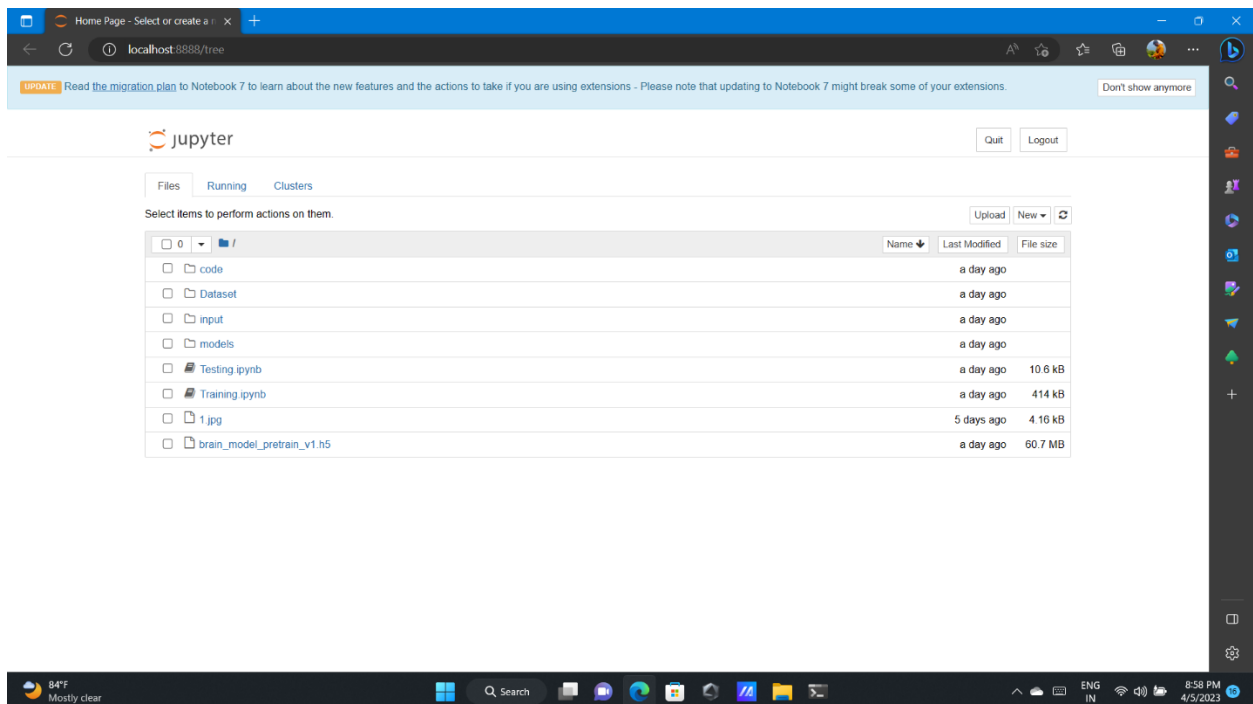
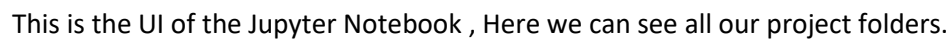
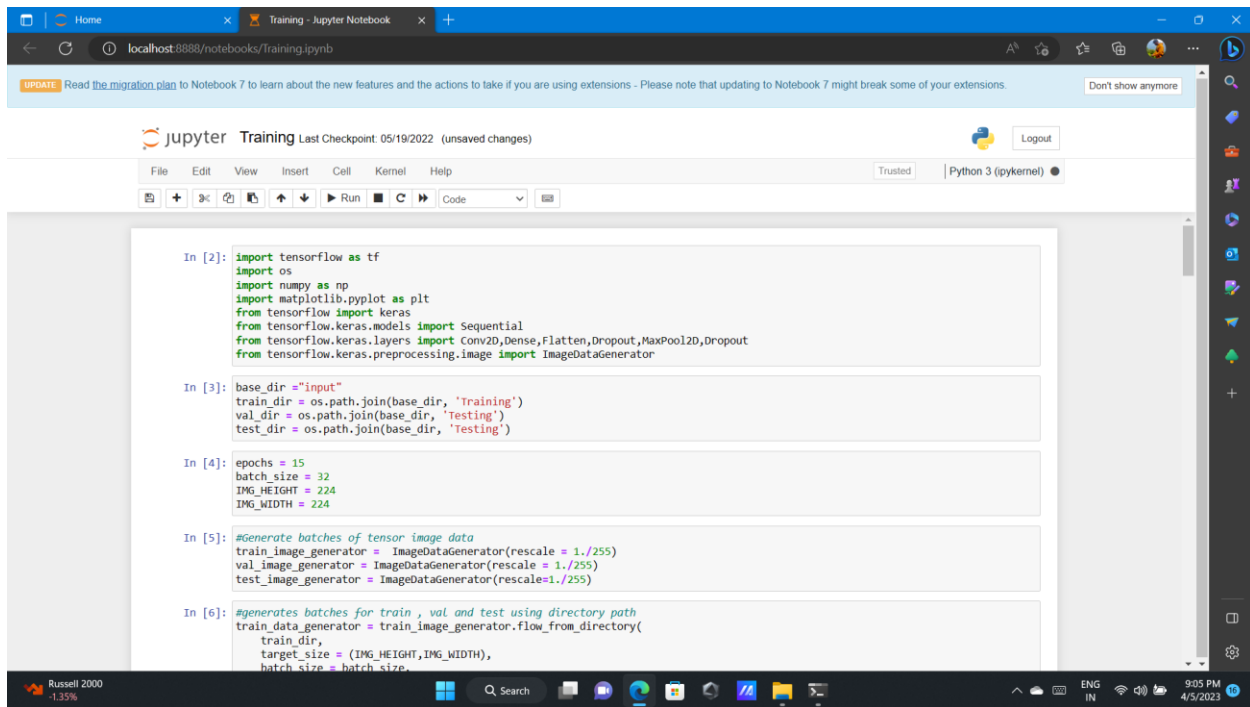


STEP 3 : Call the Jupyter Notebook from here.



#### STEP 4 : Run all the cells in the Training folder (module)



The screenshot shows a Jupyter Notebook titled 'Training' with the following code cells:

```
In [2]: import tensorflow as tf
import os
import numpy as np
import matplotlib.pyplot as plt
from tensorflow import keras
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D,Dense,Flatten,Dropout,MaxPool2D,Dropout
from tensorflow.keras.preprocessing.image import ImageDataGenerator

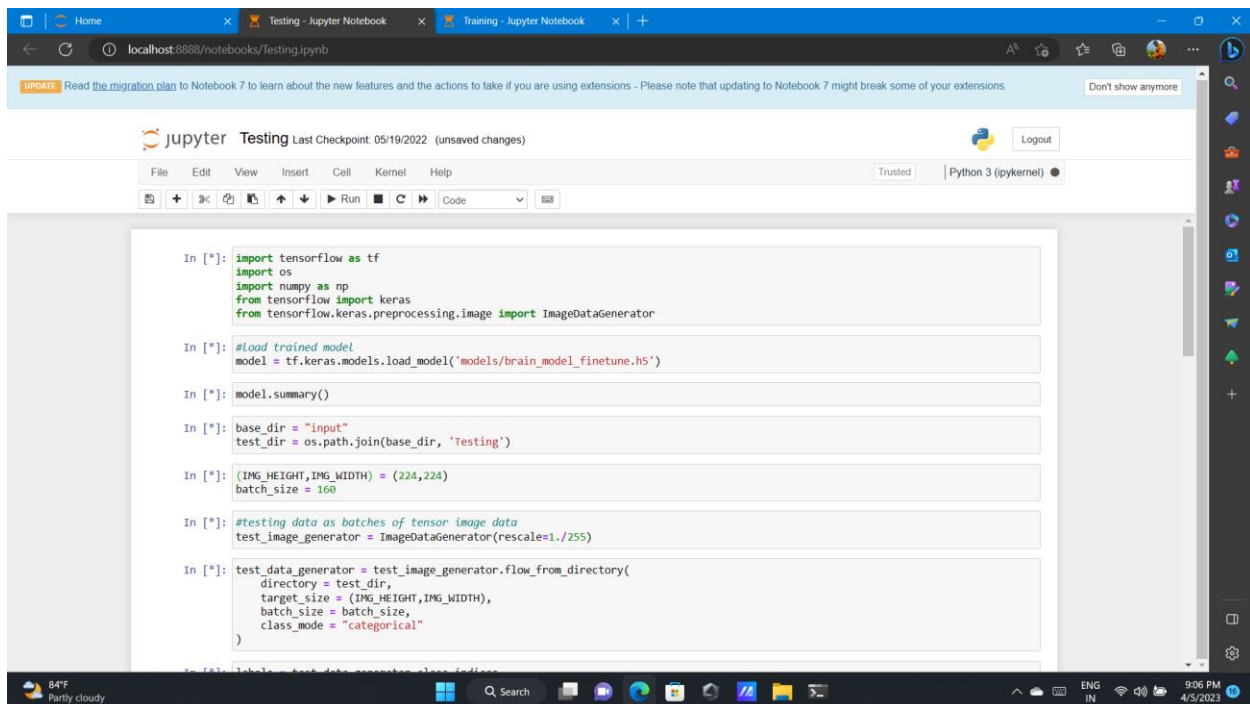
In [3]: base_dir = "input"
train_dir = os.path.join(base_dir, 'training')
val_dir = os.path.join(base_dir, 'Testing')
test_dir = os.path.join(base_dir, 'Testing')

In [4]: epochs = 15
batch_size = 32
IMG_HEIGHT = 224
IMG_WIDTH = 224

In [5]: #Generate batches of tensor image data
train_image_generator = ImageDataGenerator(rescale = 1./255)
val_image_generator = ImageDataGenerator(rescale = 1./255)
test_image_generator = ImageDataGenerator(rescale=1./255)

In [6]: #generates batches for train , val and test using directory path
train_data_generator = train_image_generator.flow_from_directory(
    train_dir,
    target_size = (IMG_HEIGHT,IMG_WIDTH),
    batch_size = batch_size,
```

#### STEP 5 : Run all the cells in the Testing folder (module)



The screenshot shows a Jupyter Notebook titled 'Testing' with the following code cells:

```
In [*]: import tensorflow as tf
import os
import numpy as np
from tensorflow import keras
from tensorflow.keras.preprocessing.image import ImageDataGenerator

In [*]: #Load trained model
model = tf.keras.models.load_model('models/brain_model_finetune.h5')

In [*]: model.summary()

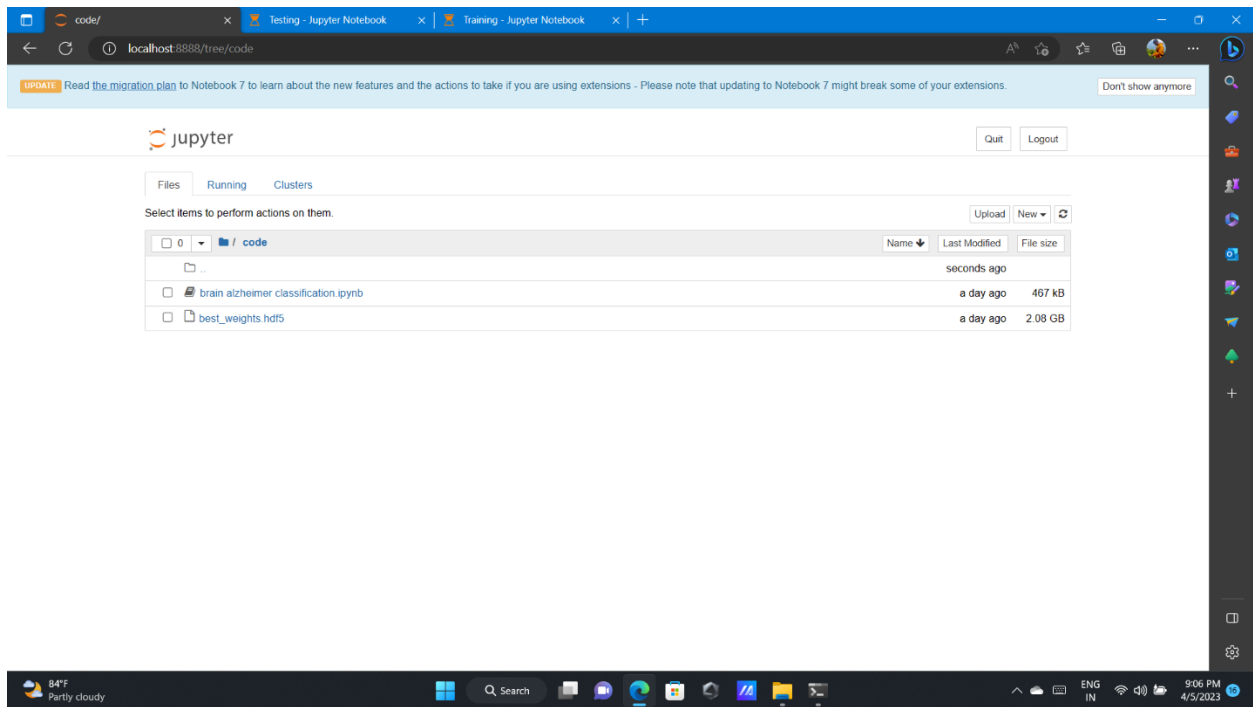
In [*]: base_dir = "input"
test_dir = os.path.join(base_dir, 'Testing')

In [*]: (IMG_HEIGHT,IMG_WIDTH) = (224,224)
batch_size = 160

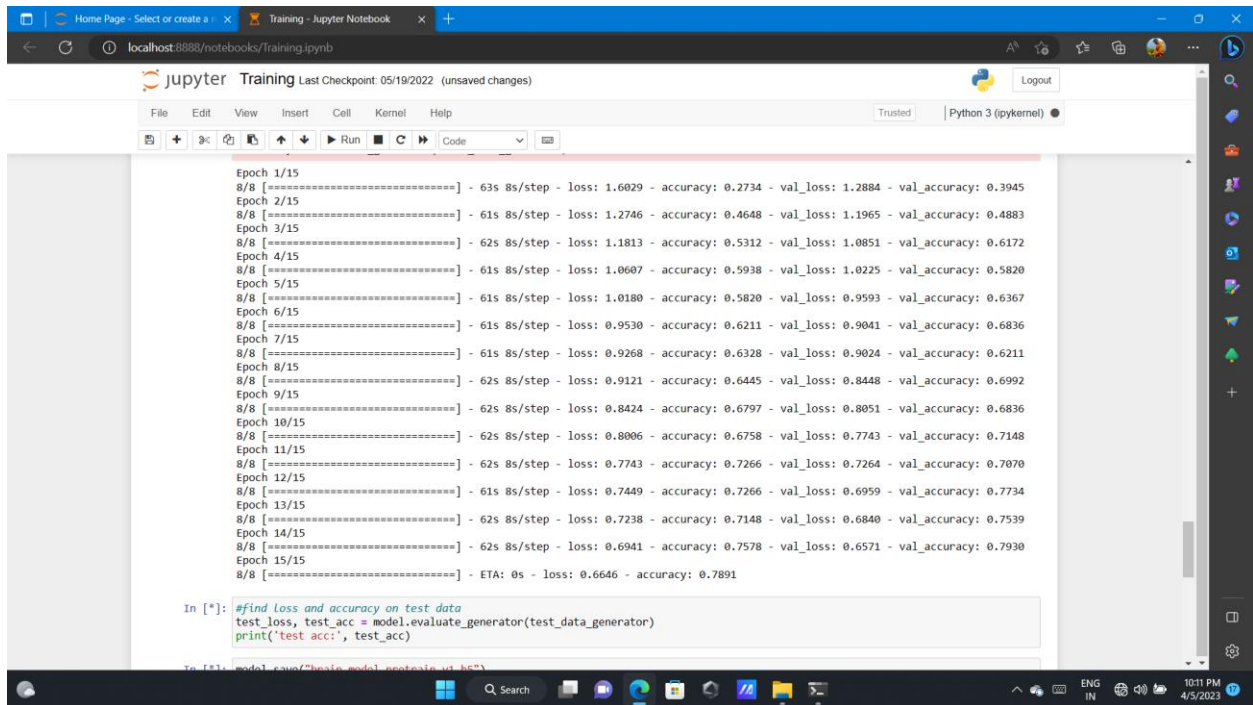
In [*]: #testing data as batches of tensor image data
test_image_generator = ImageDataGenerator(rescale=1./255)

In [*]: test_data_generator = test_image_generator.flow_from_directory(
    directory = test_dir,
    target_size = (IMG_HEIGHT,IMG_WIDTH),
    batch_size = batch_size,
    class_mode = "categorical"
```

STEP 6 : Now click on the code folder and select the main source file.



This is how all the Epoches are completed after a successful run.



```
jupyter brain alzheimer classification Last Checkpoint: Yesterday at 5:34 PM (unsaved changes)
Python 3 (pykernel)

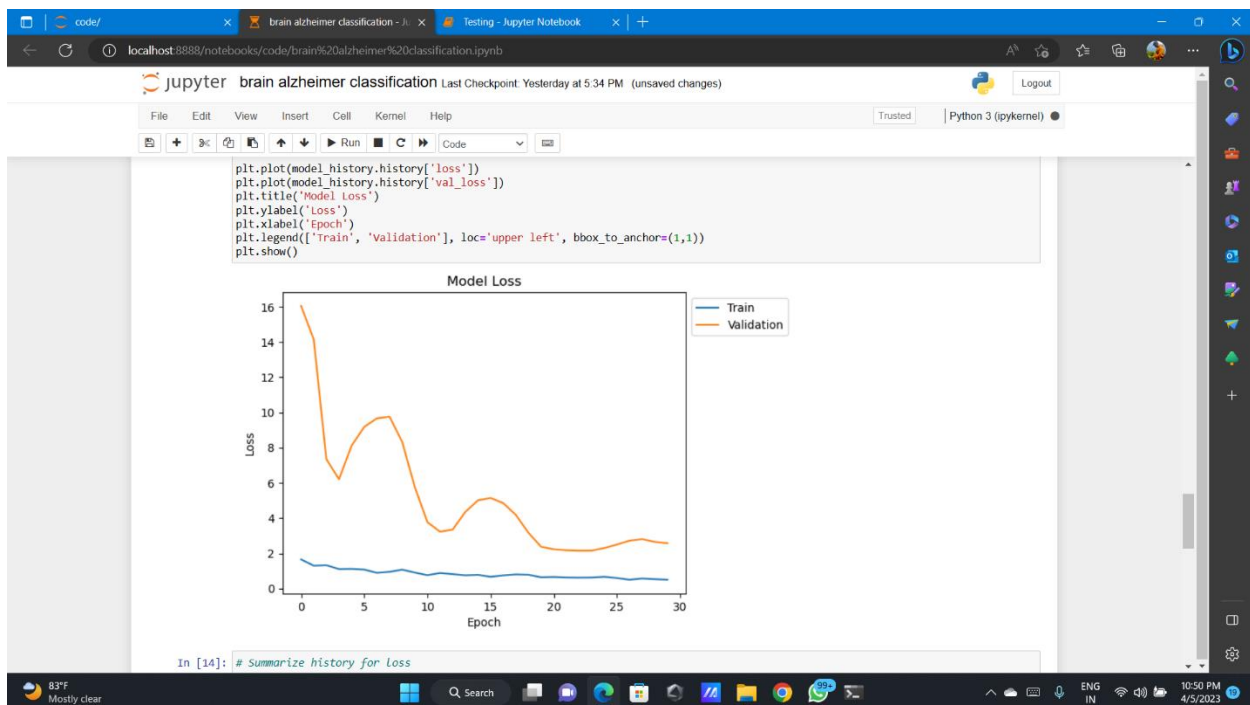
File Edit View Insert Cell Kernel Help Trusted

In [*]:
# Summarize history for Loss
plt.plot(model_history.history['loss'])
plt.plot(model_history.history['val_loss'])
plt.title('Model Loss')
plt.ylabel('Loss')
plt.xlabel('Epoch')
plt.legend(['Train', 'Validation'], loc='upper left', bbox_to_anchor=(1,1))
plt.show()

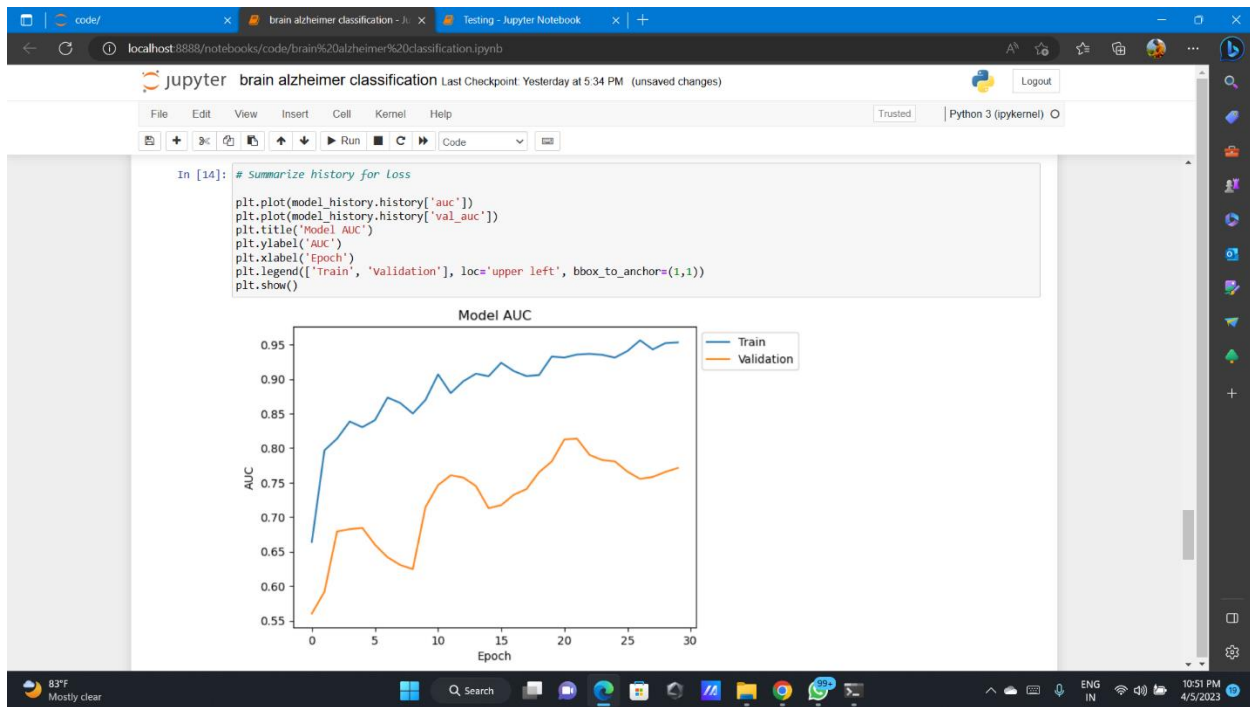
In [*]: # Summarize history for Loss
plt.plot(model_history.history['auc'])
plt.plot(model_history.history['val_auc'])
plt.title('Model AUC')
plt.ylabel('AUC')
plt.xlabel('Epoch')
plt.legend(['Train', 'Validation'], loc='upper left', bbox_to_anchor=(1,1))
plt.show()
```

2/2 [=====] - 32s 22s/step - loss: 0.5008 - auc: 0.9501 - val\_loss: 2.7187 - val\_auc: 0.7553  
Epoch 28/30  
2/2 [=====] - ETA: 0s - loss: 0.5712 - auc: 0.9428  
Epoch 28: val\_auc did not improve from 0.81367  
2/2 [=====] - 29s 19s/step - loss: 0.5712 - auc: 0.9428 - val\_loss: 2.8053 - val\_auc: 0.7581  
Epoch 29/30  
2/2 [=====] - ETA: 0s - loss: 0.5327 - auc: 0.9520  
Epoch 29: val\_auc did not improve from 0.81367  
2/2 [=====] - 31s 13s/step - loss: 0.5327 - auc: 0.9520 - val\_loss: 2.6476 - val\_auc: 0.7654  
Epoch 30/30  
2/2 [=====] - ETA: 0s - loss: 0.5018 - auc: 0.9530

The Train and Validation graph is successfully plotted and shown below (Model Loss)



The Train and Validation graph is successfully plotted and shown below (Model Accuracy)



The final successful input and output are shown below

