In [1]:
```
import numpy as np
import pandas as pd
import tensorflow as tf
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import confusion_matrix, classification_report
from tensorflow.keras import layers, models
import matplotlib.pyplot as plt
```

In [4]:
```
# Load the ECG dataset
ecg_dataset = pd.read_csv("ecg.csv",header=None)
```

In [7]:
```
data = ecg_dataset.iloc[:,:-1].values
labels = ecg_dataset.iloc[:,-1].values
```

In [11]:
```
scaler = StandardScaler()
X = scaler.fit_transform(data)
```

In [12]:
```
train_data, test_data, train_labels, test_labels = train_test_split(X, labels,
```

In [16]:
```
input_dim=train_data.shape[1]
input_dim
```

Out[16]:
```
140
```

In [14]:
```
y = X  # Autoencoder input and output are the same
```

In [18]:
```
encoder = models.Sequential([
    layers.Input(shape=(input_dim,)),
    layers.Dense(32, activation='relu'),
    layers.Dense(16, activation='relu'),
    layers.Dense(8, activation='relu')
])
decoder = models.Sequential([
    layers.Input(shape=(8,)),
    layers.Dense(16, activation='relu'),
    layers.Dense(32, activation='relu'),
    layers.Dense(input_dim, activation='sigmoid')  # Use linear activation for
])
```

In [28]:
```
autoencoder = models.Sequential([
    encoder,
    decoder
])
autoencoder.compile(optimizer='adam', loss='mean_squared_error')
autoencoder.fit(train_data, train_data, epochs=500, batch_size=32)
```

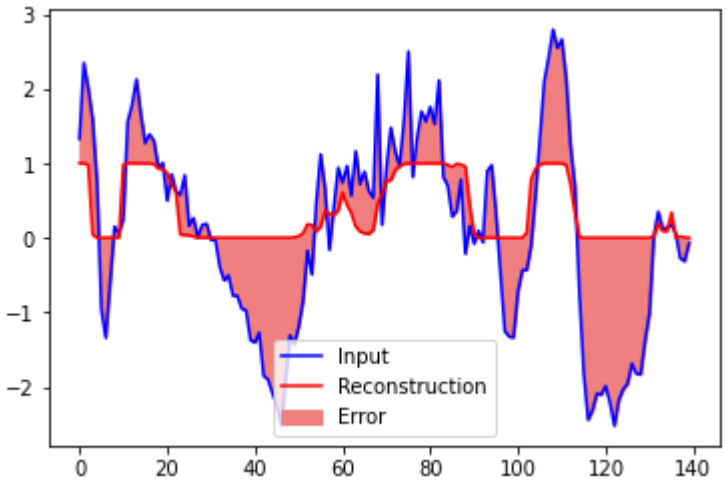
```
Epoch 492/500
125/125 [==============================] - 0s 4ms/step - loss: 0.6464
Epoch 493/500
125/125 [==============================] - 0s 3ms/step - loss: 0.6464
Epoch 494/500
125/125 [==============================] - 0s 3ms/step - loss: 0.6465
Epoch 495/500
125/125 [==============================] - 0s 3ms/step - loss: 0.6464
Epoch 496/500
125/125 [==============================] - 0s 3ms/step - loss: 0.6463
Epoch 497/500
125/125 [==============================] - 0s 3ms/step - loss: 0.6463
Epoch 498/500
125/125 [==============================] - 0s 3ms/step - loss: 0.6463
Epoch 499/500
125/125 [==============================] - 0s 4ms/step - loss: 0.6464
Epoch 500/500
125/125 [==============================] - 0s 4ms/step - loss: 0.6465
```

Out[28]: <keras.callbacks.History at 0x2454122be20>

In [33]:
```
def dplot(data,n):
    enc_img=encoder(data)
    dec_img=decoder(enc_img)


    plt.plot(data[n],"b")
    plt.plot(dec_img[n],"r")
    plt.fill_between(np.arange(140),data[n], dec_img[n], color = "lightcoral")
    plt.legend(labels=['Input', 'Reconstruction', 'Error'])
    plt.show()
```

In [34]:
```
dplot(test_data,2)
```



In [ ]: