In [1]:

```
#task3
import os
import numpy as np
import matplotlib.pyplot as plt
from PIL import Image
from sklearn.model_selection import train_test_split

def preprocess_image(img_path):
    # Load the image
    img = Image.open(img_path)
    # Resize the image to a fixed size, for example, 64x64 pixels
    img = img.resize((64, 64))
    # Convert the image to a numpy array
    img_array = np.array(img)
    # Normalize the image array (values between 0 and 1)
    img_array = img_array / 255.0
    return img_array

def load_data(image_dir):
    data = []
    labels = []
    valid_extensions = ('.jpg', '.jpeg', '.png', '.bmp', '.gif')  # Add more extensior

    for img in os.listdir(image_dir):
        img_path = os.path.join(image_dir, img)
        if os.path.isfile(img_path) and img.lower().endswith(valid_extensions):
            if 'cat' in img.lower():
                labels.append(0)  # Label for cats
            elif 'dog' in img.lower():
                labels.append(1)  # Label for dogs
            else:
                print(f"Skipping unknown label for image: {img}")
                continue  # Skip images without 'cat' or 'dog' in the filename

            data.append(preprocess_image(img_path))

    return np.array(data), np.array(labels)

def visualize_images(images, labels, class_names):
    # Select a random subset of images
    num_samples = min(9, len(images))
    idx = np.random.choice(len(images), num_samples, replace=False)

    # Plot the images
    plt.figure(figsize=(10, 10))
    for i, index in enumerate(idx):
        plt.subplot(3, 3, i + 1)
        plt.imshow(images[index])
        plt.title(class_names[labels[index]])
        plt.axis('off')
    plt.show()

# Specify the directory containing all images
image_dir = r"C:\Users\k.Navathi\OneDrive\Desktop\images"

# Load data
X, y = load_data(image_dir)
```

```
# Split the data (simple split due to small dataset size)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.5, random_state=

# Define class names
class_names = ['Cat', 'Dog']

# Visualize a random subset of images from the dataset
visualize_images(X_train, y_train, class_names)
```

In [ ]: