Team Project

인물 맞추기 게임

고테아 홍세나 최명환 김지영

< who are you ? >

5초 안에 인물의 이름을 정확하게 맞추기

1문제당 5초 50명의 리스트중 20명만 보여줌

ALL clear 인 경우에만 명예의 전당에 등극

개발목적

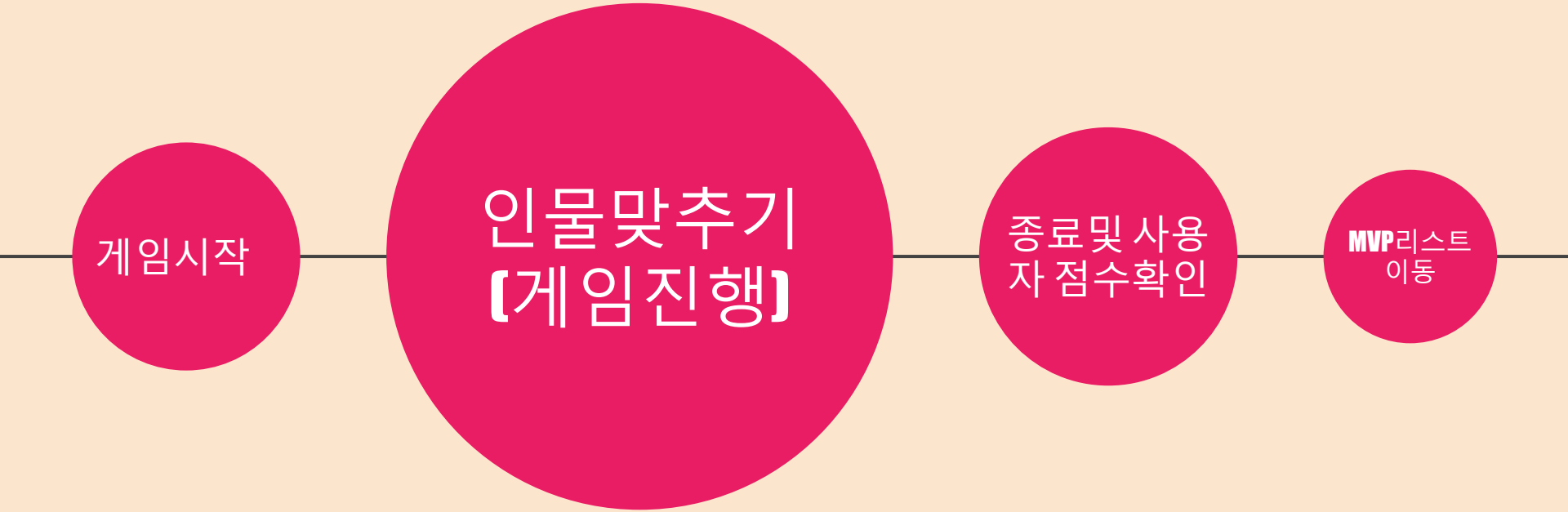
신서유기, 지구오락실 등 예능프로그램에서 진행하는

유명인 이름 맞추기 게임을 웹으로 간단하게

즐길수 있도록 구현하면서 올해의 유명인 순위를

간접적으로 알 수 있도록 개발

과정



페이지 소개

- START page (규칙설명 , 닉네임 입력 , 게임시작)
- GAME page (게임 메인페이지)
- END page (점수확인 및 restart 버튼 / mvp 버튼)
- MVP page (만점자 리스트 ♪)

점수산출기준

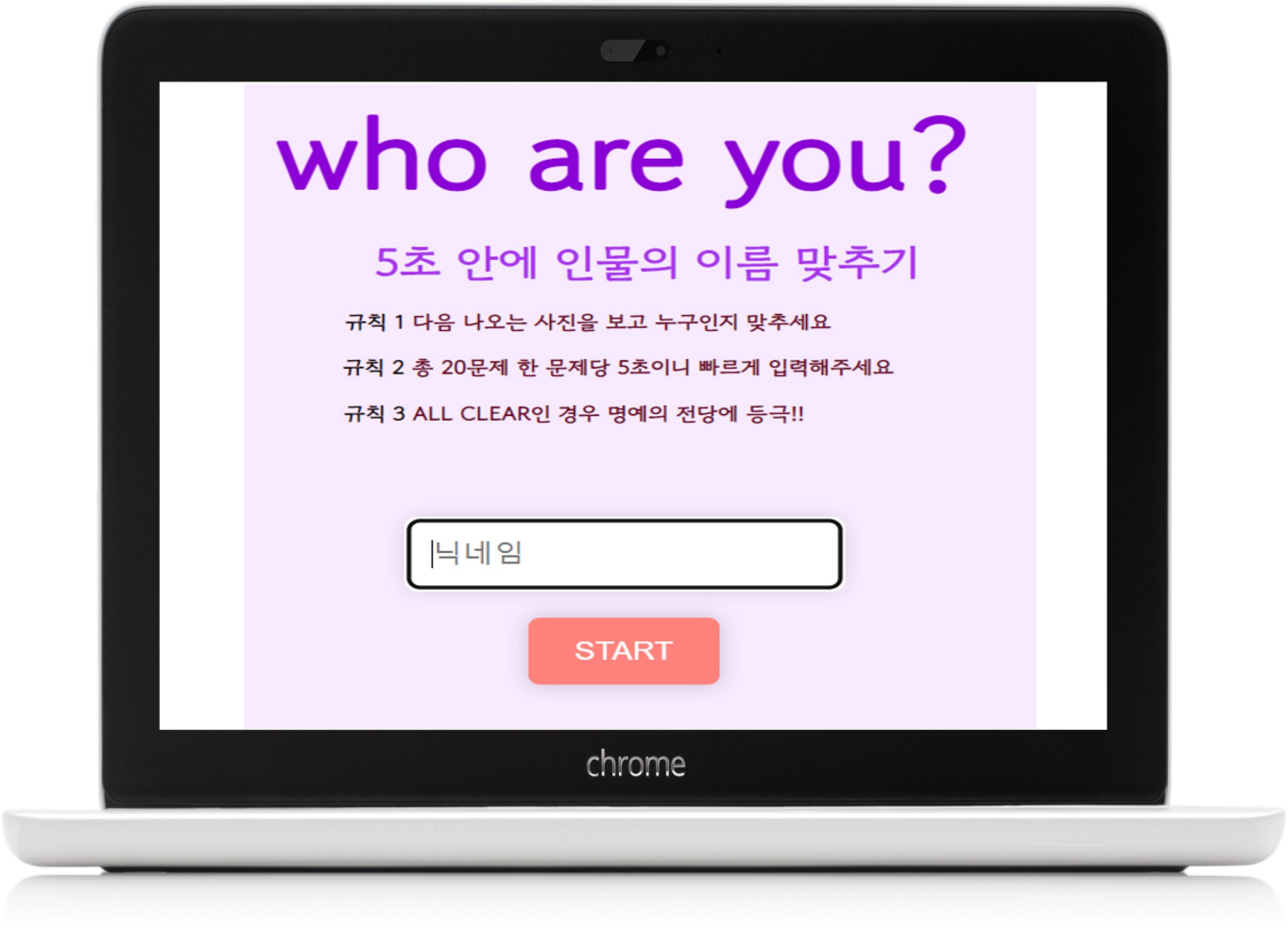
정답

- 인물명과 사용자가 입력한 값이 일치 1점추가
- 만점 시 mvp 리스트에 등극

오답

- DB에 입력된 이름만 정답으로 인정

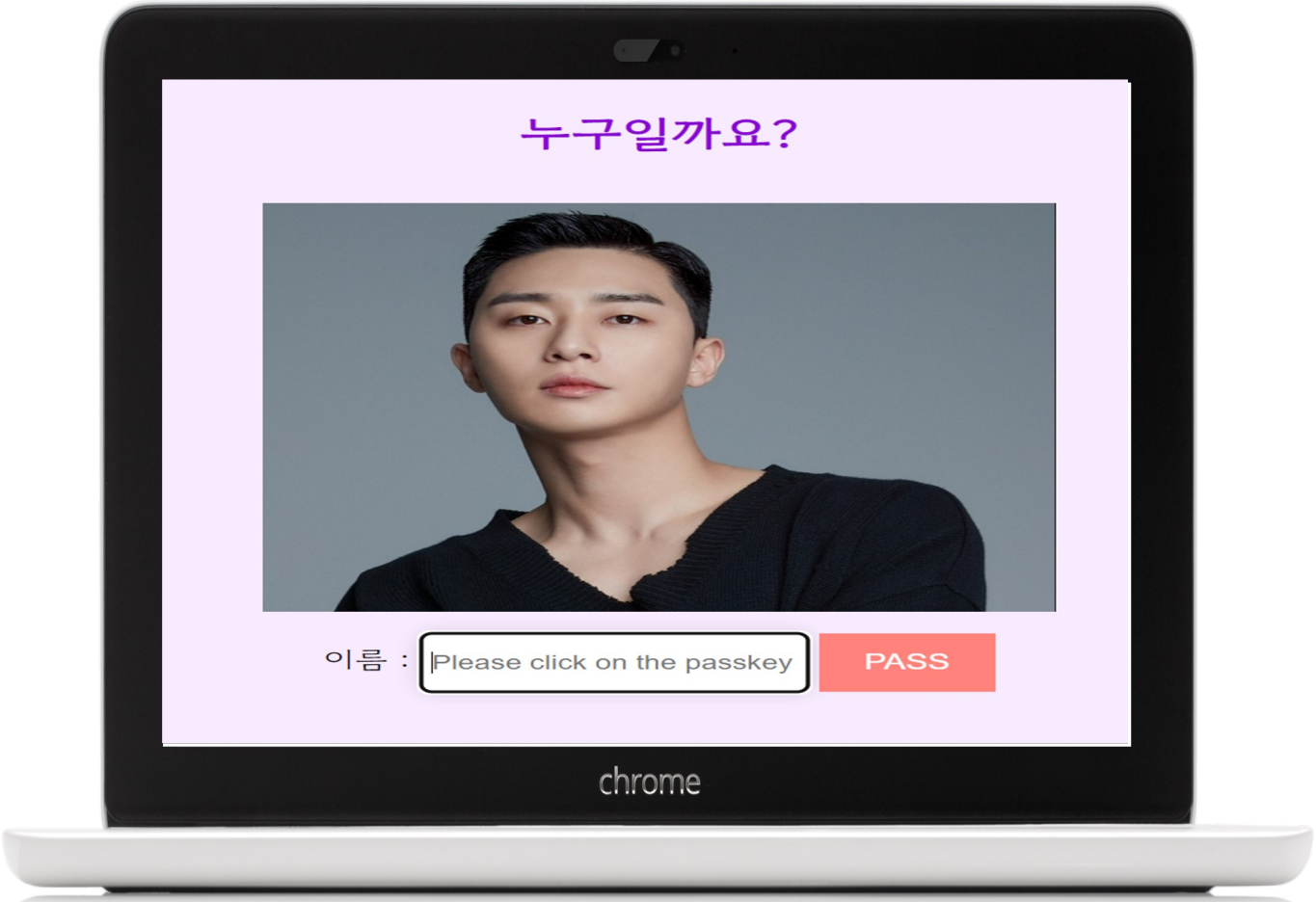
  ex) 방탄소년단 (0) , BTS (x)



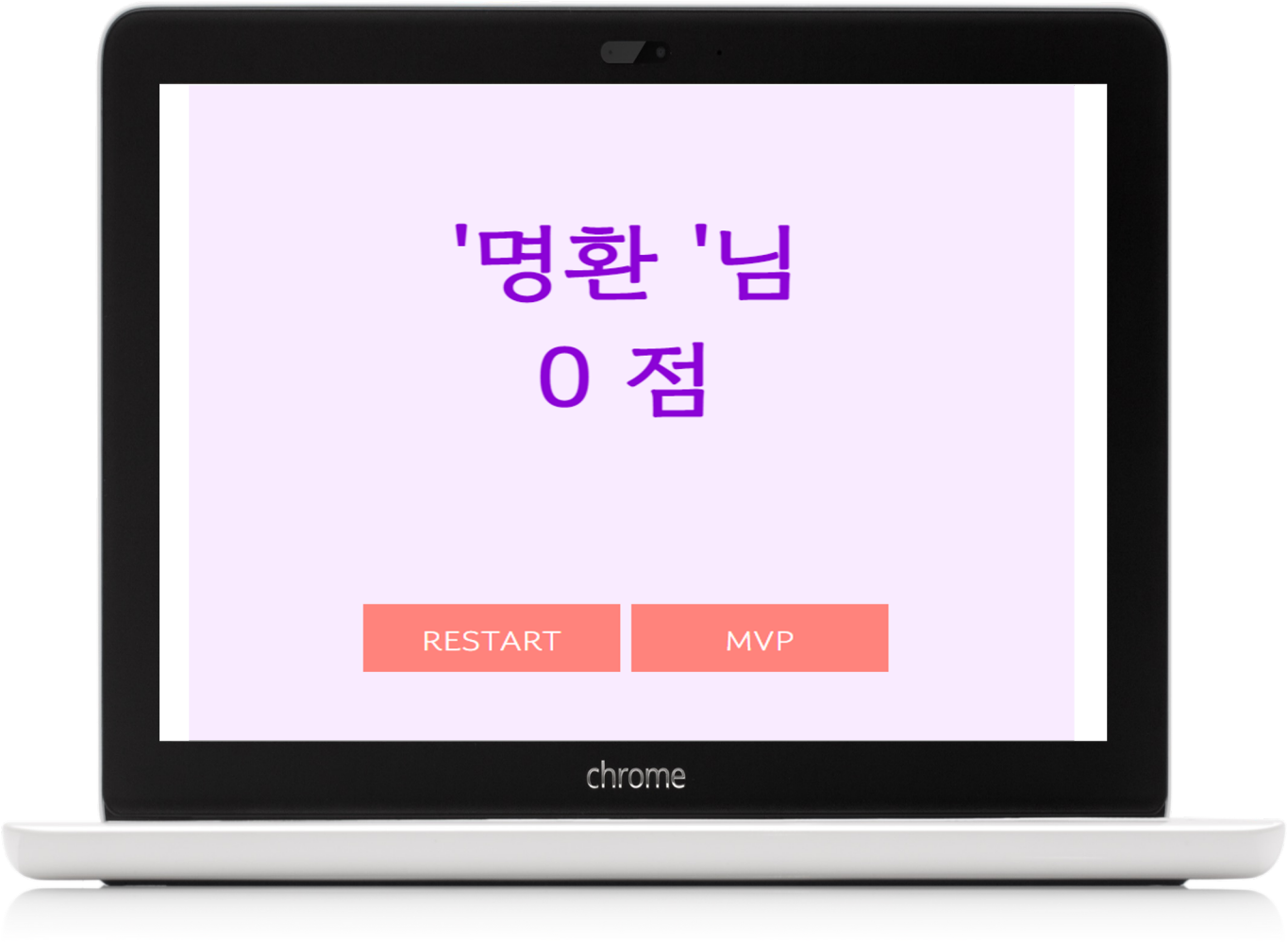
메인페이지에서

이름입력과 중복검사 진행

게임시작

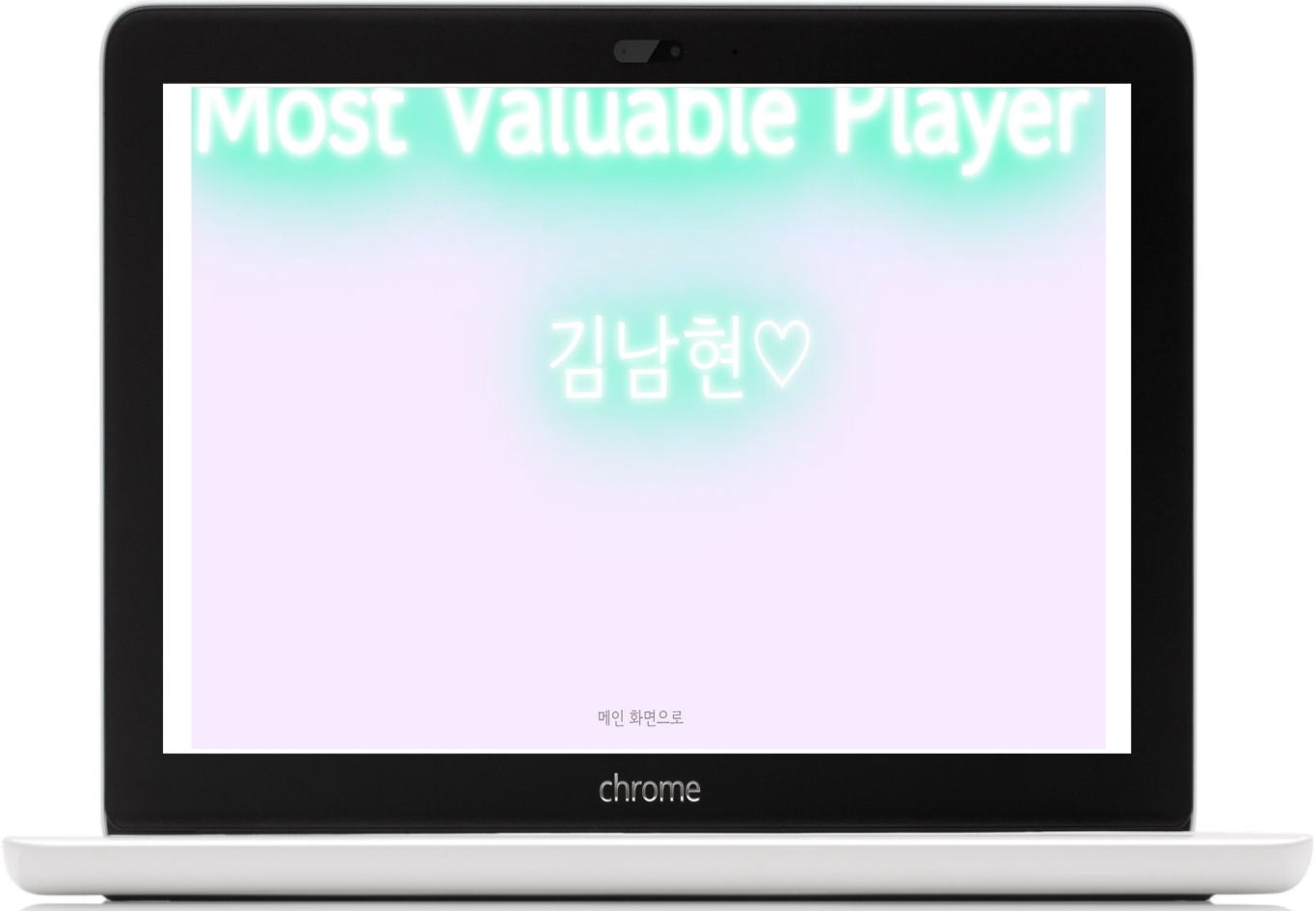


게임화면
DB 이미지 로드후
인덱스 순으로 송출
입력된 값과 DB 의 값을
비교하여 점수산출
input 유효성검사 및
사용자 편의성 추가



DB 에 저장된 결과 산출

메인이동,명예의전당이동



DB 에 저장된 결과 중

MVP 선별 후 산출

기획포인트

페이지별 어플 별도생성

css 파일 별도 생성

게임에 맞는 다이나믹 css

사용자 편의성을 고려한 코딩

필요내용만 DB업로드 할수 있도록 설계

향후 업데이트

MVP 리스트에 클리어타임 추가

인물의 분야별 카테고리 설정 추가

주요코드

1.사용자가 입력한 값을 가져와 null여부 및 중복 유효성 검사 진행후 DB저장

2.중복닉네임인경우 사용자에게 안내 편의성을 위해 포커스 추가

3.저장된 사용자의 닉네임 기본키를 세션에 저장

(게임종료시 점수산출 유효성을위함)

```python
from django.shortcuts import render, redirect
from django.contrib import messages
from .models import Main_User

def startmain(request):
    mvplist = Main_User.objects.filter(score=20)
    if request.method == 'POST':
        print(1)
        name = request.POST.get('nickname')
        if name != "":
            if Main_User.objects.filter(nickname=name).exists():
                messages.error(request,'이미 사용중인 닉네임입니다.')
                return redirect('/main/startmain/?msg=1')
            u = Main_User()
            u.nickname = name
            u.save()
            print(2, name)
            request.session['id'] = u.id
            request.session['nickname'] = u.nickname

            return redirect('/game/startgame/',{})
        return render(request, 'main/main.html', {'mvplist':mvplist})
    return render(request, 'main/main.html', {'mvplist':mvplist})

def static(request):
    return render(request, '/static.html')
```

주요코드

1.DB내부의 이미지 이름을 리스트로 가져와

사용자가 작성한 값이 리스트의 이름과 같은경우 점수부여 아닌경우 패스하며 사용자의 점수를 DB에 저장

2.이미지 전체를 가져와 중복되지 않게 랜덤으로 10개 추출

```python
from django.shortcuts import render,redirect
from django.http import HttpResponse
from .models import MyModel #User_Score
from main.models import Main_User

import random

def startgame(request):
    if request.method == 'POST':
        img_names = request.POST.getlist('img_name')
        print(img_names)
        score = 0
        for name in img_names:
            try:
                MyModel.objects.get(name=name)
                score += 1
            except:
                pass
        id = request.session.get('id')
        user = Main_User.objects.get(pk=id)
        user.score = score
        user.save()

        return redirect('/end/endgame/')


    # DB에서 이미지 목록 가져오기
    items =MyModel.objects.all()
    img_list = [item.img.url for item in items]
    selected_imgs = random.sample(set(img_list), 10)#랜덤문으로 40개의 수량만 확보 후 리스트로 재정렬
    context = {'selected_imgs':selected_imgs}

    return render(request, 'game/game.html', context)
```

주요코드

1.for 랜덤문을 이용해 이미지 리스트추출

2.show() 함수는 인덱스순으로 순차적인 이미지 디스플레이 모드를 변경해주고

3.js를 이용하여 순서가 지나간 인덱스에 클래스를 부여하여 디스플레이를 숨기고 현재 이미지에 포커스를 맞춰줍니다.

4.작성후 엔터를 치면 next함수가 발동할수 있도록 EventLisner사용

5. next함수발동시 잔여시간과 무관하게 다음이미지를 송출하기 위해 함수에 부여한 setInterval을 초기화한후 다음 이미지에 다시 시간을 부여합니다.

7. send() 함수는 "결과보기" 버튼을 클릭했을 때 호출되며, 동시에 사용자가 입력한 값을 데이터베이스와 대조후 점수부여

```
const selected_imgs = document.querySelectorAll('.selected_img');
let idx = 0
show()
  function show() {
    if (idx >= selected_imgs.length) {
      clearInterval(time);
      const nxetPage = document.querySelector('.next-page');
      nxetPage.style.display = 'block';
      const answerDisplay = document.querySelector('.active');
      answerDisplay.style.display = 'none';
      return;
    }
    if(idx > 0) {
      selected_imgs[idx-1].style.display = 'none';
      selected_imgs[idx-1].classList.remove('active')
    }
    selected_imgs[idx].style.display = 'block';
    selected_imgs[idx].classList.add('active')
    selected_imgs[idx].querySelector('[name=img_name]').focus()
    idx += 1;
  }
  let time = setInterval(show, 5000);
  function next() {
    clearInterval(time);
    show();
    time = setInterval(show, 5000);
  }
  document.addEventListener('keyup', function(event) {
    if (event.code === 'Enter') {
      let value =
        document.querySelector('.active [name=img_name]').value;
        if (value=="")return
        document.querySelector('.active [name=img_name]').value = value
        event.preventDefault();
        next();
    }});
```