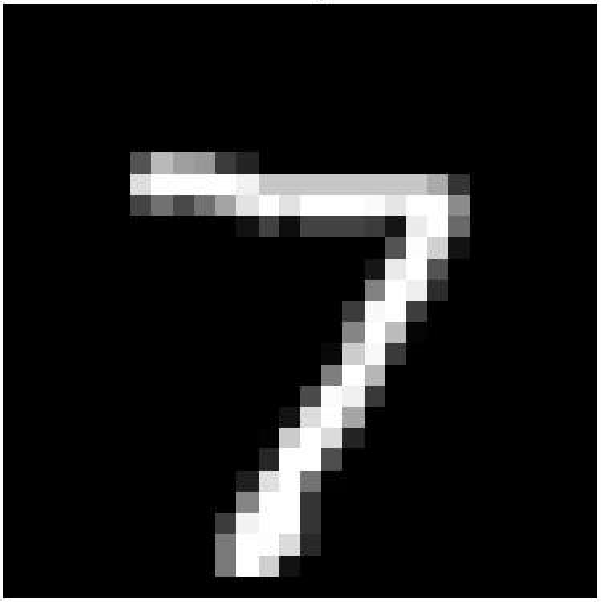
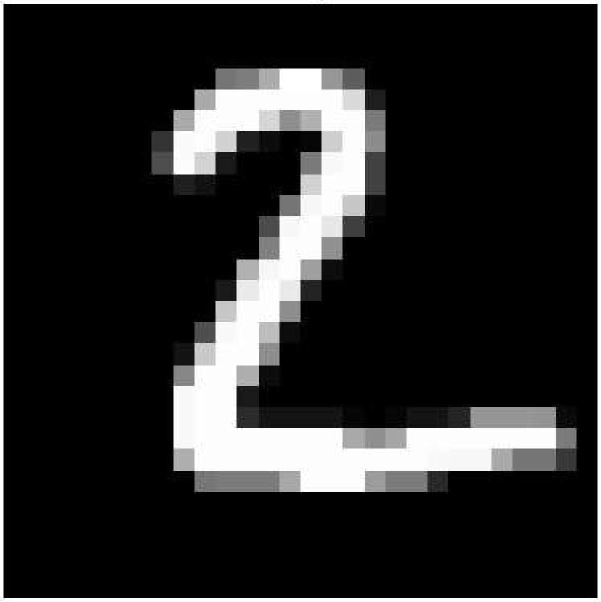
```
import tensorflow as tf
from tensorflow.keras.datasets import mnist
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Flatten
from tensorflow.keras.utils import to_categorical
import matplotlib.pyplot as plt

# 1. Load and preprocess the data
(x_train, y_train), (x_test, y_test) = mnist.load_data()

# Normalize the pixel values (0-255 to 0-1)
x_train = x_train / 255.0
x_test = x_test / 255.0

# One-hot encode the labels
y_train_cat = to_categorical(y_train, 10)
y_test_cat = to_categorical(y_test, 10)

# 2. Build the model
model = Sequential([
    Flatten(input_shape=(28, 28)),
    Dense(128, activation='relu'),
    Dense(64, activation='relu'),
    Dense(10, activation='softmax')
])

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

# 4. Train the model
model.fit(x_train, y_train_cat, epochs=5, validation_split=0.1)

# 5. Evaluate the model
test_loss, test_accuracy = model.evaluate(x_test, y_test_cat)
print(f"Test accuracy: {test_accuracy:.4f}")

# 6. Predict and display sample results
predictions = model.predict(x_test)

# Display 5 sample predictions
for i in range(5):
    plt.imshow(x_test[i], cmap='gray')
    plt.title(f"Predicted: {predictions[i].argmax()} | Actual: {y_test[i]}")
    plt.axis('off')
    plt.show()
```

```
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/mnist.npz
11490434/11490434 ━━━━━━━━━━━━━━━━━━━━ 0s 0us/step
/usr/local/lib/python3.11/dist-packages/keras/src/layers/reshaping/flatten.py:37: UserWarning: Do not pass an `i
  super().__init__(**kwargs)
Epoch 1/5
1688/1688 ━━━━━━━━━━━━━━━━━━━━ 13s 7ms/step - accuracy: 0.8688 - loss: 0.4549 - val_accuracy: 0.9663 - val_loss:
Epoch 2/5
1688/1688 ━━━━━━━━━━━━━━━━━━━━ 17s 5ms/step - accuracy: 0.9655 - loss: 0.1131 - val_accuracy: 0.9725 - val_loss:
Epoch 3/5
1688/1688 ━━━━━━━━━━━━━━━━━━━━ 10s 5ms/step - accuracy: 0.9769 - loss: 0.0736 - val_accuracy: 0.9735 - val_loss:
Epoch 4/5
1688/1688 ━━━━━━━━━━━━━━━━━━━━ 10s 4ms/step - accuracy: 0.9848 - loss: 0.0541 - val_accuracy: 0.9753 - val_loss:
Epoch 5/5
1688/1688 ━━━━━━━━━━━━━━━━━━━━ 7s 4ms/step - accuracy: 0.9870 - loss: 0.0410 - val_accuracy: 0.9772 - val_loss:
313/313 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - accuracy: 0.9726 - loss: 0.0909
Test accuracy: 0.9775
313/313 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step
```

Predicted: 7 | Actual: 7



Predicted: 2 | Actual: 2



Predicted: 1 | Actual: 1