

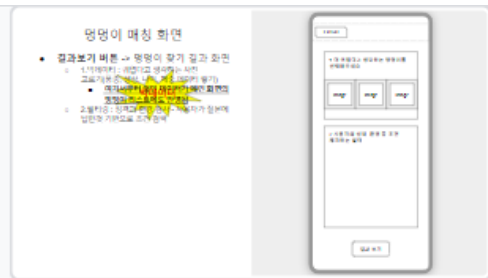
소개해듀개



1



2



3



4



5



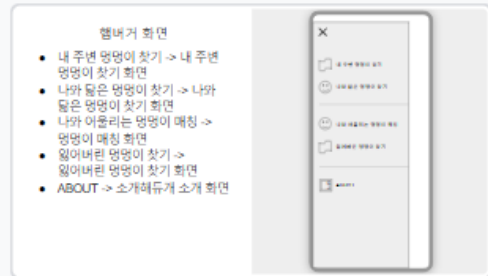
6



7



8



9



10



11



12



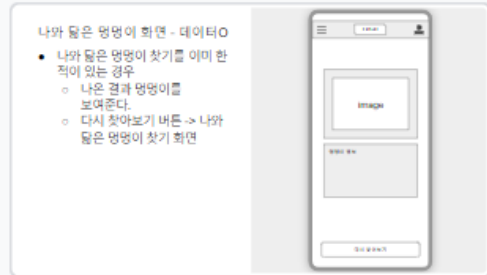
13



14



15



Visual Studio Code interface showing the Explorer, Source Explorer, and Terminal panels.

**EXPLORER**

- FRONTEND
  - build
    - build.js
    - check-versions.js
    - logo.png
    - utils.js
    - vue-loader.conf.js
    - webpack.base.conf.js
    - webpack.dev.conf.js
    - webpack.prod.conf.js
  - config
  - node\_modules
  - src
    - assets
    - components
      - s-header.vue
      - s-mypage-draw...
      - s-navigation-dra...
      - s-navigation.vue
    - layouts
      - s-first-layout.vue
      - s-main-layout.vue
    - pages
      - index.vue
      - login.vue
    - plugins
      - api.js
      - vueify.js
    - router
      - index.js
    - store
      - index.js
    - App.vue
    - main.js
  - static
  - test
  - .babelrc
  - .editorconfig

**Source Explorer**

```
er.vue > {} "s-navigation-drawer.vue" > style > .s-navigation-drawer-header
104 created() {
105   let path = this.$route.path;
106   for (let i = 0; i < this.pathArray.length;
107     if (path == this.pathArray[i].path) {
108       this.selectedMenuId = this.pathArray[i]
109     }
110   }
111 }
112 },
113 };
114 </script>
115
116 <style>
117 .s-navigation-drawer-container {
118   background: #000;
119   padding-left: 20px;
120 }
121
122 .s-navigation-drawer-header {
123   padding: 16px;
124   padding-left: 0;
125   margin-left: -6px;
126 }
127
128 .s-navigation-drawer-bar {
129   margin-top: 32px;
130   border-top: solid 2px #686868;
131 }
132
133 .s-navigation-drawer-menu {
134   padding: 32px 8px 0 0;
135   font-weight: 800;
136   color: #686868;
137   font-size: 18px;
138   line-height: 23px;
139 }
140
141 .s-navigation-drawer-menu--selected {
142   color: #ff501;
143 }
144 </style>
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```

**Terminal**

```
1: node
DONE Compiled successfully in 276ms
Your application is running here: http://localhost:3000
```

Web browser interface showing the application running on localhost:3000. The application displays a navigation drawer with a list of items: "내 주변 멍멍이 찾기", "나와 닮은 멍멍이 찾기", "나와 어울리는 멍멍이 매칭", "잃어버린 멍멍이 찾기", and "ABOUT". The drawer is currently open, showing the "ABOUT" page.

**Navigation Drawer Items:**

- 내 주변 멍멍이 찾기
- 나와 닮은 멍멍이 찾기
- 나와 어울리는 멍멍이 매칭
- 잃어버린 멍멍이 찾기
- ABOUT

**ABOUT Page Content:**

햄버거 화면

- 내 주변 멍멍이 찾기 -> 내 주변 멍멍이 찾기 화면
- 나와 닮은 멍멍이 찾기 -> 나와 닮은 멍멍이 찾기 화면
- 나와 어울리는 멍멍이 매칭 -> 멍멍이 매칭 화면
- 잃어버린 멍멍이 찾기 -> 잃어버린 멍멍이 찾기 화면
- ABOUT -> 소개해듀개 소개 화면

**Footer:**

발표자 노트를 추가하려면 클릭하세요.

User 사용자

user_id	사용자 번호	autoincrease	INT
user_name	사용자 이름	Domain	VARCHAR(128)
user_email	사용자 이메일	Domain	VARCHAR(128)
user_password	사용자 비밀번호	Domain	VARCHAR(128)
user_profile	사용자 사진	Domain	VARCHAR(300)
match_dog	나와 어울리는 강아지의 id	Domain	INT
same_dog	나와 닮은 강아지 사진	null이라면 아직 안했음	VARCHAR(300)

Like 관심 있는 강아지

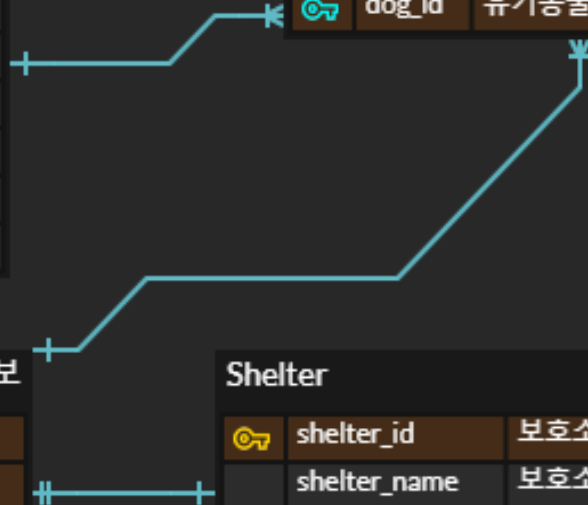
user_id	사용자 번호	autoincrease	INT
dog_id	유기동물 접수 번호 (desertionNo)	Domain	INT

Dog 강아지 정보

dog_id	유기동물 접수 번호 (desertionNo)	Domain	INT
shelter_id	보호소 번호	Domain	INT
age	유기동물 나이. 몇년생 (age)	Domain	INT
weight	유기동물 몸무게 (weight)	Domain	INT
sex	유기동물 성별 (sexCd)	M: 수컷, F: 암컷, Q: 미상	VARCHAR(10)
kind	유기동물 종 (kindCd)	Domain	VARCHAR(128)
color	유기동물 색깔 (colorCd)	Domain	VARCHAR(128)
neuter	중성화 여부 (neuterYn)	Y: 예, N: 아니오, U: 미상	VARCHAR(10)
photo	유기동물 사진 (popfile)	Domain	VARCHAR(300)
special	특징 (specialMark)	Domain	VARCHAR(300)
find_place	발견장소 (happenPlace)	Domain	VARCHAR(300)
find_date	발견 날짜 (happenDt)	Domain	INT
end_date	광고 끝나는 날짜 (noticeEdt)	Domain	INT

Shelter 강아지 보호소

shelter_id	보호소 번호	Domain	INT
shelter_name	보호소 이름 (careNm)	Domain	VARCHAR(128)
shelter_address	보호소 위치 (careAddr)	Domain	VARCHAR(300)
shelter_tel	보호소 전화번호 (careTel)	Domain	VARCHAR(128)
charge_name	관리자 이름 (chargeNm)	Domain	VARCHAR(128)



A	B	C	D	E	F	G
Group	METHOD	RESOURCE	DESCRIPTION	예상 INPUT	예상 OUTPUT	INPUT
회원	POST	/accounts/signup	회원가입	사용자 이름 이메일		{ "user_name" : "jin"
	POST	/accounts/profile	회원가입할 때 사용자 사진 업로드 받기	사용자 사진		
	POST	/accounts/login	로그인	이메일 패스워드		{ "user_email" : "aa"
	POST	/accounts/logout	로그아웃			
	GET	/accounts/islogin	백에서 세션 있는지 확인	-	장고에서 session이 비어있으면 오류를 넘 null이라고 출력 아해주고 !	
	GET	/accounts/mypage	유저 프로필 보기	-	사용자 이름, 사진, email 정도..?	
메인 화면	POST	/filters	필터가 적용된 리스트 보이기	필터 내용들.. (단모, 색깔.. 등등)	강아지들 리스트...	
나와 어울리는 멍멍이 찾기	POST	/matches	나와 어울리는 멍멍이 조건들 입력하기	사진 들 중 귀엽다 생각한 멍멍이를 id 여러개 배열 필터에 특어가 조건들 (사용자 성격, 환경 등등)		
	GET	/matches	나와 어울리는 멍멍이 결과 출력하기		멍멍이 사진 멍멍이 정보를 (나이, 몸무게, 성별, 종, 색깔)	
잃어버린 멍멍이 찾기	POST	/losts	잃어버린 멍멍이 사진 업로드 받기	멍멍이 사진		
	GET	/losts/{my_location}	내 위치 유기 멍멍이를 리스트 받기		간략한 강아지 정보를 (사진, 이름) => 누르면 자세히 보기로 간	
	PUT	/losts	위치 검색해서 수정된 멍멍이 리스트 받기	검색하고싶은 위치	간략한 강아지 정보를 (사진, 이름) => 누르면 자세히 보기로 간	
내 주변 멍멍이 찾기	GET	/arounds	내 위치 기준 보호소 정보 받기		위도, 경도, 보호소 이름	
	POST	/arounds	보호소 이름 받으면 그 보호소 내 멍멍이 리스트	보호소 이름	간략한 강아지 정보를 (사진, 이름) => 누르면 자세히 보기로 간	
나와 닮은 멍멍이 찾기	POST	/resemblances	내 이미지 업로드하기	내 이미지		
	GET	/resemblances/{profile_img}	그 이미지와 어울리는 강아지 정보 주기		이건 유기견이 아니져? 그럼 그냥 강아지 사진, 종 2 정도	
관심있는 멍멍쓰	POST	/likes	관심 멍멍 누르기	유기동물 접수 번호 (dog_id)		
	GET	/likes	관심 멍멍 리스트		멍멍이 사진 멍멍이 정보를 (나이, 몸무게, 성별, 종, 색깔)	
공통 화면	GET	/details/{dog_id}	멍멍이 자세히 보기		멍멍이 사진 멍멍이 정보를 (나이, 몸무게, 성별, 종, 색깔)	

```

5 class User(models.Model):
6     user_id = models.AutoField(primary_key=True) #사용자 id
7     user_name = models.CharField(max_length=128, null=True) #사용자 이름
8     user_email = models.CharField(max_length=128, null=True) #사용자 email
9     user_password = models.CharField(max_length=128, null=True) #사용자 password
10    user_profile = models.CharField(max_length=300, null=True) #사용자 프로필 사진
11    match_dog = models.IntegerField( null=True) #사용자와 잘 어울리는 강아지 id
12    same_dog = models.CharField(max_length=300, null=True) #사용자와 닮은 강아지 사진

```

The screenshot displays a REST client interface with a sidebar on the left and a main workspace on the right.

**Left Sidebar:**

- History:** Contains a list of recent requests. The first three are POST requests to `http://localhost:8000/accounts/signup`. Below them, under the heading "Yesterday", are several more requests, including POST requests to `http://localhost:8000/accounts/signup` and `http://localhost:8000/user/signup`.
- Collections:** Empty.
- APIs:** Empty.
- Save Responses:** A toggle switch is currently turned off.
- Clear all:** A button to clear the history.

**Main Workspace:**

- Untitled Request:** The top section shows a POST request to `http://localhost:8000/accounts/signup` with a "Send" button.
- Body Tab:** The request body is formatted as JSON:
 

```

{
  "user_name": "jin4",
  "user_email": "dd",
  "user_password": "aaAA11!!",
  "user_profile": "jin.jpg",
  "match_dog": 12345,
  "same_dog": "jin_same_dog.jpg"
}

```
- Response Section:** Below the request, the response is shown. The status is 200 OK, with a time of 823 ms and a size of 228 B. The response body is:
 

```

{
  "message": "회원가입 성공!"
}

```

```

23 model = load_model("../model/multi_img_classification_model")
24
25 prediction = model.predict(X)
26 np.set_printoptions(formatter={'float': lambda x: "{0:0.3f}".format(x)})
27 cnt = 0
28
29 #이 비교는 그냥 파일들이 있으면 해당 파일과 비교. 카테고리랑 함께
30 for i in prediction:
31     pre_ans = i.argmax() # 예측 레이블
32     print(i)
33     print(pre_ans)
34     pre_ans_str = ""
35     if pre_ans == 0: pre_ans_str = "포메"
36     elif pre_ans == 1: pre_ans_str = "말티즈"
37     elif pre_ans == 2: pre_ans_str = "치와와"
38     elif pre_ans == 3: pre_ans_str = "토이푸들"
39     else: pre_ans_str = "골든리트리버"
40     if i[0] >= 0.8: print("해당 "+filenames[cnt].split("/")[2]+"0")
41     if i[1] >= 0.8: print("해당 "+filenames[cnt].split("/")[2]+"0")
42     if i[2] >= 0.8: print("해당 "+filenames[cnt].split("/")[2]+"0")
43     #print(filenames[cnt].split("/"))
44     if i[3] >= 0.8: print("해당 "+filenames[cnt].split("/")[2]+"0")
45     if i[4] >= 0.8: print("해당 "+filenames[cnt].split("/")[2]+"0")
46     cnt += 1
47     # print(i.argmax()) #애가 레이블 [1. 0. 0.] 이런식으로 되어 있
48     # 즉 애랑, 나중에 카테고리 데이터 불러와서 카테고리랑 비교를 하
49     # 이걸 한 것은 _4.py에.
50

```

```
[0.000 0.000 0.000 0.000 1.000]
```

```
4
```

```
해당 포메1.png이미지는 골든리트리버으로 추정됩니다.
```

```
[0.000 0.000 0.000 0.003 0.997]
```

```
4
```

```
해당 골리1.jpg이미지는 골든리트리버으로 추정됩니다.
```

```
[0.000 1.000 0.000 0.000 0.000]
```

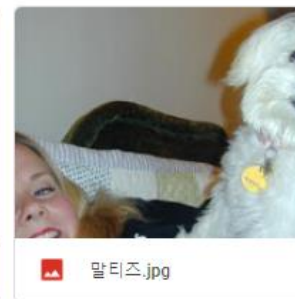
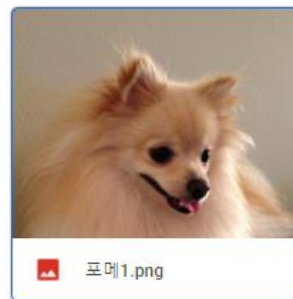
```
1
```

```
해당 말티즈.jpg이미지는 말티즈으로 추정됩니다.
```

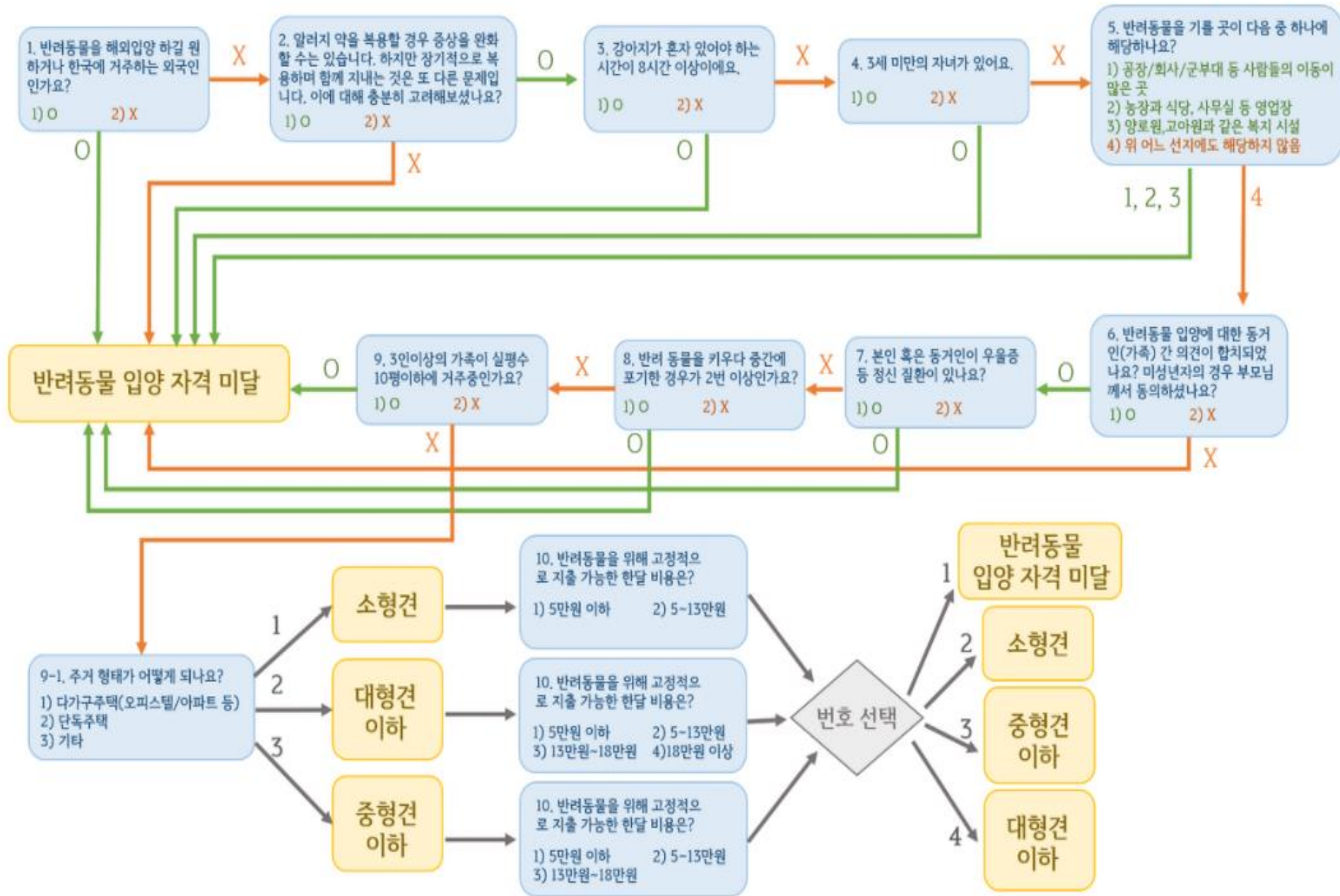
🔍 드라이브에서 검색

내 드라이브 > DOG\_Reco > img\_other\_test ▾

파일









kind	성격
말티즈	젠틀, 장난, 매력
진도견	경고, 지능, 대담
푸들	다정, 친절, 독립, 민첩, 지능, 자신감
포메라니안	호기심, 대담, 생동감
치와와	차밍, 우아함, 콧대
스피츠	친절, 자연, 활기, 헌신, 생동감, 세심
시츄	다정, 장난, 외향
닥스훈트	친절, 호기심, 섹시
골든 리트리버	친절, 지능, 헌신
요크셔 테리어	다정, 멋짐, 말괄량이
시베리안 허스키	충성, 외향, 장난
라브라도 리트리버	친절, 활동, 외향
웰시 코기 펴브로크	다정, 똑똑, 경고
보스턴 테리어	친절, 밝음, 재미
코카 스파니엘	젠틀, 똑똑, 행복
발바리	
프렌치 불독	적응, 장난, 똑똑
비글	친절, 호기심, 명량
미니어처 핀셔	겁없는, 재미, 자랑
비숑 프리제	장난, 호기심, 스릴
보더 콜리	다정, 영리, 활기

```
In [79]: sim_index = sim_index[sim_index != target_dog_index]
```

```
In [81]: result = data.iloc[sim_index].sort_values('popularity', ascending=False)[:10]
```

```
In [82]: result
```

Out [82]:

	breed	temper	popularity	height	weight	life_min	size	life_max	GROOMING - Grooming Frequency (%)	GROOMING - Shedding (%)	EXERCISE - Energy Level (%)	TRAINING - Trainability (%)	TRAINING - Temperament/Demeanor (%)
9	요크셔 테리어	다정 멋짐 말괄량이	9	7-8 inches	7 pounds	11	S	15	100%	20%	60%	20%	80%
2	토이 푸들	민첩 지능 자신감	7	No more than 10 inches	4-6 pounds	10	S	18	100%	20%	60%	100%	80%
17	미니 어처 핀셔	겁없는 재미 자랑	68	10-12.5 inches	8-10 pounds	12	S	16	20%	NaN	60%	40%	60%
15	프렌치 불독	적응 장난 포퓰러	6	11-13 inches	under 28 pounds	10	S	12	20%	80%	40%	60%	60%

```
In [48]: data.columns
```

Out [48]: Index(['breed', 'temper', 'popularity', 'height', 'weight', 'life\_min', 'size', 'life\_max', 'GROOMING - Grooming Frequency (%)', 'GROOMING - Shedding (%)', 'EXERCISE - Energy Level (%)', 'TRAINING - Trainability (%)', 'TRAINING - Temperament/Demeanor (%)'], dtype='object', length=13)