|  |
| --- |
| My Django project  DB Assignment |

학과 : 컴퓨터공학과

학번 : 2015311152

이름 : 서한결

목 차

[1. Overview 5](#_Toc43060678)

[1.1. Topic 5](#_Toc43060679)

[1.2. Objective 5](#_Toc43060680)

[1.3. Functionality 5](#_Toc43060681)

[1.4. Explanation 5](#_Toc43060682)

[2. ER diagram 6](#_Toc43060683)

[3. Descriptions of ER diagram 6](#_Toc43060684)

[3.1. User entity 6](#_Toc43060685)

[3.2. Blog entity 7](#_Toc43060686)

[3.3. Write relation 7](#_Toc43060687)

[3.4. Imageuploadmodel entity 7](#_Toc43060688)

[3.5. Post relation 7](#_Toc43060689)

[3.6. Upload entity 8](#_Toc43060690)

[3.7. Charac entity 8](#_Toc43060691)

[3.8. Compare relation 8](#_Toc43060692)

[3.9. Select relation 8](#_Toc43060693)

[3.10. Choice entity 8](#_Toc43060694)

[3.11. Question entity 8](#_Toc43060695)

[3.12. Include relation 9](#_Toc43060696)

[4. Relational Model 9](#_Toc43060697)

[5. Descriptions of Relational Model 9](#_Toc43060698)

[5.1. Blog table 9](#_Toc43060699)

[5.2. User table 9](#_Toc43060700)

[5.3. Imageuploadmodel table 10](#_Toc43060701)

[5.4. Choice table 10](#_Toc43060702)

[5.5. Question table 10](#_Toc43060703)

[5.6. Charac table 10](#_Toc43060704)

[6. Implementation results 10](#_Toc43060705)

[6.1. Function 1: Users can sign up, log in and out the website 10](#_Toc43060706)

[6.2. Function 2: Users can see the posts that is written before including other users’ 12](#_Toc43060707)

[6.3. Function 3: Users can see the details about interesting posts 13](#_Toc43060708)

[6.4. Function 4: Users can test their psychopath scores 13](#_Toc43060709)

[6.5. Function 5: Users can upload their photos to know which are the most similar cartoon characters to users’ faces 14](#_Toc43060710)

[6.6. Function 6: After two tests, users can write the posts about tests results 14](#_Toc43060711)

그림 목차

[[그림 1] ER Diagram 5](#_Toc43060665)

[[그림 2] Relational Model 8](#_Toc43060666)

[[그림 3] 회원가입 화면 9](#_Toc43060667)

[[그림 4] 로그인 화면 10](#_Toc43060668)

[[그림 5] 로그인 후 홈 화면 10](#_Toc43060669)

[[그림 6] 로그아웃 후 첫 화면 11](#_Toc43060670)

[[그림 7] 작성된 게시물 11](#_Toc43060671)

[[그림 8] more 버튼을 눌러 해당 게시물 detail 확인 12](#_Toc43060672)

[[그림 9] 첫 번째 검사인 싸이코패스 검사 실시 12](#_Toc43060673)

[[그림 10] 두 번째 검사인 나와 닮은 캐릭터 찾기 13](#_Toc43060674)

[[그림 11] 검사 결과를 게시물로 작성 13](#_Toc43060675)

[[그림 12] 작성된 게시물 14](#_Toc43060676)

[[그림 13] 작성된 게시물 홈 화면에서 확인 14](#_Toc43060677)

1. Overview
   1. Topic

A website that shows cartoon characters which look similar to users’ faces and test users’ psychopath scores.

* 1. Objective

I want to make a website for people who want to know which cartoon characters look similar to their face and know their psychopath scores.

* 1. Functionality

Function 1: Users can sign up, log in and out the website

Function 2: Users can see the posts that is written before including other users’

Function 3: Users can see the details about interesting posts

Function 4: Users can test their psychopath scores

Function 5: Users can upload their photos to know which are the most similar cartoon characters to users’ faces

Function 6: After two tests, users can write the posts about tests results

* 1. Explanation

» 제가 진행한 프로젝트는 이용자들이 재미로 그들의 싸이코패스 점수를 테스트하고, 사진을 올려 그 사진과 제일 닮은 캐릭터를 볼 수 있는 웹 사이트에 대한 것입니다.

» 이용자들은 사용할 사용자명과 비밀번호를 지정하여 회원가입이 가능합니다. 그 계정으로 로그인, 로그아웃도 가능합니다.

» 이용자들이 로그인을 하게 되면 다른 이용자들이 작성한 것을 포함하여 지금까지 작성되었던 게시물들을 간략하게 볼 수 있습니다.

» 이용자들은 간략하게 보이는 게시물 중 관심이 있는 것들을 자세히 보는 것이 가능합니다.

» 이용자들은 두 가지의 간단하지만 흥미로운 테스트를 진행함으로써 그 결과를 게시물로 작성할 수 있습니다.

» 첫 번째 테스트는 몇 가지 간단한 질문을 통해 이용자의 싸이코패스 점수를 확인하는 것입니다.

» 두 번째 테스트는 이용자가 자신의 얼굴이 담긴 사진을 업로드 하게 되면 그 얼굴과 가장 닮은 캐릭터를 확인하는 것입니다.

» 두 번째 테스트까지 마무리되면 이용자들은 게시물을 작성할 수 있게 되고 게시물 작성을 통해 결과를 확인할 수 있습니다.

1. ER diagram

텍스트, 화이트보드이(가) 표시된 사진

자동 생성된 설명

[그림 ] ER Diagram

1. Descriptions of ER diagram
   1. User entity

id: User entity의 primary key이며int형입니다. user가 생성될 때 자동으로 이전 숫자보다 1씩 커지며 부여됩니다.

password: varchar형이며 user가 회원가입 시에 입력했던 것을 노출되지 않게 POST형으로 입력 받아 저장합니다.

last\_login: datetime형이며 해당 user가 마지막으로 로그인했던 시간을 기록하여 저장합니다.

is\_superuser: tinyint형이며 해당 user가 superuser인지 아닌지 판단하는 역할을 합니다.

username: varchar 형이며 user가 회원가입 시에 입력했던 사용자명을 저장합니다.

first\_name: 구현되어 있는 모델에서 사용하지 않은 attribute입니다. 모든 tuple에 null로 저장되어 있습니다.

last\_name: first\_name과 마찬가지로 사용하지 않은 attribute이고 모든 tuple에 null로 저장되어 있습니다.

email: varchar형으로 superuser를 제외한 다른 user들은 모두 null 값이 저장되어 있습니다.

is\_staff: tinyint형으로 해당 user가 staff인지 아닌지 판단하는 역할을 합니다.

is\_active: tinyint형으로 해당 user가 활동 중인지 아닌지 판단하는 역할을 합니다.

date\_joined: datetime형으로 해당 user가 회원가입한 시간을 저장합니다.

* 1. Blog entity

id: Blog entity의 primary key이며 int형이고 게시물이 작성될 때 자동으로 이전보다 1 증가하여 부여됩니다.

title: varchar형으로 게시물 작성시 입력했던 제목을 저장합니다.

pub\_date: date형으로 게시물을 작성한 날짜를 저장합니다.

body: longtext형으로 게시물 작성시 입력했던 본문을 저장합니다.

score: int형으로 default값은 0입니다. 게시물 작성시 첫 번째 검사의 점수로 update합니다.

* 1. Write relation

writer: varchar형으로 default는 superuser로 되어있고 게시물 작성 후 게시물을 작성한 username으로 update 됩니다.

* 1. Imageuploadmodel entity

id: Imageuploadmodel entity의 primary key이며 int 형이고 이미지가 업로드 될 때 자동으로 이전보다 1 증가하여 부여됩니다.

document: varchar형으로 업로드 된 이미지의 url을 저장합니다.

uploaded\_at: datetime형으로 업로드 된 시간을 저장합니다.

* 1. Post relation

게시물 작성시 업로드 되었던 이미지가 사용되는 것에 대한 relation입니다.

* 1. Upload entity

이미지가 업로드 되는 것에 대한 relation입니다.

* 1. Charac entity

id: Charac entity의 primary key로 int형입니다. 저장된 이미지들을 구분 짓는 역할을 합니다.

name: varchar형으로 업로드 된 캐릭터와 비교할 캐릭터들의 이름이 저장되어 있습니다.

imgurl: varchar형으로 각 캐릭터 이미지의 url이 저장되어 있습니다.

* 1. Compare relation

best\_character: varchar형으로 업로드 된 캐릭터와 저장된 캐릭터들을 비교했을 때 가장 비슷한 캐릭터의 이름을 저장합니다.

sim: int형으로 best\_character와 업로드 된 사진의 유사도를 저장합니다.

* 1. Select relation

첫 번째 검사를 진행할 때 choice를 선택하는 것에 대한 relation입니다.

* 1. Choice entity

id: Choice entity의 primary key로 int형이고 저장된 choice들을 구분 짓는 역할을 합니다.

choice\_text: varchar형으로 각 question에 대한 choice가 저장되어 있습니다.

votes: int형으로 각 choice들이 선택된 수를 저장하고 검사가 끝난 이후에 다시 0으로 update됩니다.

(question\_id: include relation에 total participation하고 있어 ER diagram에는 표시되어 있지 않은 실제 database 내 Choice table의 foreign key역할을 하는 column입니다. int형으로 Question table의 primary key를 참조하고 있습니다.)

* 1. Question entity

id: Question entity의 primary key로 int형이고 저장된 question들을 구분 짓는 역할을 합니다.

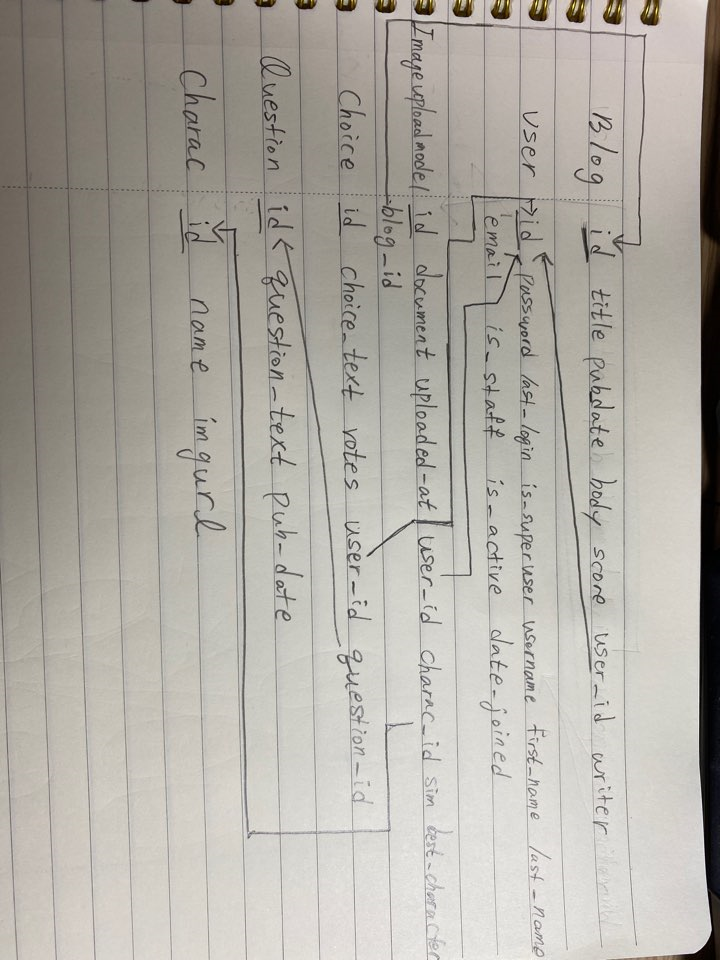
question\_text: varchar형으로 검사에 사용되는 질문들이 저장되어 있습니다.

pub\_date: datetime형으로 question이 저장된 시간을 저장합니다.

* 1. Include relation

각 question들은 choice를 두 개씩 가지고 있고 choice는 question을 참조하고 있습니다. Choice entity는 include relation에 total participation을 가집니다.

1. Relational Model



[그림 ] Relational Model

1. Descriptions of Relational Model
   1. Blog table

ER diagram에서 Blog entity와 User entity가 Write relation을 가지고 있었다. Blog table에 user\_id는 User table의 primary key인 id를 참조하고 writer는 Write relation의 attribute이다.

* 1. User table

ER diagram에서 표현된 attribute들을 그대로 표현하면 했다.

* 1. Imageuploadmodel table

user\_id는 User entity와 Post relation을 가진 결과로 생긴 column이고 User table의 primary key인 id를 참조한다. charac\_id는 Charac entity와 Compare relation을 가지고 있는 결과로 생긴 column이고 Charac table의 primary key인 id를 참조한다. sim과 best\_character는 Compare relation의 attribute를 추가한 것이다. blog\_id는 Blog entity와 Post relation을 가진 결과로 생긴 column이고 Blog table의 primary key인 id를 참조한다.

* 1. Choice table

user\_id는 User entity와 Select relation을 가진 결과로 생긴 column이고 User table의 primary key인 id를 참조한다. question\_id는 Quesiton entity와 Include relation을 가진 결과로 생긴 column이고 Question table의 primary key인 id를 참조한다.

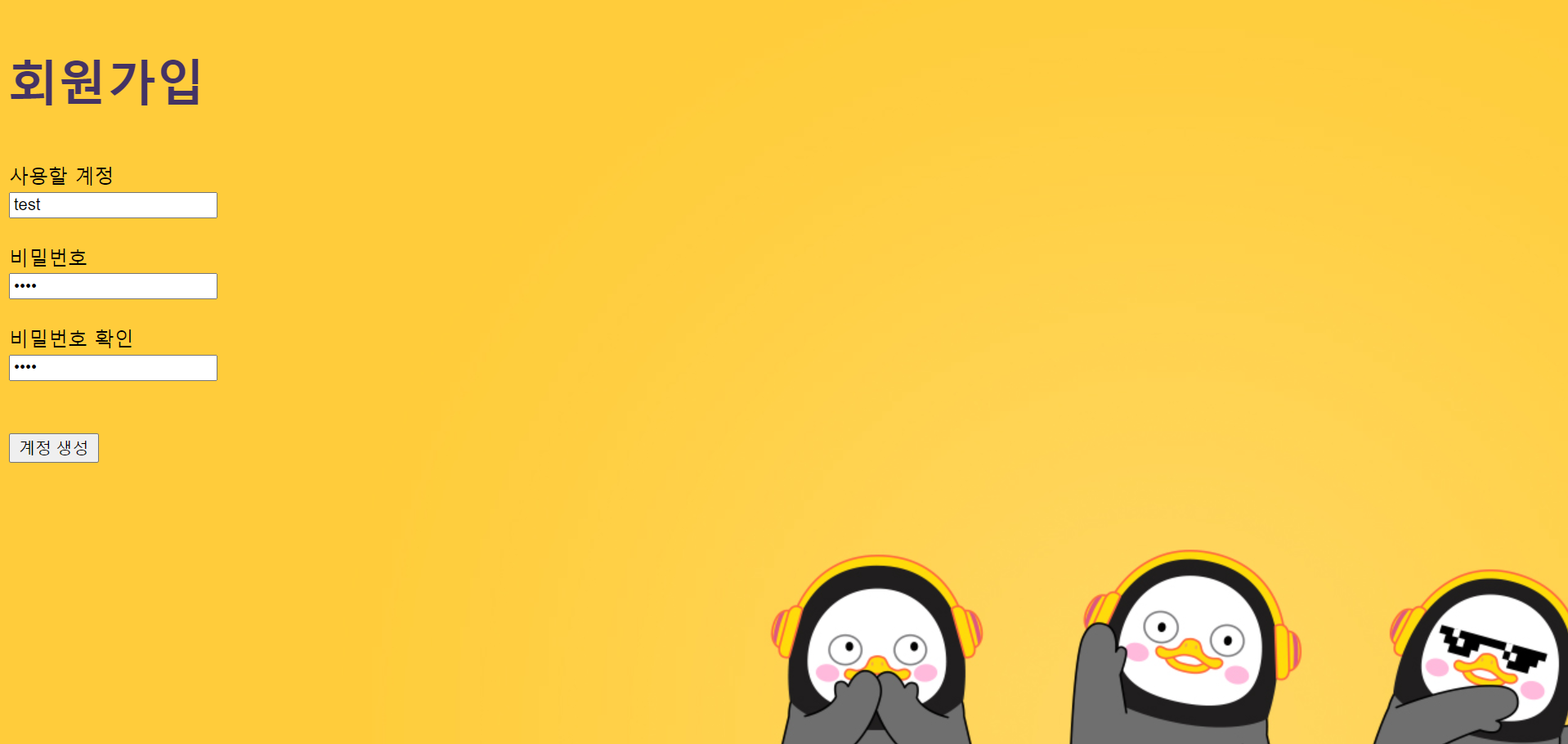
* 1. Question table

ER diagram에서 표현된 attribute들 그대로 표현했다.

* 1. Charac table

ER diagram에서 표현된 attribute들 그대로 표현했다.

1. Implementation results
   1. Function 1: Users can sign up, log in and out the website



[그림 3] 회원가입 화면



[그림 4] 로그인 화면



[그림 5] 로그인 후 홈 화면



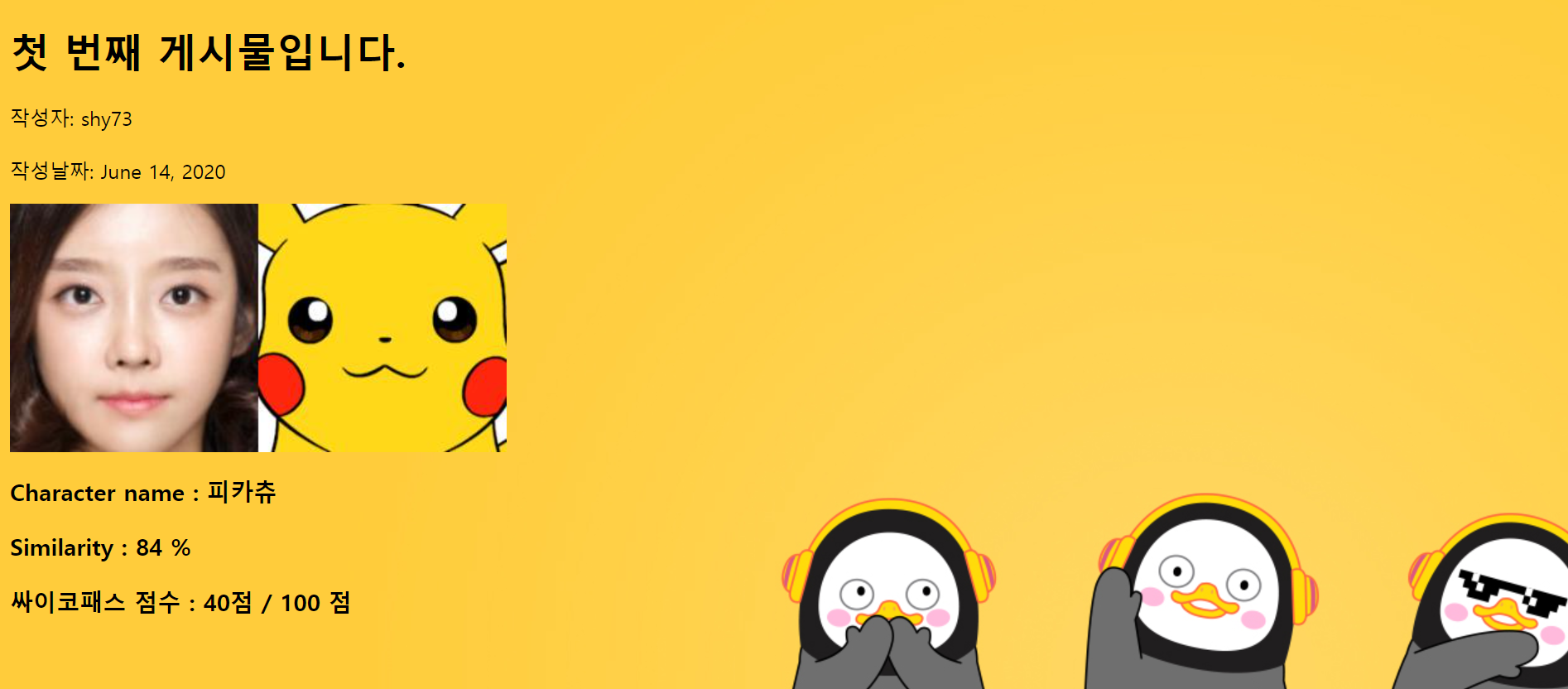
[그림 6] 로그아웃 후 첫 화면

* 1. Function 2: Users can see the posts that is written before including other users’



[그림 7] 작성된 게시물

* 1. Function 3: Users can see the details about interesting posts



[그림 8] more 버튼을 눌러 해당 게시물 detail 확인

* 1. Function 4: Users can test their psychopath scores



[그림 9] 첫 번째 검사인 싸이코패스 검사 실시

* 1. Function 5: Users can upload their photos to know which are the most similar cartoon characters to users’ faces

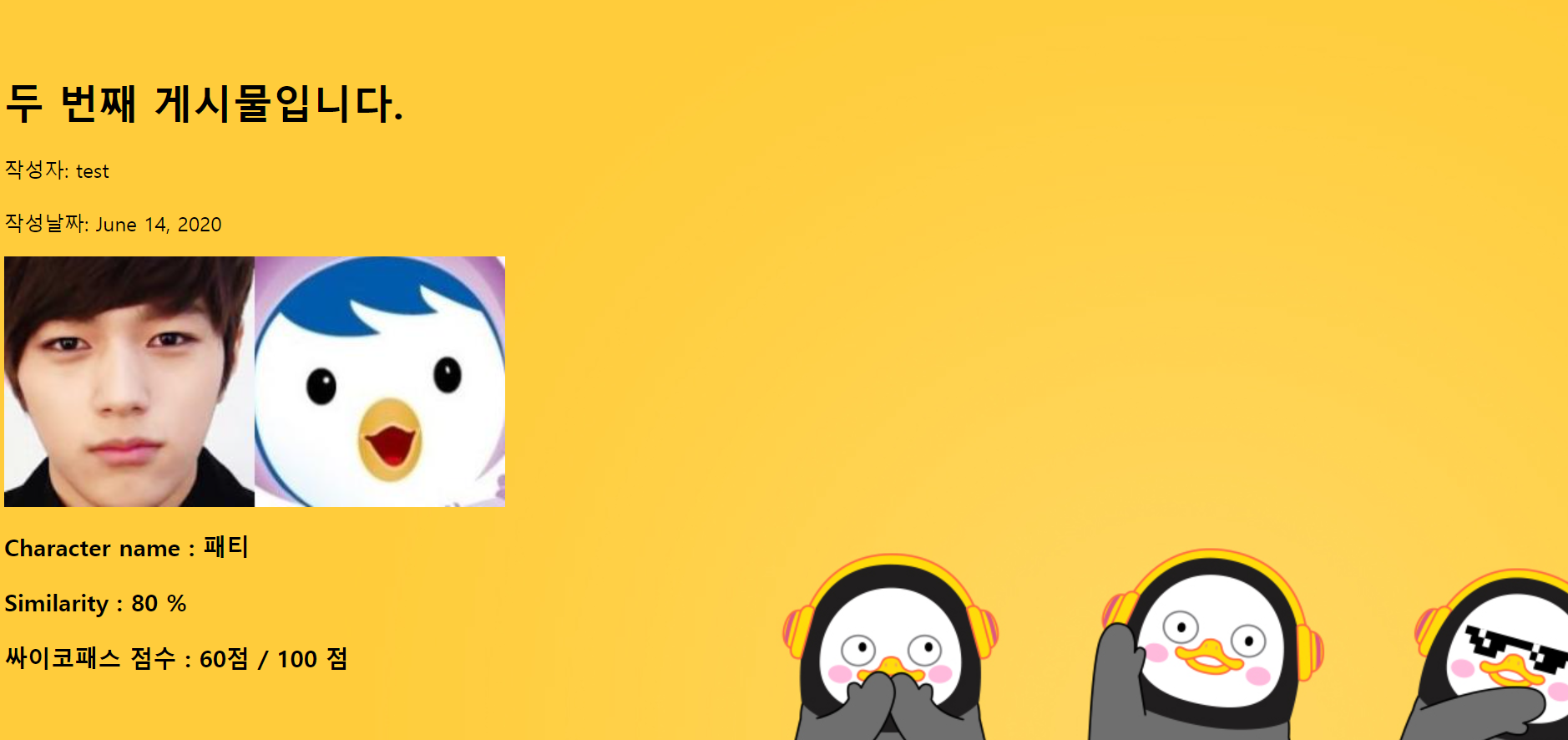


[그림 10] 두 번째 검사인 나와 닮은 캐릭터 찾기

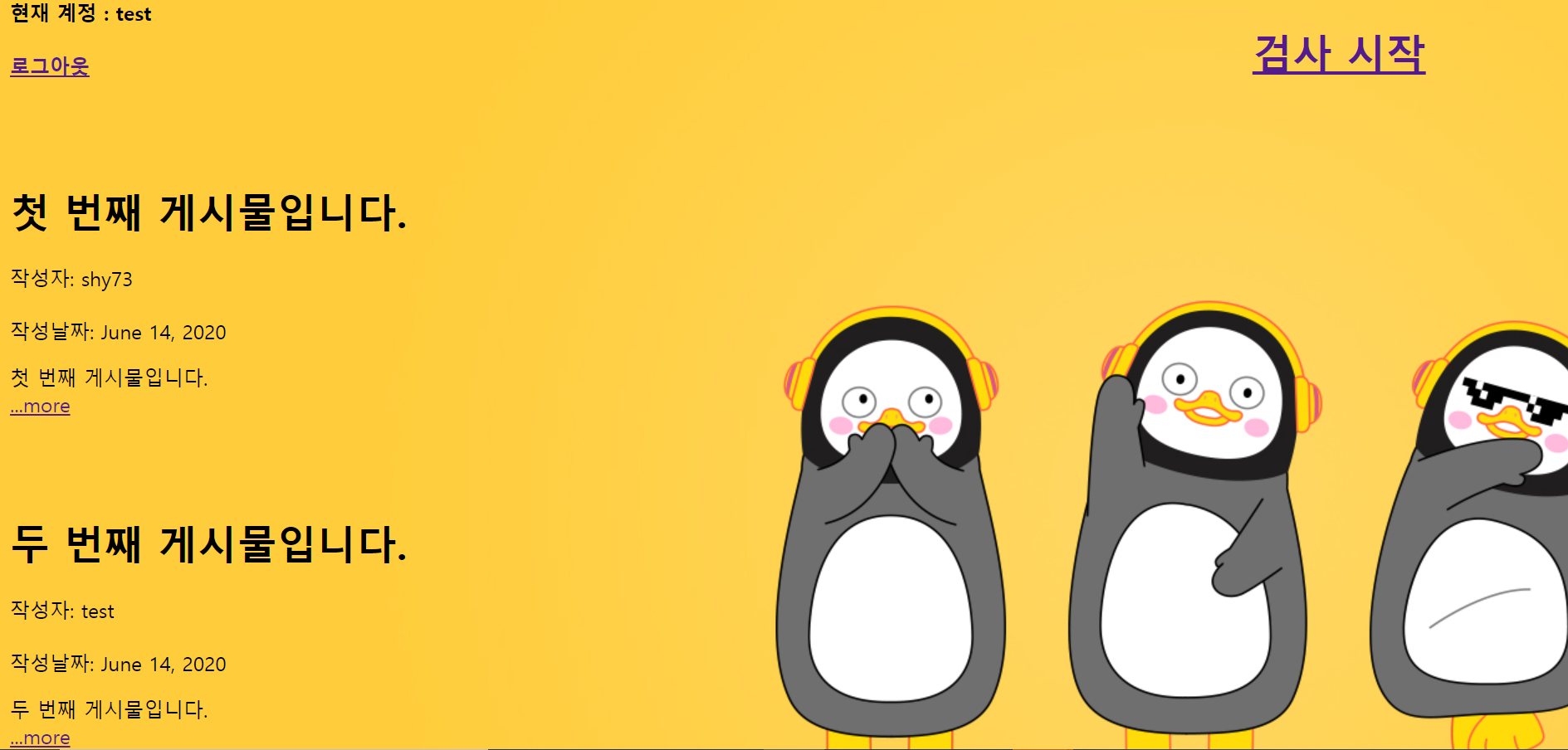
* 1. Function 6: After two tests, users can write the posts about tests results



[그림 11] 검사 결과를 게시물로 작성



[그림 12] 작성된 게시물



[그림 13] 작성된 게시물 홈 화면에서 확인