In [45]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix
from google.colab import drive
drive.mount('/content/drive')

%matplotlib inline
face = pd.read_csv("/content/drive/MyDrive/156 Proj/data/face.csv") # make sure
face.head()
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call dr
ive.mount("/content/drive", force_remount=True).
```

Out[45]:

	emotion	pixels
0	0	70 80 82 72 58 58 60 63 54 58 60 48 89 115 121...
1	0	151 150 147 155 148 133 111 140 170 174 182 15...
2	2	231 212 156 164 174 138 161 173 182 200 106 38...
3	4	24 32 36 30 32 23 19 20 30 41 21 22 32 34 21 1...
4	6	4 0 0 0 0 0 0 0 0 0 0 0 0 3 15 23 28 48 50 58 84...

In [46]:

```
face["pixels"] = face.pixels.apply(lambda x: np.array(list(map(int, x.split()))))
emo_di = {0: "Angry", 1:"Disgust", 2: "Fear", 3: "Happy", 4: "Sad", 5: "Surprise
plt.figure(figsize = (20,20))
for i in range(7):
    plt.subplot(1, 7, i+1)
    plt.grid(False)
    plt.xticks([])
    plt.yticks([])
    plt.imshow(face[face.emotion == i].pixels.iloc[20].reshape(48, 48), cmap='gr
    plt.xlabel("{} = {}".format(i, emo_di[i]))
```

In [47]:

```
face_ = face.copy()
train_x, test_x, train_y, test_y = train_test_split(np.apply_along_axis(list, 0,
val_x, test_x, val_y, test_y = train_test_split(test_x, test_y, train_size=0.5,
```

Logistic Regression

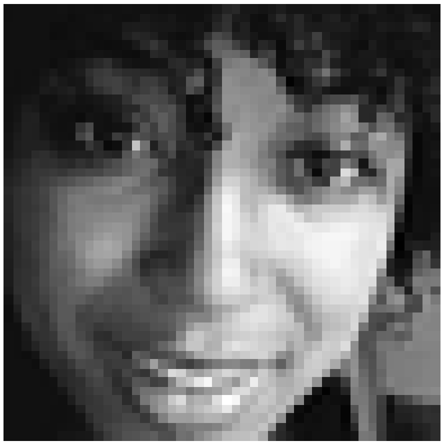
In [48]:

```
#scaling the features
train_x_scaled = train_x/255.0
test_x_scaled = test_x/255.0
from sklearn.linear_model import LogisticRegression
lr = LogisticRegression(penalty='none', tol=0.1, solver='saga', multi_class='mul
```

In [49]:

```
#sample number
sample_idx = 117
#plotting image
test_sample = test_x_scaled[sample_idx].reshape(48,48)
pred_y_prob = lr.predict_proba(test_x_scaled)
pred_y = lr.predict(test_x_scaled)

plt.imshow(test_sample, cmap='gray');
plt.title('Label: %s\n' % test_y.iloc[sample_idx]);
plt.axis('off');
```

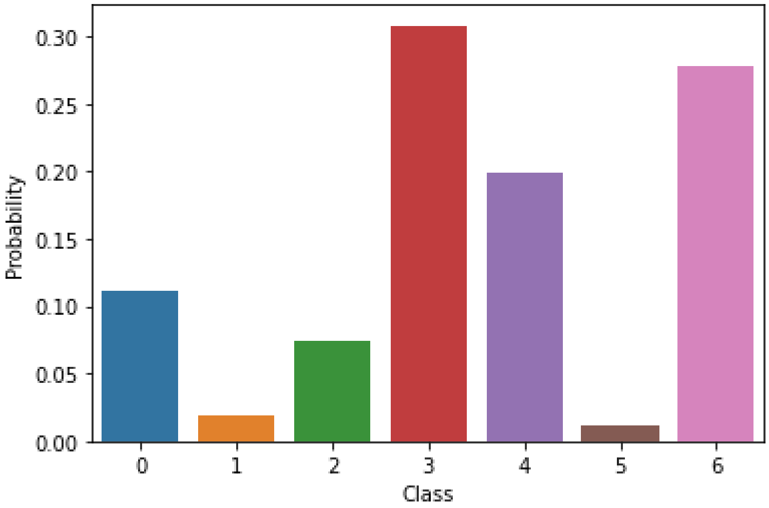
Label: 3



In [50]:

```
z = [lr.intercept_[k] + np.dot(lr.coef_[k], test_x_scaled[sample_idx]) for k in
#conditional probability
exps = [np.exp(z[k]) for k in range(7)]
exps_sum = np.sum(exps)
probs = exps/exps_sum
probs
#plot
sns.barplot(np.arange(0,7), probs);
plt.ylabel("Probability");
plt.xlabel("Class");
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning:
Pass the following variables as keyword args: x, y. From version 0.12, the only
valid positional argument will be `data`, and passing other arguments without an
explicit keyword will result in an error or misinterpretation.
  FutureWarning
```



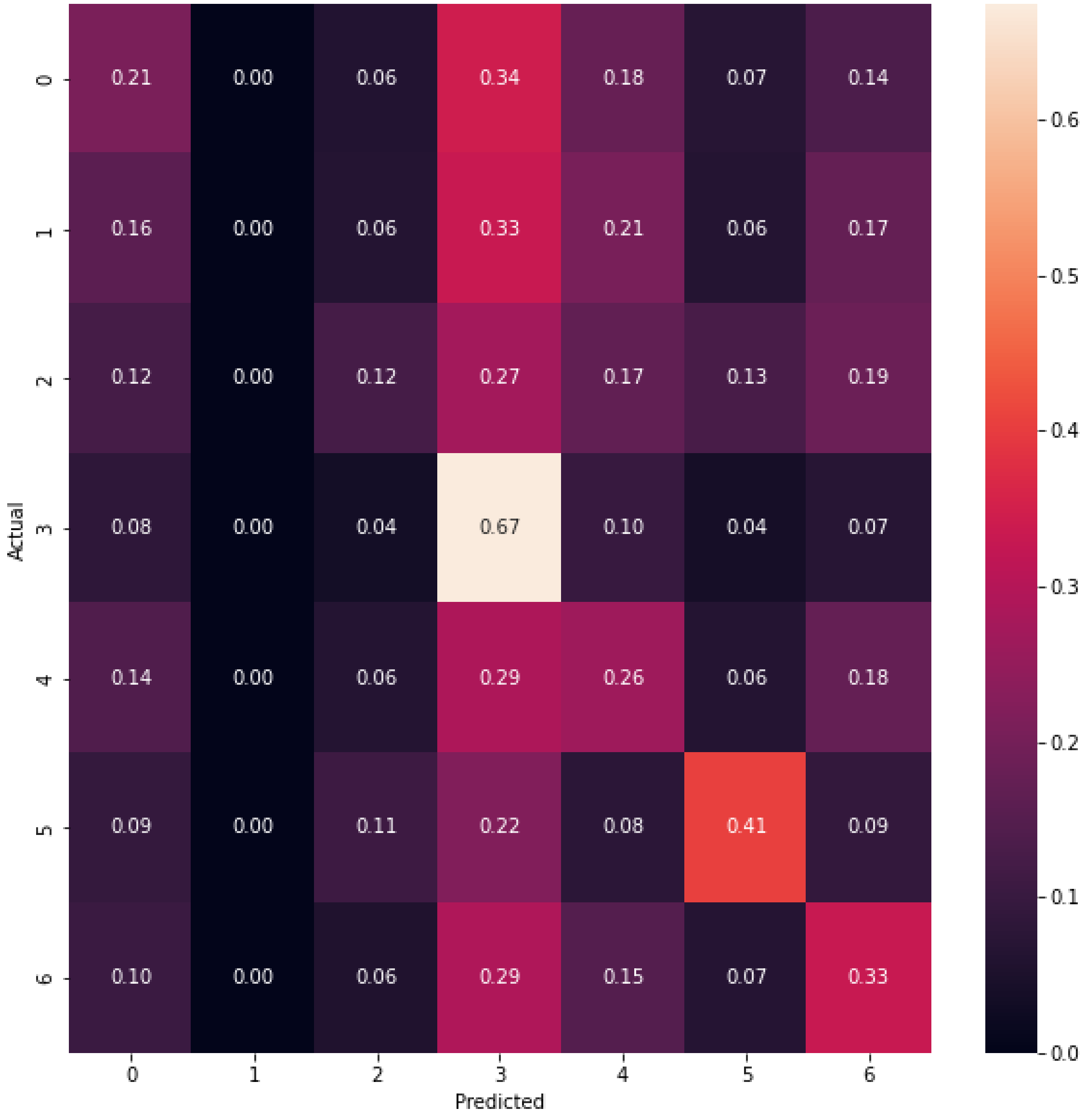
In [51]:

```
accuracy = lr.score(test_x_scaled, test_y)
print(accuracy)
```

```
0.3673090318086835
```

In [52]:

```
cm = pd.crosstab(test_y, pred_y, rownames=['Actual'], colnames=['Predicted'],nor
p = plt.figure(figsize=(10,10));
p = sns.heatmap(cm, annot=True, fmt=".2f", cbar=True)
```

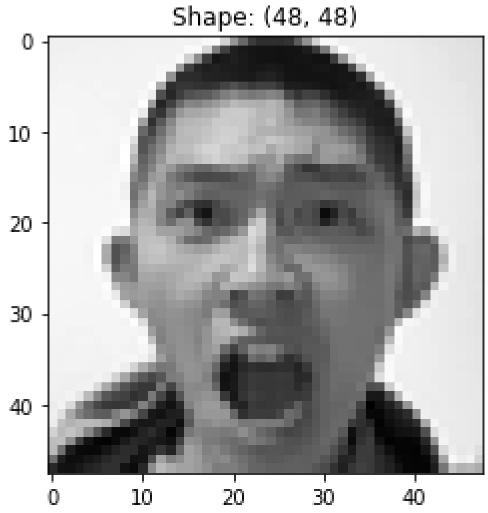


In [53]:

```
from google.colab import files
from PIL import Image

uploaded = files.upload()

filename = 'testmine.jpg'

image = Image.open(filename)
p = plt.imshow(np.asarray(image), cmap=plt.cm.gray,);
p = plt.title('Shape: ' + str(np.asarray(image).shape))

# convert to grayscale image - 'L' format means each pixel is
# represented by a single value from 0 to 255
image_bw = image.convert('L')
p = plt.imshow(np.asarray(image_bw), cmap=plt.cm.gray,);
p = plt.title('Shape: ' + str(np.asarray(image_bw).shape))
# resize image
image_bw_resized = image_bw.resize((48,48), Image.ANTIALIAS)
p = plt.imshow(np.asarray(image_bw_resized), cmap=plt.cm.gray,);
p = plt.title('Shape: ' + str(np.asarray(image_bw_resized).shape))
test_sample = np.array(image_bw_resized).reshape(1,2304)
```

Choose Files No file chosen Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

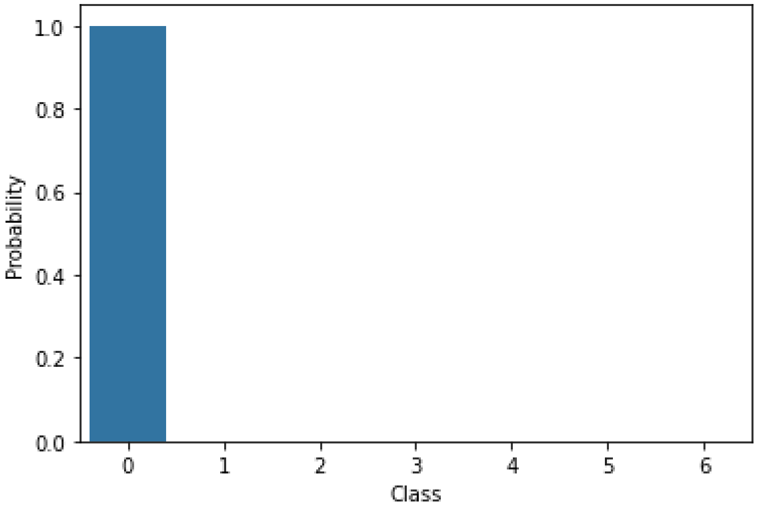
```
Saving testmine.jpg to testmine (3).jpg
```



In [54]:

```python
test_probs = lr.predict_proba(test_sample)

sns.barplot(np.arange(0,7), test_probs.squeeze());
plt.ylabel("Probability");
plt.xlabel("Class");
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning:
Pass the following variables as keyword args: x, y. From version 0.12, the only
valid positional argument will be `data`, and passing other arguments without an
explicit keyword will result in an error or misinterpretation.
  FutureWarning
```


Probability
Class