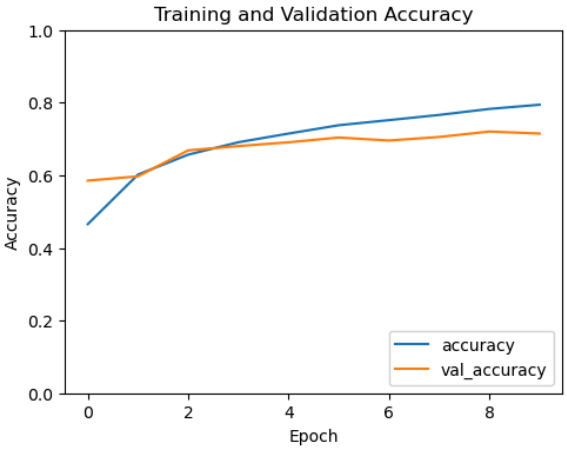
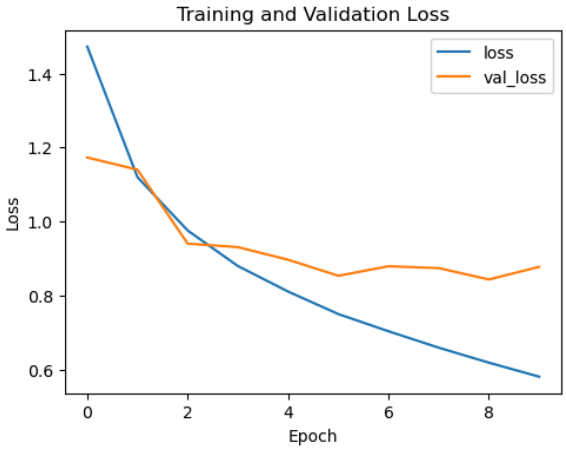
In [1]:
```
import tensorflow as tf
from tensorflow.keras import layers, models
import matplotlib.pyplot as plt
import numpy as np

# --- 1. Load and Preprocess the Dataset ---
# Load the CIFAR-10 dataset
(train_images, train_labels), (test_images, test_labels) = tf.keras.datasets.cif

# Normalize pixel values to be between 0 and 1
train_images, test_images = train_images / 255.0, test_images / 255.0

# Define class names for visualization
class_names = ['airplane', 'automobile', 'bird', 'cat', 'deer',
               'dog', 'frog', 'horse', 'ship', 'truck']

# --- 2. Build the Model ---
model = models.Sequential([
    layers.Conv2D(32, (3, 3), activation='relu', input_shape=(32, 32, 3)),
    layers.MaxPooling2D((2, 2)),
    layers.Conv2D(64, (3, 3), activation='relu'),
    layers.MaxPooling2D((2, 2)),
    layers.Conv2D(64, (3, 3), activation='relu'),
    layers.Flatten(),
    layers.Dense(64, activation='relu'),
    layers.Dense(10, activation='softmax')
])

# --- 3. Compile the Model ---
model.compile(optimizer='adam',
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])

# --- 4. Train the Model ---
history = model.fit(train_images, train_labels, epochs=10,
                    validation_data=(test_images, test_labels))

# --- 5. Evaluate the Model ---
test_loss, test_acc = model.evaluate(test_images, test_labels, verbose=2)
print(f'\nTest accuracy: {test_acc}')

# --- 6. Visualize Results ---

# Plot training history (accuracy and loss)
plt.figure(figsize=(12, 4))
plt.subplot(1, 2, 1)
plt.plot(history.history['accuracy'], label='accuracy')
plt.plot(history.history['val_accuracy'], label = 'val_accuracy')
plt.xlabel('Epoch')
plt.ylabel('Accuracy')
plt.ylim([0, 1])
plt.legend(loc='lower right')
plt.title('Training and Validation Accuracy')

plt.subplot(1, 2, 2)
plt.plot(history.history['loss'], label='loss')
plt.plot(history.history['val_loss'], label = 'val_loss')
plt.xlabel('Epoch')
```

```
plt.ylabel('Loss')
plt.legend(loc='upper right')
plt.title('Training and Validation Loss')
plt.show()

# Visualize some predictions
predictions = model.predict(test_images)

plt.figure(figsize=(10, 10))
for i in range(25):
    plt.subplot(5, 5, i + 1)
    plt.xticks([])
    plt.yticks([])
    plt.grid(False)
    plt.imshow(test_images[i])
    predicted_label = np.argmax(predictions[i])
    true_label = test_labels[i][0]
    color = 'blue' if predicted_label == true_label else 'red'
    plt.xlabel(f'{class_names[predicted_label]} ({class_names[true_label]})', co
plt.show()
```

```
C:\ProgramData\anaconda3\Lib\site-packages\keras\src\layers\convolutional\base_co
nv.py:107: UserWarning: Do not pass an `input_shape`/`input_dim` argument to a la
yer. When using Sequential models, prefer using an `Input(shape)` object as the f
irst layer in the model instead.
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
Epoch 1/10
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 38s 22ms/step - accuracy: 0.3706 - loss: 1.7069 -
val_accuracy: 0.5853 - val_loss: 1.1729
Epoch 2/10
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 43s 24ms/step - accuracy: 0.5850 - loss: 1.1650 -
val_accuracy: 0.5972 - val_loss: 1.1401
Epoch 3/10
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 39s 25ms/step - accuracy: 0.6533 - loss: 0.9862 -
val_accuracy: 0.6687 - val_loss: 0.9405
Epoch 4/10
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 42s 26ms/step - accuracy: 0.6885 - loss: 0.8850 -
val_accuracy: 0.6805 - val_loss: 0.9311
Epoch 5/10
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 34s 22ms/step - accuracy: 0.7112 - loss: 0.8140 -
val_accuracy: 0.6909 - val_loss: 0.8971
Epoch 6/10
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 44s 23ms/step - accuracy: 0.7422 - loss: 0.7350 -
val_accuracy: 0.7040 - val_loss: 0.8538
Epoch 7/10
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 38s 21ms/step - accuracy: 0.7564 - loss: 0.6872 -
val_accuracy: 0.6958 - val_loss: 0.8794
Epoch 8/10
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 33s 21ms/step - accuracy: 0.7727 - loss: 0.6466 -
val_accuracy: 0.7056 - val_loss: 0.8744
Epoch 9/10
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 33s 21ms/step - accuracy: 0.7883 - loss: 0.6013 -
val_accuracy: 0.7205 - val_loss: 0.8437
Epoch 10/10
1563/1563 ━━━━━━━━━━━━━━━━━━━━ 45s 29ms/step - accuracy: 0.7981 - loss: 0.5713 -
val_accuracy: 0.7150 - val_loss: 0.8778
313/313 - 2s - 7ms/step - accuracy: 0.7150 - loss: 0.8778

Test accuracy: 0.7149999737739563
```






```
313/313 ━━━━━━━━━━━━━━━━━━━━ 3s 8ms/step
```