 활용한 이미지 종류 별 분류
4. 로컬 스토리지보다 안정적인 저장소 연결

```
const imageContainer = document.getElementById('galleryArea');

// 화면에 이미지를 추가하는 함수
function addImageToContainer(imageUrl) {

    // 이미지를 생성하고 설정
    var image = new Image();
    image.src = imageUrl;

    // 이미지에 .ws-gallery-item 클래스를 추가
    image.classList.add('ws-gallery-item');

    // 이미지를 div에 추가
    imageContainer.appendChild(image);
}
```

```
// 로컬 스토리지에서 이미지를 검색하고 화면에 추가
for (var i = 1; i <= imageCount; i++) {
    var key = 'shot' + i;
    var imageUrl = localStorage.getItem(key);

    if (imageUrl) {
        // 이미지를 화면에 추가
        addImageToContainer(imageUrl);
    }
}

for (var k = 1; k <= mergedHCount; k++) {
    var mergedHKey = 'mergedH' + k; // 변수명을 k로 수정
    var mergedHImgUrl = localStorage.getItem(mergedHKey);

    if (mergedHImgUrl) {
        addImageToContainer(mergedHImgUrl);
    }
}

for (var j = 1; j <= mergedCount; j++) {
    var mergedKey = 'merged' + j;
    var mergedImgUrl = localStorage.getItem(mergedKey);

    if (mergedImgUrl) {
        addImageToContainer(mergedImgUrl);
    }
}

// 이미지 클릭 시 URL을 prompt로 띄우기
var galleryItems = document.querySelectorAll('.ws-gallery-item');

galleryItems.forEach(function(item) {
    item.addEventListener('click', function() {
        var imageUrl = prompt('이미지 URL을 입력하세요:', this.querySelector('img').src);
        if (imageUrl) {
            window.open(imageUrl, '_blank');
        }
    });
});
```

7. Offline Photo booth

구글맵 API 및 마커 연결 - 오프라인 포토 부스 위치 확인 가능

1. 배열에 직접 지점별 좌표 입력 후 for문 통해 마커로 출력
2. 브랜드마다 새 배열 구성

[개선 필요한 부분]

1. 좌표 직접 입력을 통해 위치 표시하는 방법 외 다른 방법 모색
2. 브랜드 검색 기능 추가 : 브랜드별 위치 마커 출력 기능
3. 구글맵 위치 및 비율 설정 정교화
4. 실시간 이용 현황 확인까지 가능하면 좋을 것으로 예상

2023.09.18.

감사합니다

KOSTA 265기 최혜린