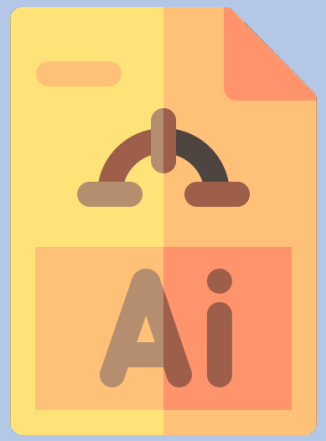
PinPlace: CNN based location image search And its adaptation to social network

TEAM H Week 9

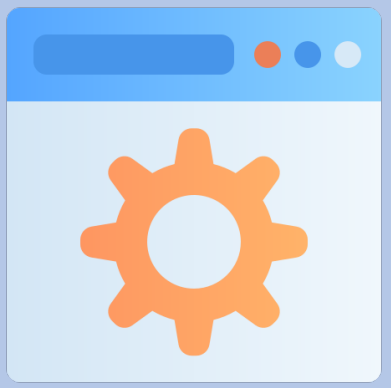


CNN Build

- Data processing
  HONG SEONGJUN
- Modify CNN model & improve accuracy
  CHE SEUNG YUN

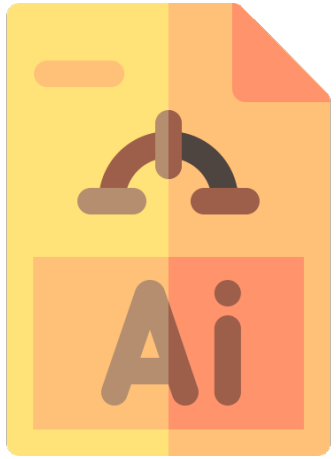
Front end

- List up page & Search Location page
  JEONG CHAEWON, LEE JI SEOP



Back end

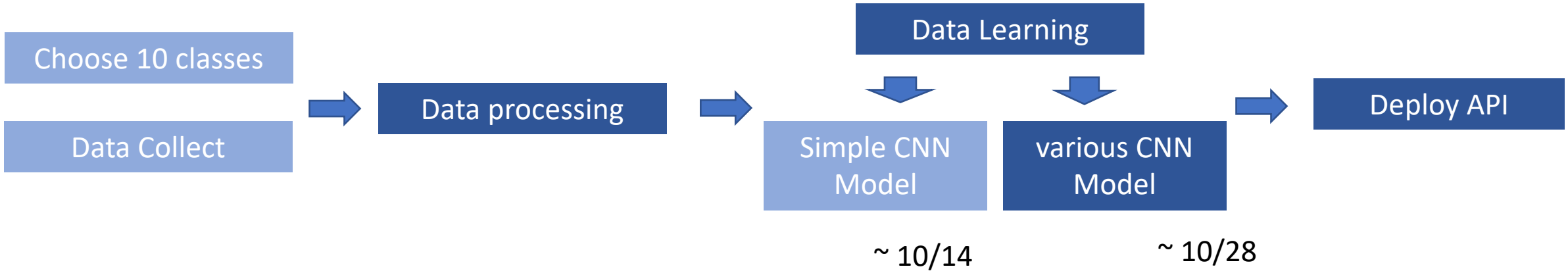
- Test CNN model in web application
  UHM JI YONG



CNN Build

Our CNN model development process

1. Collect and process data necessary for learning.
2. Train an appropriate artificial intelligence model using the processed learning data.
3. Deploy the trained model to utilize it in application.



CNN Model build > Week 9

Data processing

```
data_aug_gen = ImageDataGenerator(rescale=1./255,
                                  rotation_range=15,
                                  width_shift_range=0.1,
                                  height_shift_range=0.1,
                                  shear_range=0.5,
                                  zoom_range=[0.8, 2.0],
                                  horizontal_flip=True,
                                  vertical_flip=False,
                                  fill_mode='nearest')
```


```
(15468, 128, 128, 3)
15468
```

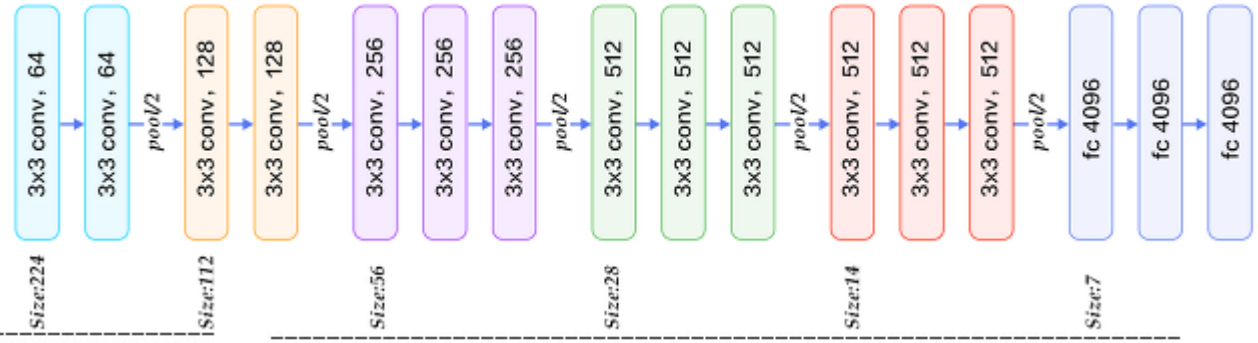
- We collect specific places images.
- We downloaded Instagram images manually.
- We used data augmentation and save the images.

- We tried kfold(cross validation), but because of limits of ram capacity, we did learn model in classic way.

CNN Model build > Week 9

Modified CNN model & test it

- We tried VGG16 model



Layer (type)	Output Shape	Param #
conv2d_39 (Conv2D)	(None, 128, 128, 64)	1792
conv2d_40 (Conv2D)	(None, 128, 128, 64)	36928
max_pooling2d_15 (MaxPooling	(None, 64, 64, 64)	0
conv2d_41 (Conv2D)	(None, 64, 64, 128)	73856
conv2d_42 (Conv2D)	(None, 64, 64, 128)	147584
max_pooling2d_16 (MaxPooling	(None, 32, 32, 128)	0
conv2d_43 (Conv2D)	(None, 32, 32, 256)	295168
conv2d_44 (Conv2D)	(None, 32, 32, 256)	590080
conv2d_45 (Conv2D)	(None, 32, 32, 256)	590080
max_pooling2d_17 (MaxPooling	(None, 16, 16, 256)	0
conv2d_46 (Conv2D)	(None, 16, 16, 512)	1180160
conv2d_47 (Conv2D)	(None, 16, 16, 512)	2359808
conv2d_48 (Conv2D)	(None, 16, 16, 512)	2359808
max_pooling2d_18 (MaxPooling	(None, 8, 8, 512)	0
conv2d_49 (Conv2D)	(None, 8, 8, 512)	2359808
conv2d_50 (Conv2D)	(None, 8, 8, 512)	2359808
conv2d_51 (Conv2D)	(None, 8, 8, 512)	2359808
max_pooling2d_19 (MaxPooling	(None, 4, 4, 512)	0
flatten_3 (Flatten)	(None, 8192)	0
dense_8 (Dense)	(None, 4096)	33558528
dense_9 (Dense)	(None, 4096)	16781312
dense_10 (Dense)	(None, 4096)	16781312
dense_11 (Dense)	(None, 10)	40970

Total params: 81,876,810
Trainable params: 81,876,810
Non-trainable params: 0

- But, it has so many parameters to learn. It can not be learned in google colab because of limits of ram capacity

Modified CNN model & test it

- So, we tried Resnet50 model

```
model = ResNet50V2(include_top=True, weights=None, input_shape=(128,128,3), classes=10)
model.compile(loss='categorical_crossentropy', optimizer='Nadam', metrics=['accuracy'])
```

```
conv5_block3_2_conv (Conv2D)    (None, 4, 4, 512)    2359296    conv5_block3_2_pad[0][0]
conv5_block3_2_bn (BatchNormali (None, 4, 4, 512)    2048       conv5_block3_2_conv[0][0]
conv5_block3_2_relu (Activation (None, 4, 4, 512)    0          conv5_block3_2_bn[0][0]
conv5_block3_3_conv (Conv2D)    (None, 4, 4, 2048)   1050624    conv5_block3_2_relu[0][0]
conv5_block3_out (Add)          (None, 4, 4, 2048)   0          conv5_block2_out[0][0]
                                                                conv5_block3_3_conv[0][0]
                                                    )2         conv5_block3_out[0][0]
post_relu (Activation)          (None, 4, 4, 2048)   0          post_bn[0][0]
avg_pool (GlobalAveragePooling2 (None, 2048)         0          post_relu[0][0]
predictions (Dense)             (None, 10)           20490      avg_pool[0][0]
Total params: 23,585,290
Trainable params: 23,539,850
Non-trainable params: 45,440
```

- We set train set, validation set, test set as 6:2:2

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
```

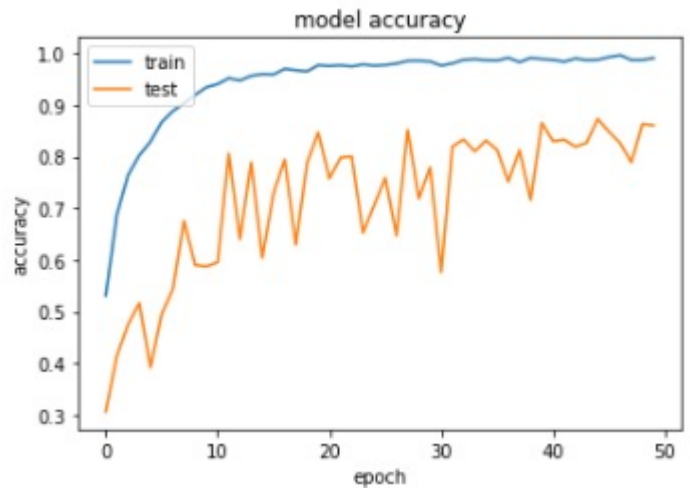
```
history = model.fit(X_train, y_train, batch_size=32, epochs=50, validation_split=0.2)
```

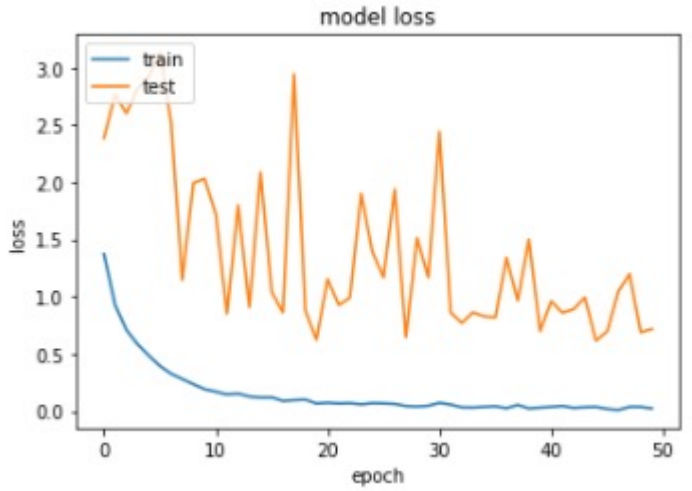
Modified CNN model & test it

- Accuracy of model is 86.61%

```
#모델 정확도 출력
print("정확도 : %.4f" % (model.evaluate(X_test, y_test)[1]))
```

```
121/121 [==============================] - 4s 30ms/step - loss: 0.7465 - accuracy: 0.8661
정확도 : 0.8661
```





Next week

1. Increasing test set size & Modify another model & Data augmentation

   ↳ CHE SEUNG YUN

2. Collect more images as possible

   ↳ HONG SEONG JUN

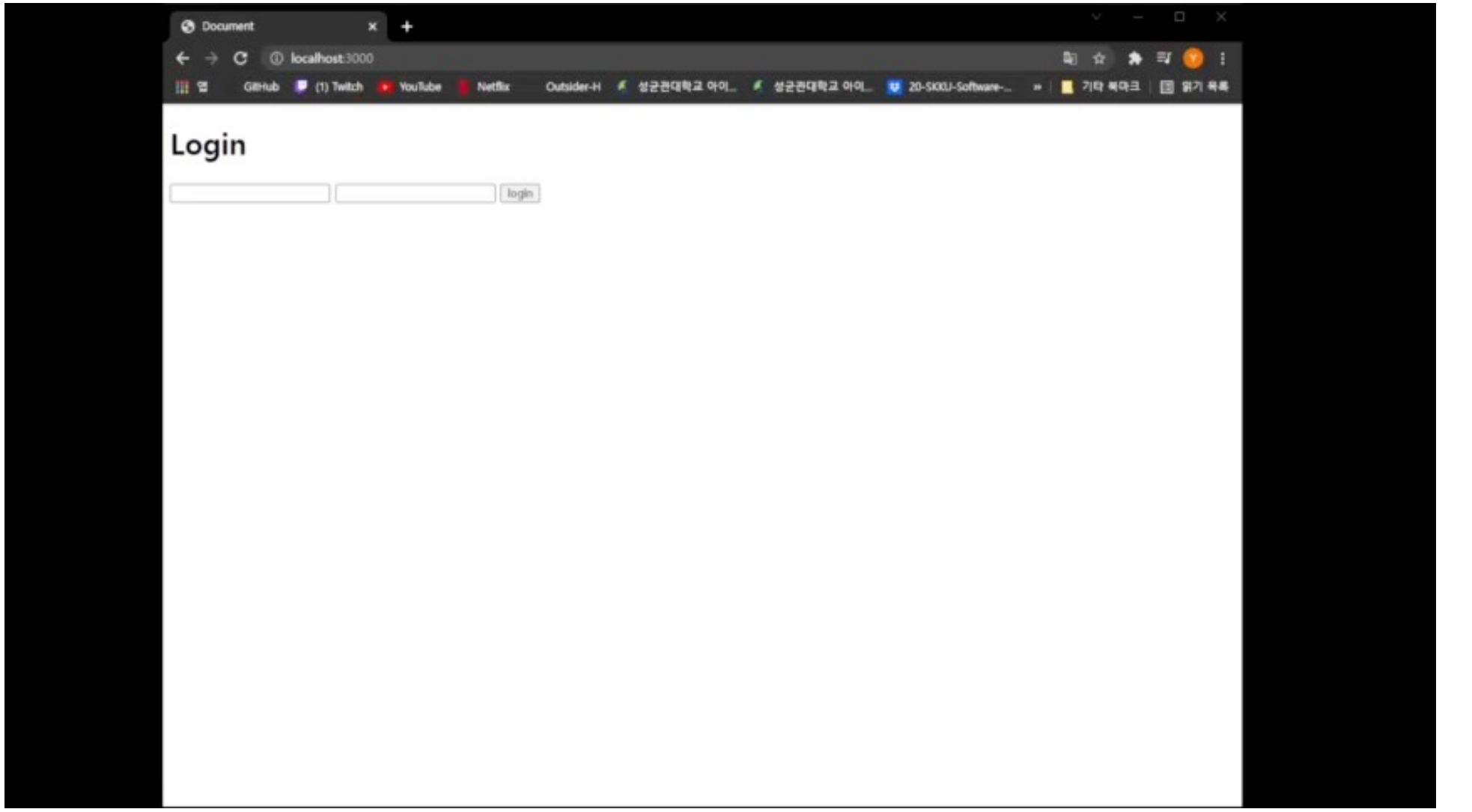
3. Select the final model & Check the model works well

   ↳ CHE SEUNG YUN, UHM JI YONG

Test CNN in web app

Back end

Demo



Front end

- *@Collaboration Work*

Confirm Concept
- ~10/17

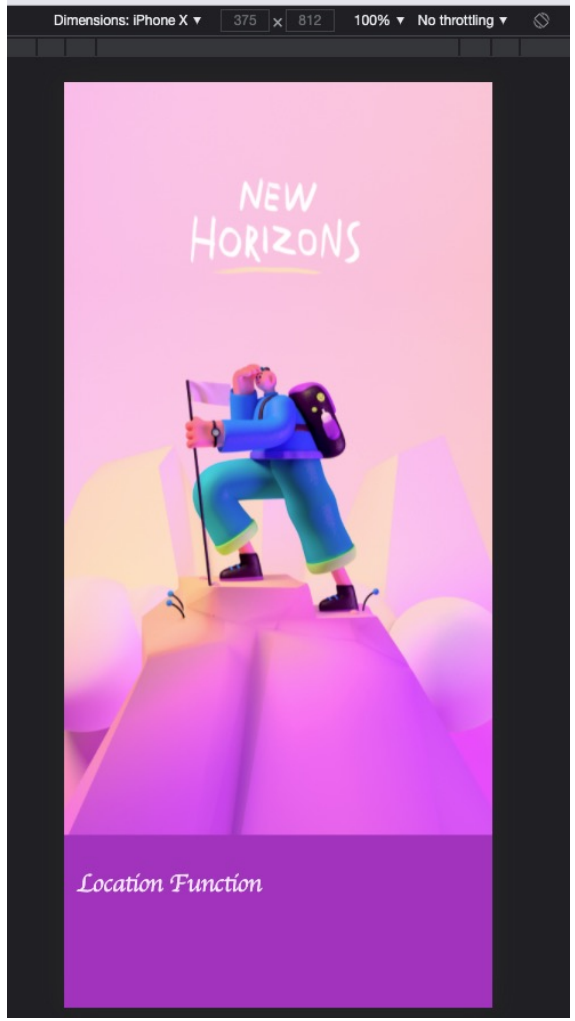
- *@Collaboration Work*

Make Web Structure
- ~10/24

- *@Individual Work*

Make Specific Web page
- ~10/31

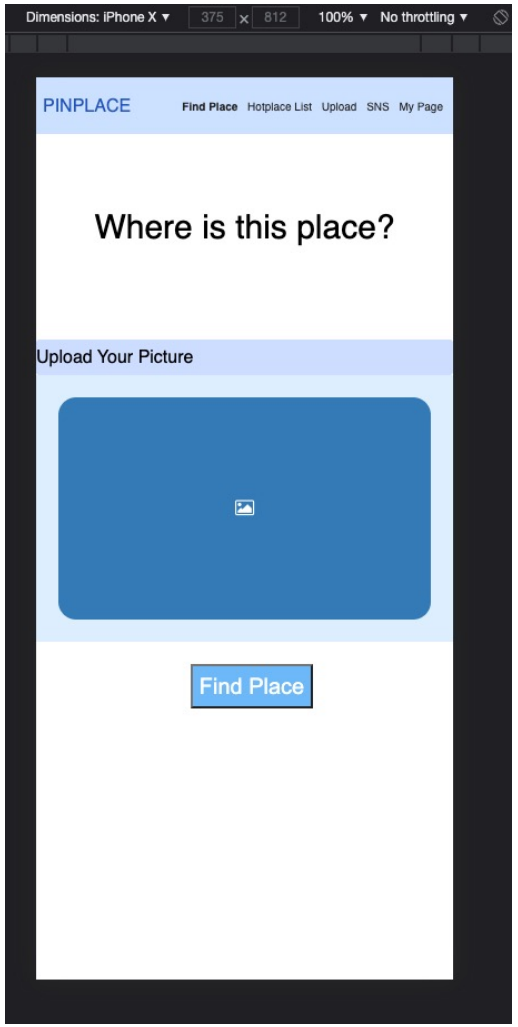
Guide Page

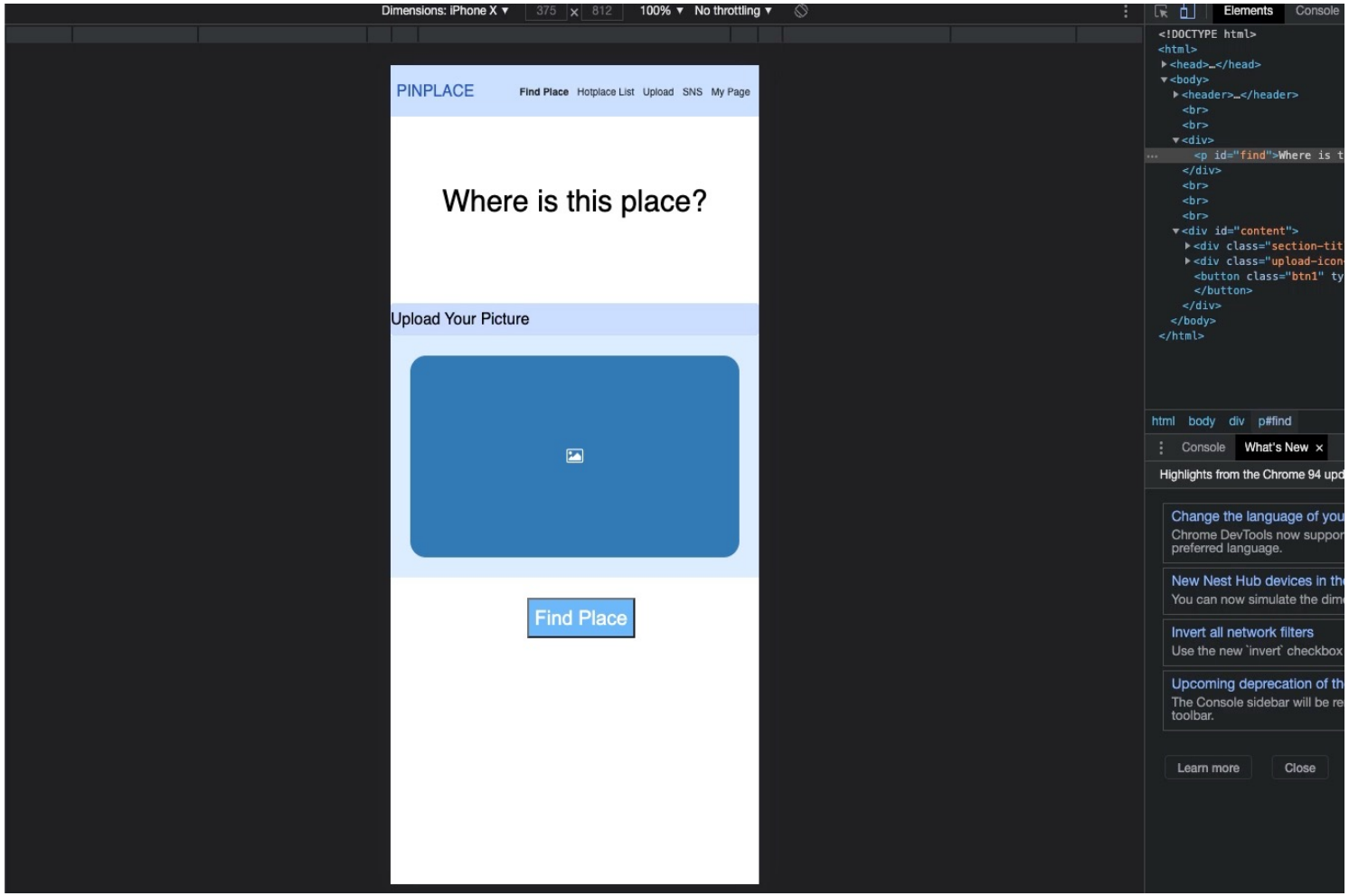


- Check Reference & Design
- Collect 3D Graphic assets
- Develop page’s structure

Reference

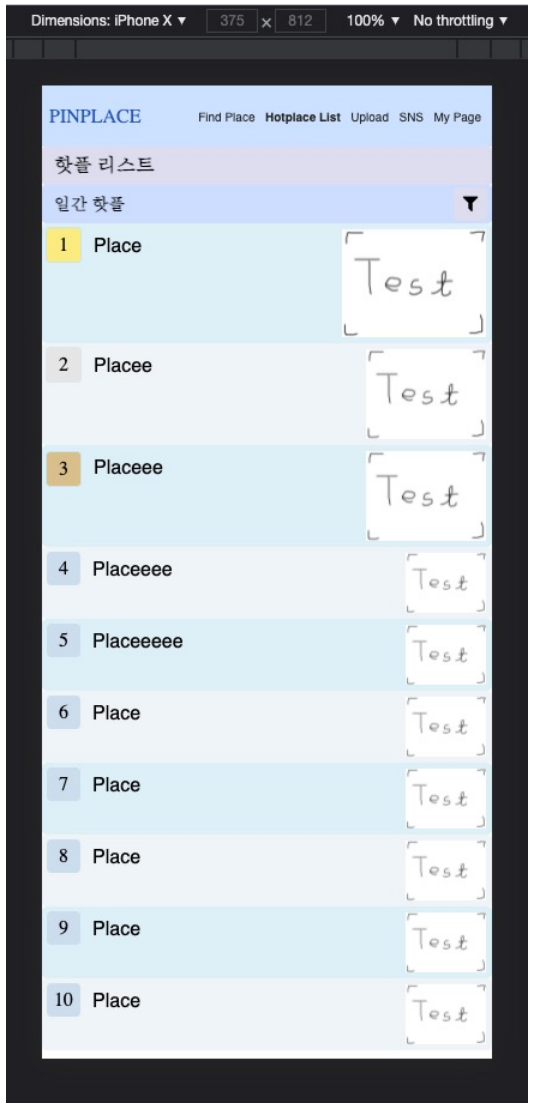
Location Search Page

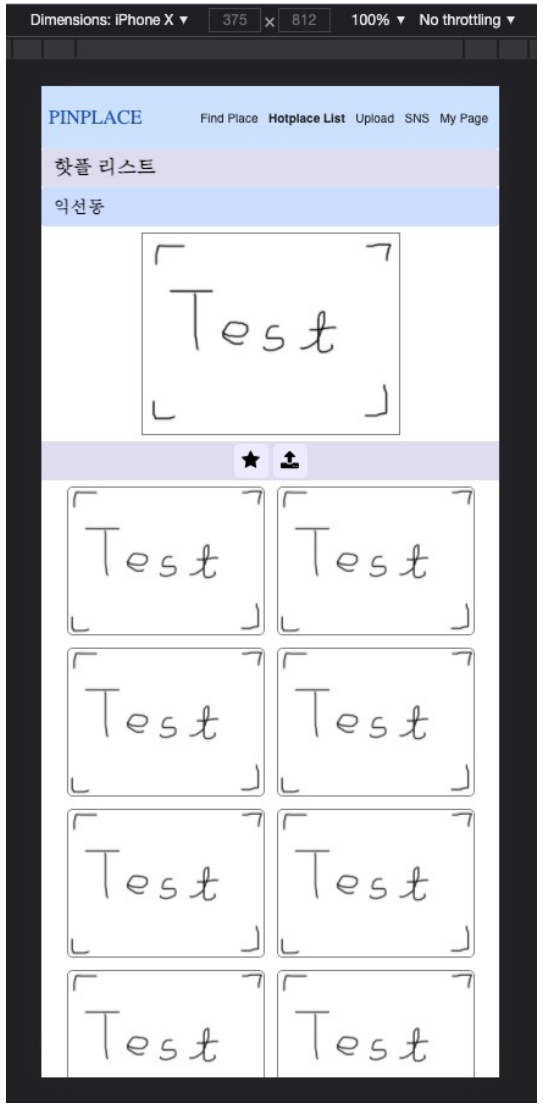




List Up Page


Dimensions: iPhone X 375 x 812 100% No throttling
PINPLACE
Find Place Hotplace List Upload SNS My Page
기억해줘요
사진을 올려주세요
여기는 어디인가요?
지도가 들어갈 공간입니다.
업로드
취소


Dimensions: iPhone X 375 x 812 100% No throttling
PINPLACE
Find Place Hotplace List Upload SNS My Page
핫플 리스트
일간 핫플
1 Place
2 Placee
3 Placeee
4 Placeeee
5 Placeeeee
6 Place
7 Place
8 Place
9 Place
10 Place
Test


Dimensions: iPhone X 375 x 812 100% No throttling
PINPLACE
Find Place Hotplace List Upload SNS My Page
핫플 리스트
익선동
Test

Next week

1. Finish the development of Guide Page

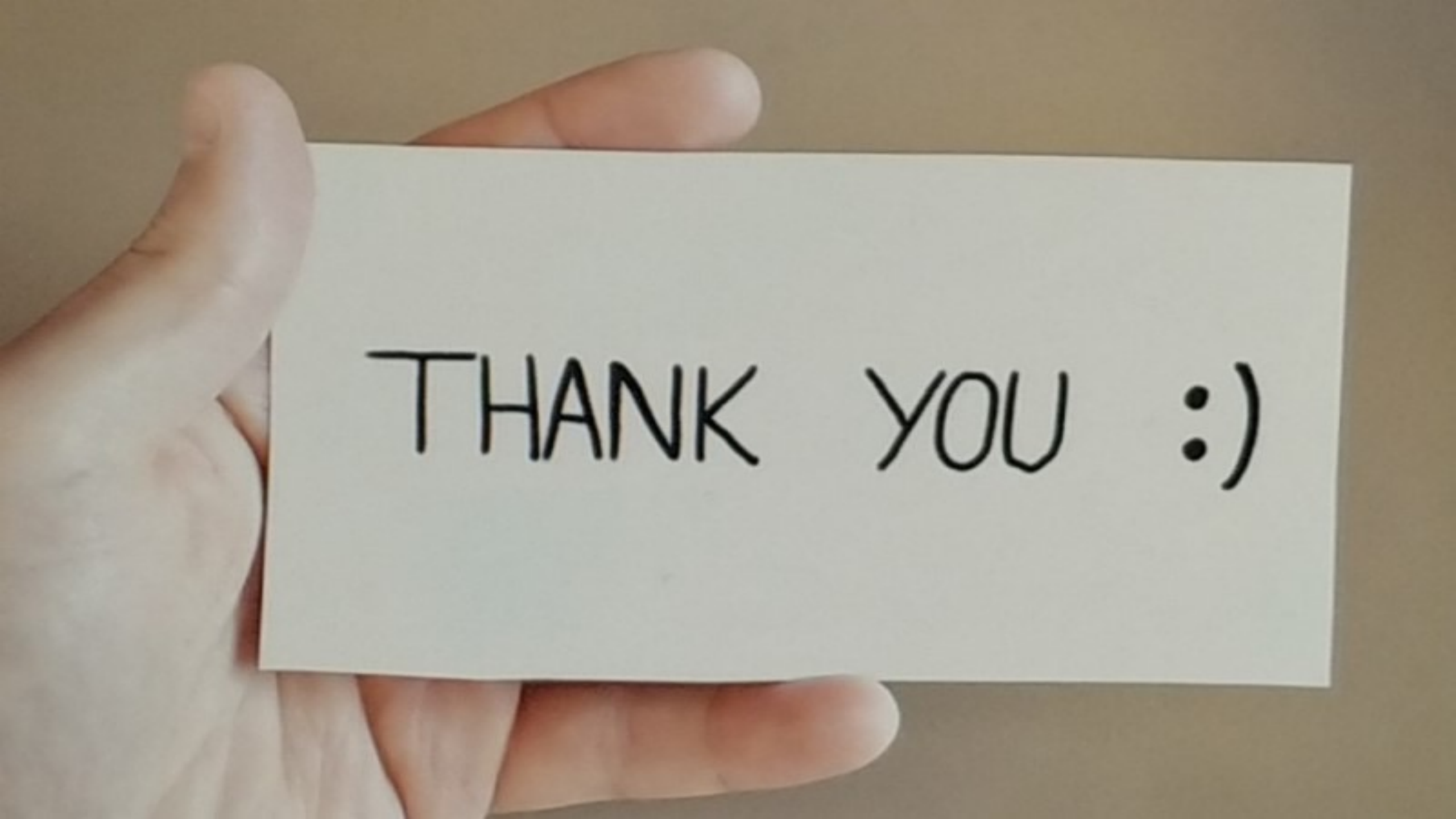
   ↳ Chaewon Jeong

2. Finish the development of Find Location Page

   ↳ Chaewon Jeong

3. Finish the development of List up Page

   ↳ Jiseop Lee


THANK YOU :)