In [16]:
```
!pip install tensorflow
```

```
Requirement already satisfied: tensorflow in c:\users\lenovo\anaconda3\lib\site-packages (2.14.0)
Requirement already satisfied: tensorflow-intel==2.14.0 in c:\users\lenovo\anaconda3\lib\site-packages (from tensorflow) (2.14.
0)
Requirement already satisfied: tensorboard<2.15,>=2.14 in c:\users\lenovo\anaconda3\lib\site-packages (from tensorflow-intel==
2.14.0->tensorflow) (2.14.1)
Requirement already satisfied: flatbuffers>=23.5.26 in c:\users\lenovo\anaconda3\lib\site-packages (from tensorflow-intel==2.1
4.0->tensorflow) (23.5.26)
Requirement already satisfied: ml-dtypes==0.2.0 in c:\users\lenovo\anaconda3\lib\site-packages (from tensorflow-intel==2.14.0->
tensorflow) (0.2.0)
Requirement already satisfied: grpcio<2.0,>=1.24.3 in c:\users\lenovo\anaconda3\lib\site-packages (from tensorflow-intel==2.14.
0->tensorflow) (1.59.0)
Requirement already satisfied: absl-py>=1.0.0 in c:\users\lenovo\anaconda3\lib\site-packages (from tensorflow-intel==2.14.0->te
nsorflow) (2.0.0)
Requirement already satisfied: six>=1.12.0 in c:\users\lenovo\anaconda3\lib\site-packages (from tensorflow-intel==2.14.0->tenso
rflow) (1.16.0)
Requirement already satisfied: libclang>=13.0.0 in c:\users\lenovo\anaconda3\lib\site-packages (from tensorflow-intel==2.14.0->
tensorflow) (16.0.6)
Requirement already satisfied: h5py>=2.9.0 in c:\users\lenovo\anaconda3\lib\site-packages (from tensorflow-intel==2.14.0->tenso
rflow) (3.7.0)
Requirement already satisfied: termcolor>=1.1.0 in c:\users\lenovo\anaconda3\lib\site-packages (from tensorflow-intel==2.14.0->
tensorflow) (2.3.0)
Requirement already satisfied: tensorflow-io-gcs-filesystem>=0.23.1 in c:\users\lenovo\anaconda3\lib\site-packages (from tensor
flow-intel==2.14.0->tensorflow) (0.31.0)
Requirement already satisfied: google-pasta>=0.1.1 in c:\users\lenovo\anaconda3\lib\site-packages (from tensorflow-intel==2.14.
0->tensorflow) (0.2.0)
Requirement already satisfied: protobuf!=4.21.0,!=4.21.1,!=4.21.2,!=4.21.3,!=4.21.4,!=4.21.5,<5.0.0dev,>=3.20.3 in c:\users\len
ovo\anaconda3\lib\site-packages (from tensorflow-intel==2.14.0->tensorflow) (4.24.4)
Requirement already satisfied: gast!=0.5.0,!=0.5.1,!=0.5.2,>=0.2.1 in c:\users\lenovo\anaconda3\lib\site-packages (from tensorf
low-intel==2.14.0->tensorflow) (0.5.4)
Requirement already satisfied: setuptools in c:\users\lenovo\anaconda3\lib\site-packages (from tensorflow-intel==2.14.0->tensor
flow) (65.6.3)
Requirement already satisfied: tensorflow-estimator<2.15,>=2.14.0 in c:\users\lenovo\anaconda3\lib\site-packages (from tensorfl
```

In [17]:
```
pip install keras
```

```
Requirement already satisfied: keras in c:\users\lenovo\anaconda3\lib\site-packages (2.14.0)
Note: you may need to restart the kernel to use updated packages.
```

In [18]:
```
import numpy as np
import random
import matplotlib.pyplot as plt
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Dense, Flatten
```

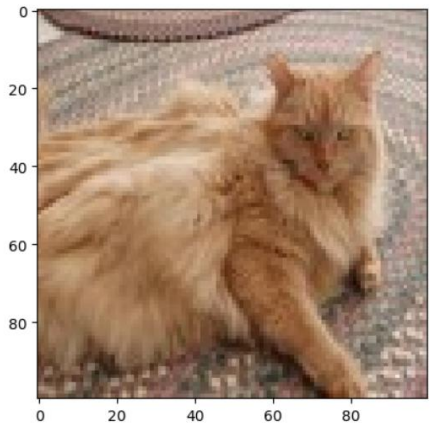
In [19]:
```
X_train = np.loadtxt('input.csv', delimiter = ',')
Y_train = np.loadtxt('labels.csv', delimiter = ',')

X_test = np.loadtxt('input_test.csv', delimiter = ',')
Y_test = np.loadtxt('labels_test.csv', delimiter = ',')
```

In [20]:
```
X_train = X_train.reshape(len(X_train), 100, 100, 3)
Y_train = Y_train.reshape(len(Y_train), 1)

X_test = X_test.reshape(len(X_test), 100, 100, 3)
Y_test = Y_test.reshape(len(Y_test), 1)

X_train = X_train/255.0
X_test = X_test/255.0
```

In [21]:
```
print("Shape of X_train: ", X_train.shape)
print("Shape of Y_train: ", Y_train.shape)
print("Shape of X_test: ", X_test.shape)
print("Shape of Y_test: ", Y_test.shape)
```

```
Shape of X_train:  (2000, 100, 100, 3)
Shape of Y_train:  (2000, 1)
Shape of X_test:  (400, 100, 100, 3)
Shape of Y_test:  (400, 1)
```

In [22]:
```
idx = random.randint(0, len(X_train))
plt.imshow(X_train[idx, :])
plt.show()
```



In [23]:
```
model = Sequential([
    Conv2D(32, (3,3), activation = 'relu', input_shape = (100, 100, 3)),
    MaxPooling2D((2,2)),

    Conv2D(32, (3,3), activation = 'relu'),
    MaxPooling2D((2,2)),
```

In [23]:
```
model = Sequential([
    Conv2D(32, (3,3), activation = 'relu', input_shape = (100, 100, 3)),
    MaxPooling2D((2,2)),

    Conv2D(32, (3,3), activation = 'relu'),
    MaxPooling2D((2,2)),

    Flatten(),
    Dense(64, activation = 'relu'),
    Dense(1, activation = 'sigmoid')
])
```

In [24]:
```
model = Sequential()

model.add(Conv2D(32, (3,3), activation = 'relu', input_shape = (100, 100, 3)))
model.add(MaxPooling2D((2,2)))

model.add(Conv2D(32, (3,3), activation = 'relu'))
model.add(MaxPooling2D((2,2)))

model.add(Flatten())
model.add(Dense(64, activation = 'relu'))
model.add(Dense(1, activation = 'sigmoid'))
```

In [25]:
```
model.compile(loss = 'binary_crossentropy', optimizer = 'adam', metrics = ['accuracy'])
```

In [26]:
```
model.fit(X_train, Y_train, epochs = 5, batch_size = 64)
```

```
Epoch 1/5
32/32 [==============================] - 6s 172ms/step - loss: 0.7219 - accuracy: 0.4970
Epoch 2/5
32/32 [==============================] - 6s 177ms/step - loss: 0.6843 - accuracy: 0.5675
Epoch 3/5
32/32 [==============================] - 7s 231ms/step - loss: 0.6501 - accuracy: 0.6205
Epoch 4/5
```

In [26]:
```
model.fit(X_train, Y_train, epochs = 5, batch_size = 64)
```

```
Epoch 1/5
32/32 [==============================] - 6s 172ms/step - loss: 0.7219 - accuracy: 0.4970
Epoch 2/5
32/32 [==============================] - 6s 177ms/step - loss: 0.6843 - accuracy: 0.5675
Epoch 3/5
32/32 [==============================] - 7s 231ms/step - loss: 0.6501 - accuracy: 0.6205
Epoch 4/5
32/32 [==============================] - 7s 214ms/step - loss: 0.6213 - accuracy: 0.6700
Epoch 5/5
32/32 [==============================] - 8s 266ms/step - loss: 0.5606 - accuracy: 0.7150
```

Out[26]:
```
<keras.src.callbacks.History at 0x224529e52d0>
```

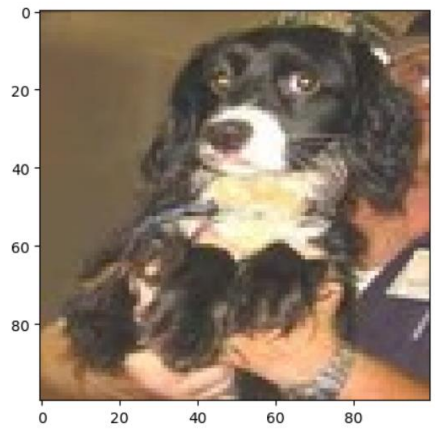
In [28]:
```
model.evaluate(X_test, Y_test)
```

```
13/13 [==============================] - 0s 23ms/step - loss: 0.6218 - accuracy: 0.6725
```

Out[28]:
```
[0.6217811107635498, 0.6725000143051147]
```

In [29]:
```
#Making Predictions
idx2 = random.randint(0, len(Y_test))
plt.imshow(X_test[idx2, :])
plt.show()

y_pred = model.predict(X_test[idx2, :].reshape(1, 100, 100, 3))
y_pred = y_pred > 0.5

if(y_pred == 0):
    pred = 'dog'
else:
    pred = 'cat'

print("Our model says it is a :", pred)
```


```
    pred = 'cat'

print("Our model says it is a :", pred)
```


```
1/1 [==============================] - 0s 74ms/step
Our model says it is a : dog
```

In [ ]: