In [2]:
```
from tensorflow.keras.datasets import imdb
```

In [3]:
```
from tensorflow.keras.datasets import imdb
from tensorflow.keras.preprocessing.sequence import pad_sequences

(x_train, y_train), (x_test, y_test) = imdb.load_data(num_words=10000)

max_len = 200
x_train = pad_sequences(x_train, maxlen=max_len)
x_test = pad_sequences(x_test, maxlen=max_len)
```

In [4]:
```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Embedding, LSTM, Dense, Dropout

model = Sequential([
    Embedding(input_dim=10000, output_dim=128, input_length=max_len),
    LSTM(64, return_sequences=False),
    Dropout(0.5),
    Dense(1, activation='sigmoid')
])

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

history = model.fit(x_train, y_train, epochs=4, batch_size=128, validation_split=0.2)
```

```
C:\Users\Admin\anaconda3\Lib\site-packages\keras\src\layers\core\embedding.py:90: UserWarning: Argument `input_length` is deprecated. Just remove it.
  warnings.warn(
Epoch 1/4
157/157 ━━━━━━━━━━━━━━━━━━━━ 73s 422ms/step - accuracy: 0.6578 - loss: 0.6100 - val_accuracy: 0.8538 - val_loss: 0.3589
Epoch 2/4
157/157 ━━━━━━━━━━━━━━━━━━━━ 65s 412ms/step - accuracy: 0.8889 - loss: 0.2824 - val_accuracy: 0.8518 - val_loss: 0.3420
Epoch 3/4
157/157 ━━━━━━━━━━━━━━━━━━━━ 65s 411ms/step - accuracy: 0.9227 - loss: 0.2138 - val_accuracy: 0.8722 - val_loss: 0.3250
Epoch 4/4
157/157 ━━━━━━━━━━━━━━━━━━━━ 65s 413ms/step - accuracy: 0.9520 - loss: 0.1390 - val_accuracy: 0.8522 - val_loss: 0.3547
```

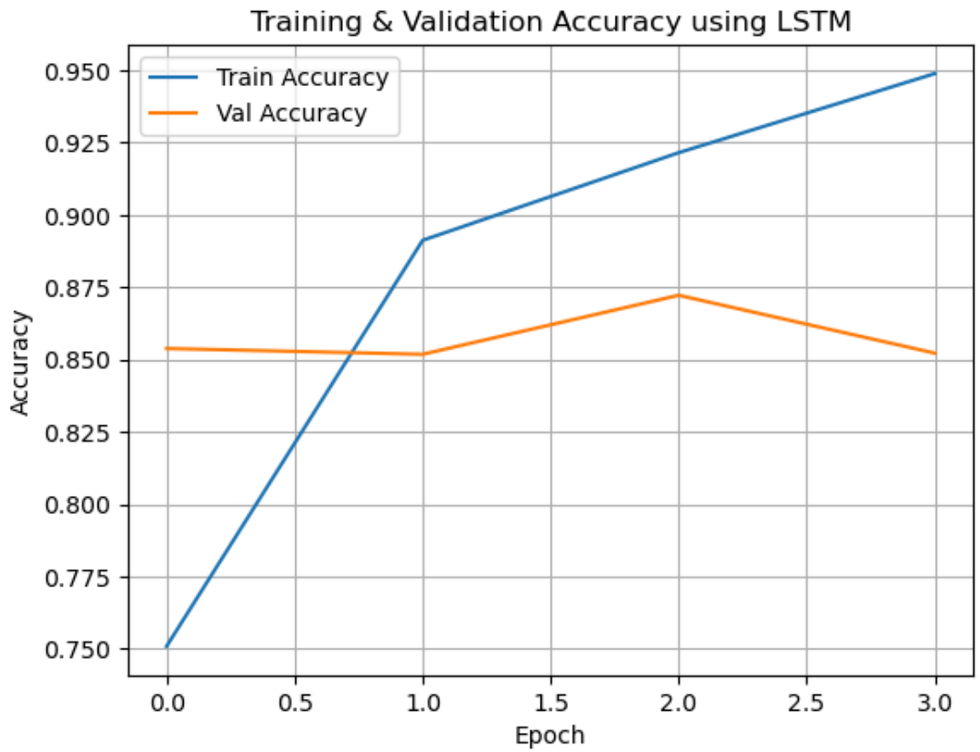
In [5]:
```
loss, accuracy = model.evaluate(x_test, y_test)
print(f"Test Accuracy: {accuracy:.4f}")
```

```
782/782 ━━━━━━━━━━━━━━━━━━━━ 48s 62ms/step - accuracy: 0.8519 - loss: 0.3630
Test Accuracy: 0.8520
```

In [6]:
```
import matplotlib.pyplot as plt

plt.plot(history.history['accuracy'], label='Train Accuracy')
plt.plot(history.history['val_accuracy'], label='Val Accuracy')
plt.title('Training & Validation Accuracy using LSTM')
```

```
plt.xlabel('Epoch')
plt.ylabel('Accuracy')
plt.legend()
plt.grid(True)
plt.show()
```



In [ ]: