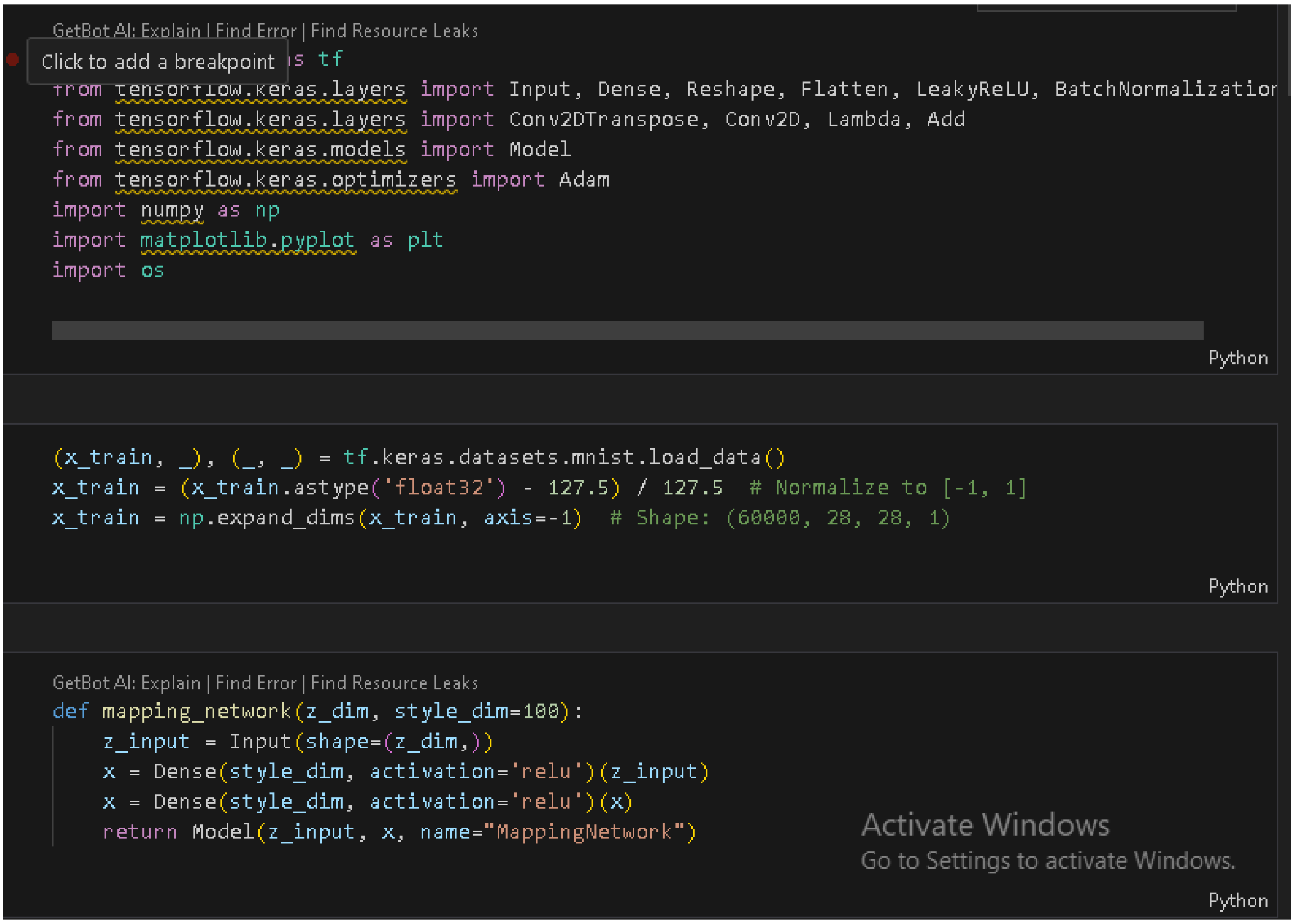
	Marwadi University **Faculty of Engineering and Technology** **Department of Information and Communication Technology**	
Subject: Gen AI	**Write a code for STYLE GAN. USE DATASET MNIST FROM KERAS**	
Experiment	**Date:**	**Enrolment No:92200133020**

CODE:



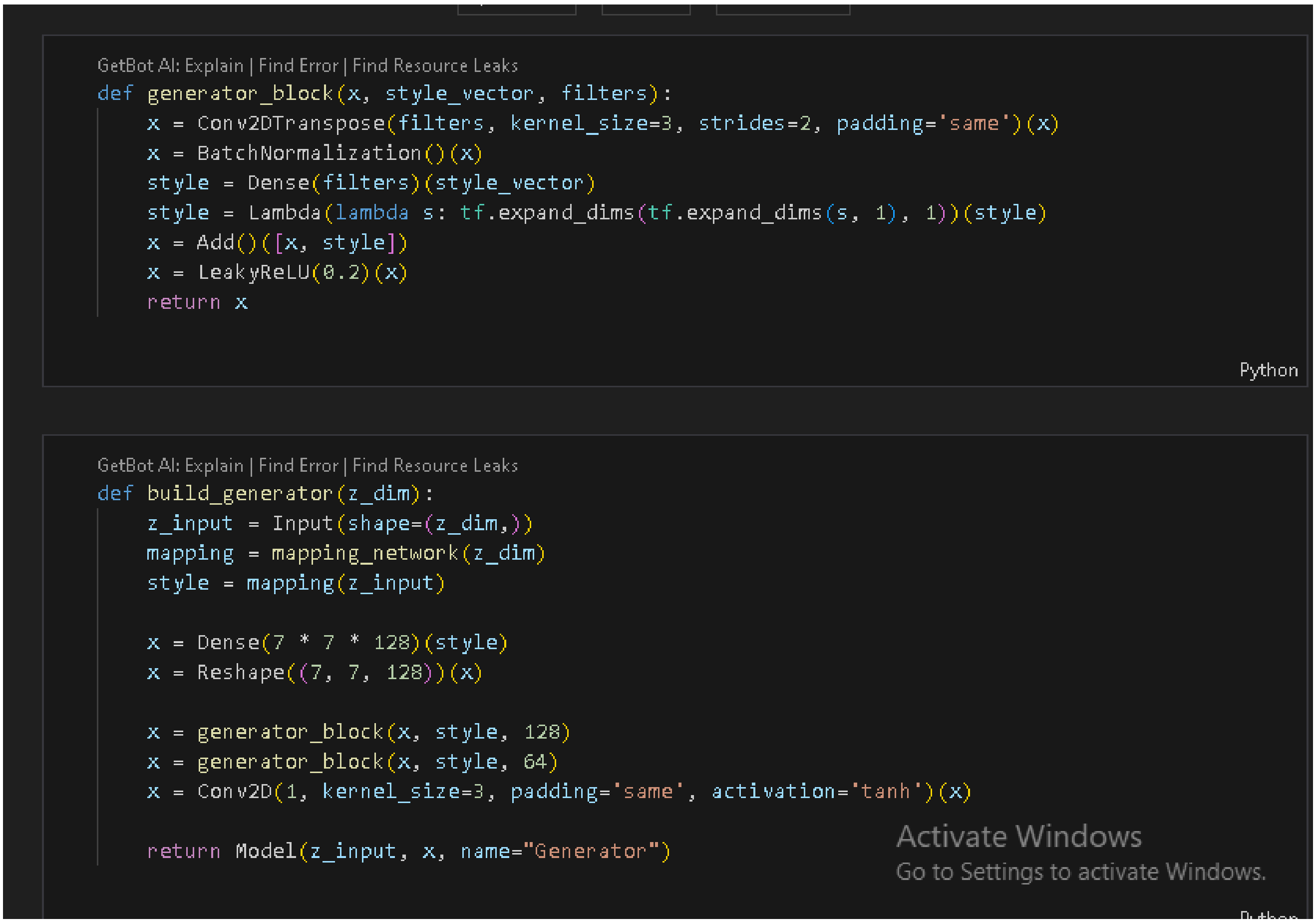
GetBot AI: Explain | Find Error | Find Resource Leaks
Click to add a breakpoint

```
                 s tf
from tensorflow.keras.layers import Input, Dense, Reshape, Flatten, LeakyReLU, BatchNormalizatior
from tensorflow.keras.layers import Conv2DTranspose, Conv2D, Lambda, Add
from tensorflow.keras.models import Model
from tensorflow.keras.optimizers import Adam
import numpy as np
import matplotlib.pyplot as plt
import os
```

```
(x_train, _), (_, _) = tf.keras.datasets.mnist.load_data()
x_train = (x_train.astype('float32') - 127.5) / 127.5  # Normalize to [-1, 1]
x_train = np.expand_dims(x_train, axis=-1)  # Shape: (60000, 28, 28, 1)
```

```
def mapping_network(z_dim, style_dim=100):
    z_input = Input(shape=(z_dim,))
    x = Dense(style_dim, activation='relu')(z_input)
    x = Dense(style_dim, activation='relu')(x)
    return Model(z_input, x, name="MappingNetwork")
```

Activate Windows
Go to Settings to activate Windows.



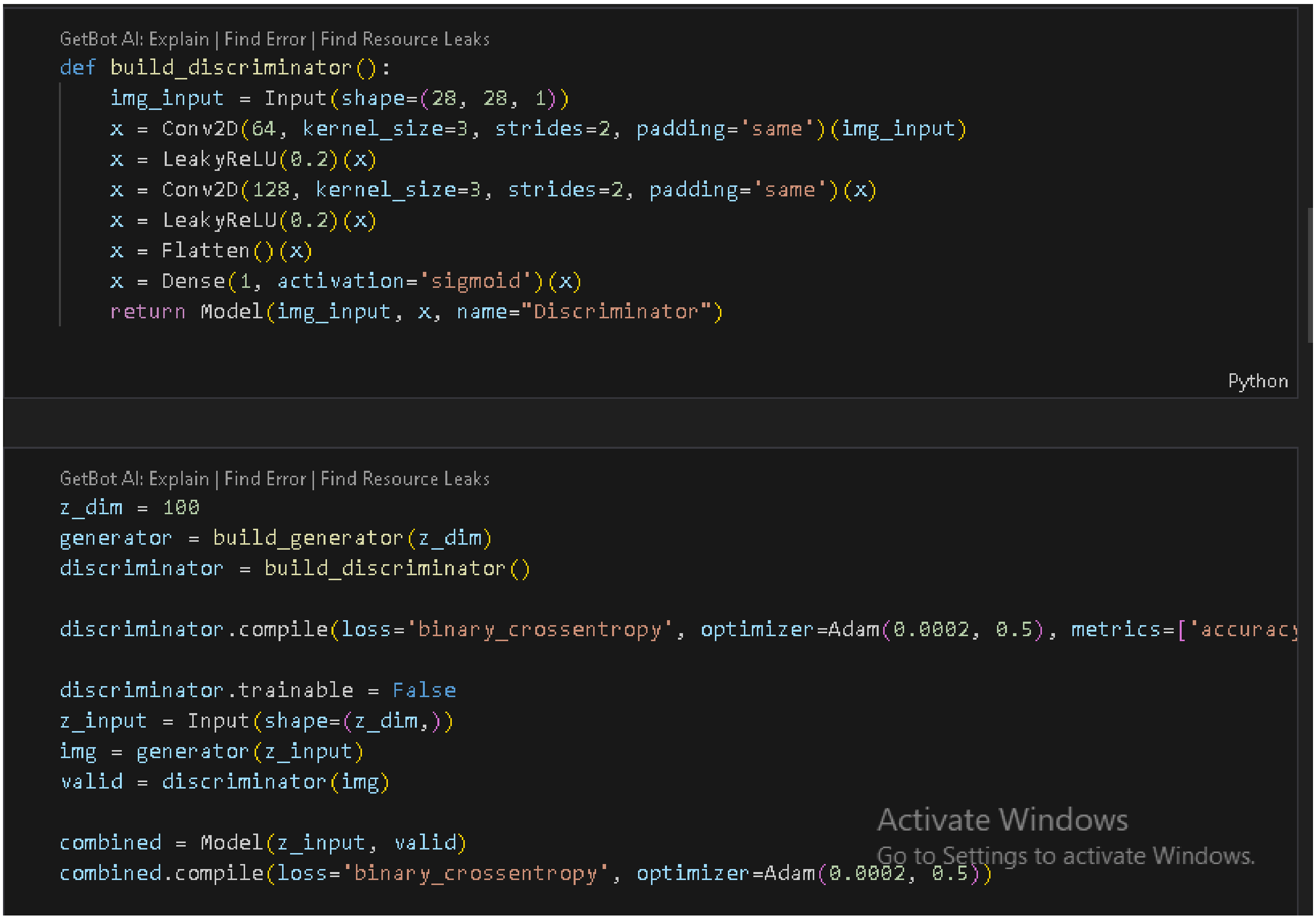
GetBot AI: Explain | Find Error | Find Resource Leaks

```python
def generator_block(x, style_vector, filters):
    x = Conv2DTranspose(filters, kernel_size=3, strides=2, padding='same')(x)
    x = BatchNormalization()(x)
    style = Dense(filters)(style_vector)
    style = Lambda(lambda s: tf.expand_dims(tf.expand_dims(s, 1), 1))(style)
    x = Add()([x, style])
    x = LeakyReLU(0.2)(x)
    return x
```

Python
GetBot AI: Explain | Find Error | Find Resource Leaks

```python
def build_generator(z_dim):
    z_input = Input(shape=(z_dim,))
    mapping = mapping_network(z_dim)
    style = mapping(z_input)

    x = Dense(7 * 7 * 128)(style)
    x = Reshape((7, 7, 128))(x)

    x = generator_block(x, style, 128)
    x = generator_block(x, style, 64)
    x = Conv2D(1, kernel_size=3, padding='same', activation='tanh')(x)

    return Model(z_input, x, name="Generator")
```

Activate Windows
Go to Settings to activate Windows.



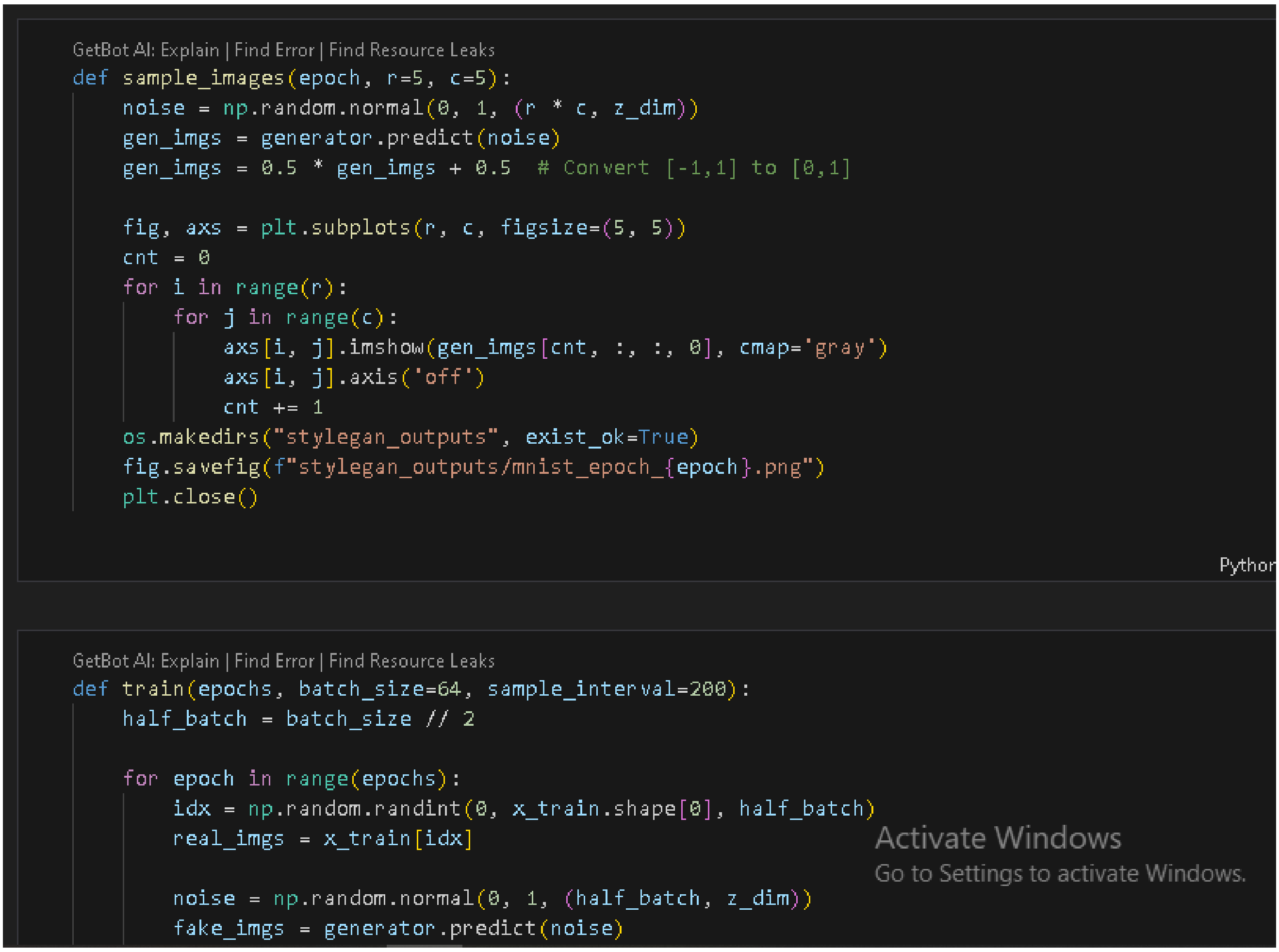
GetBot AI: Explain | Find Error | Find Resource Leaks

```python
def build_discriminator():
    img_input = Input(shape=(28, 28, 1))
    x = Conv2D(64, kernel_size=3, strides=2, padding='same')(img_input)
    x = LeakyReLU(0.2)(x)
    x = Conv2D(128, kernel_size=3, strides=2, padding='same')(x)
    x = LeakyReLU(0.2)(x)
    x = Flatten()(x)
    x = Dense(1, activation='sigmoid')(x)
    return Model(img_input, x, name="Discriminator")
```

Python
GetBot AI: Explain | Find Error | Find Resource Leaks

```python
z_dim = 100
generator = build_generator(z_dim)
discriminator = build_discriminator()

discriminator.compile(loss='binary_crossentropy', optimizer=Adam(0.0002, 0.5), metrics=['accuracy

discriminator.trainable = False
z_input = Input(shape=(z_dim,))
img = generator(z_input)
valid = discriminator(img)

combined = Model(z_input, valid)
combined.compile(loss='binary_crossentropy', optimizer=Adam(0.0002, 0.5))
```

Activate Windows
Go to Settings to activate Windows.

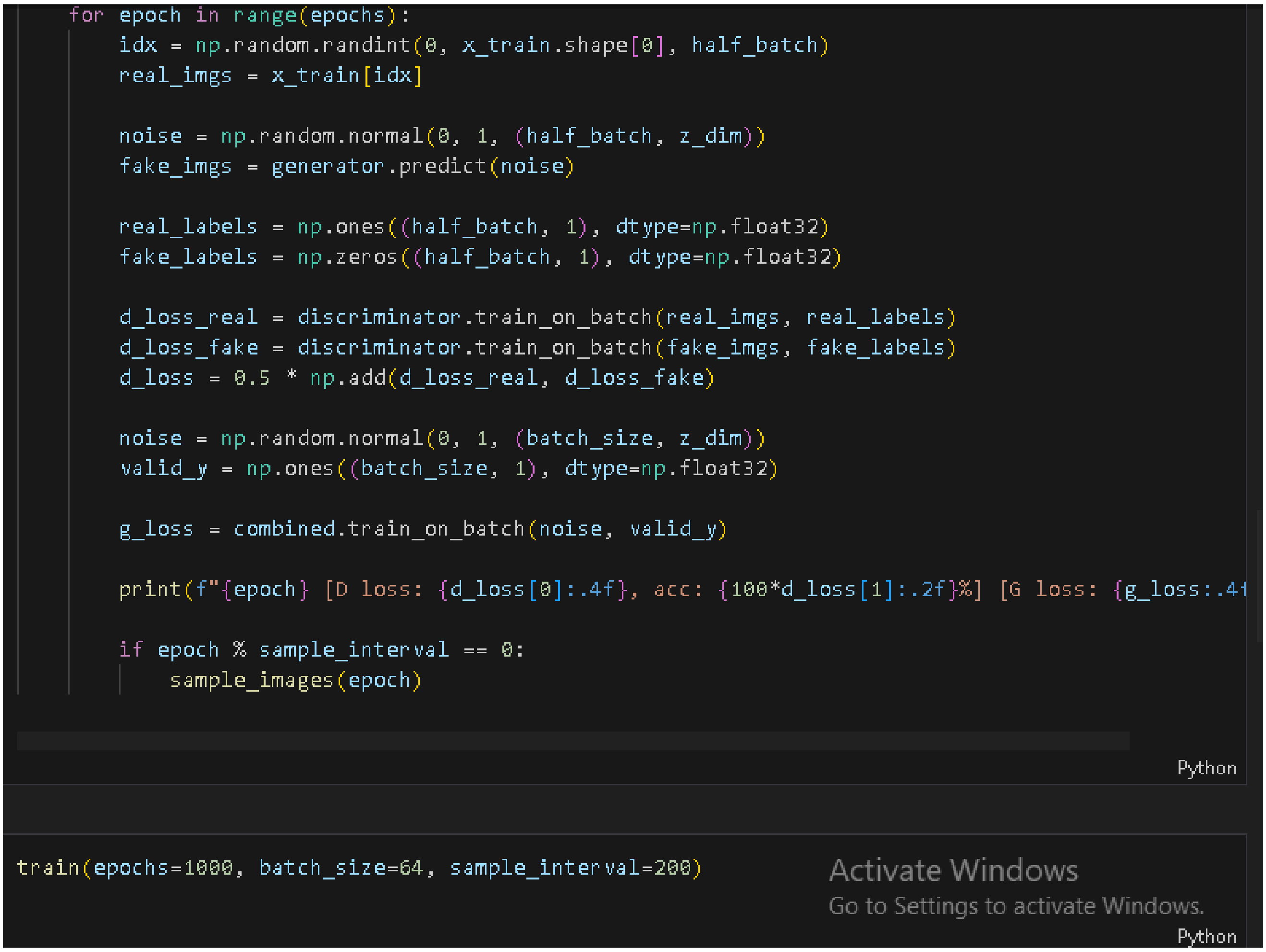

GetBot AI: Explain | Find Error | Find Resource Leaks

```
def sample_images(epoch, r=5, c=5):
    noise = np.random.normal(0, 1, (r * c, z_dim))
    gen_imgs = generator.predict(noise)
    gen_imgs = 0.5 * gen_imgs + 0.5  # Convert [-1,1] to [0,1]

    fig, axs = plt.subplots(r, c, figsize=(5, 5))
    cnt = 0
    for i in range(r):
        for j in range(c):
            axs[i, j].imshow(gen_imgs[cnt, :, :, 0], cmap='gray')
            axs[i, j].axis('off')
            cnt += 1
    os.makedirs("stylegan_outputs", exist_ok=True)
    fig.savefig(f"stylegan_outputs/mnist_epoch_{epoch}.png")
    plt.close()
```

GetBot AI: Explain | Find Error | Find Resource Leaks

```
def train(epochs, batch_size=64, sample_interval=200):
    half_batch = batch_size // 2

    for epoch in range(epochs):
        idx = np.random.randint(0, x_train.shape[0], half_batch)
        real_imgs = x_train[idx]

        noise = np.random.normal(0, 1, (half_batch, z_dim))
        fake_imgs = generator.predict(noise)
```

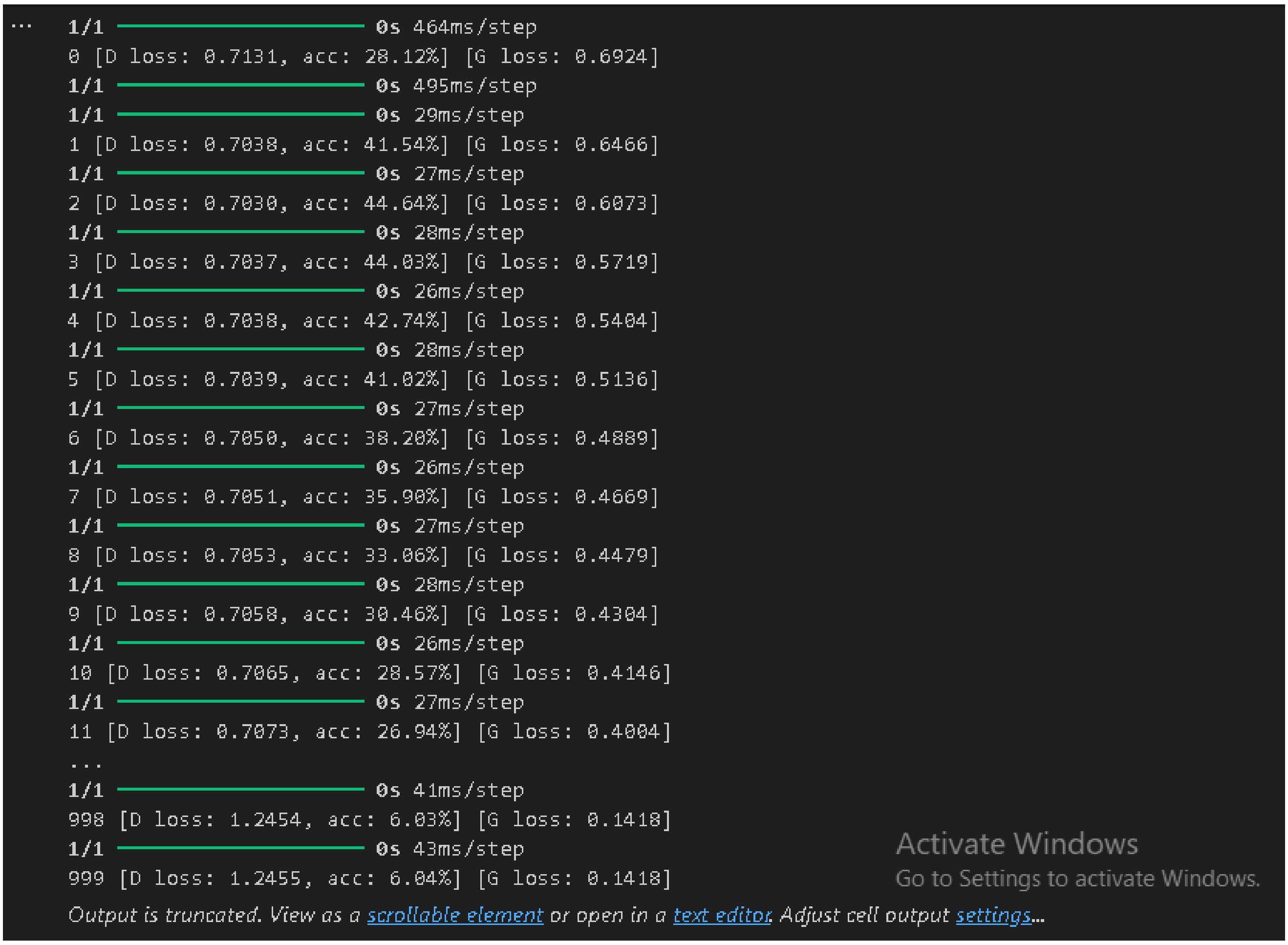
Activate Windows
Go to Settings to activate Windows.



```python
    for epoch in range(epochs):
        idx = np.random.randint(0, x_train.shape[0], half_batch)
        real_imgs = x_train[idx]

        noise = np.random.normal(0, 1, (half_batch, z_dim))
        fake_imgs = generator.predict(noise)

        real_labels = np.ones((half_batch, 1), dtype=np.float32)
        fake_labels = np.zeros((half_batch, 1), dtype=np.float32)

        d_loss_real = discriminator.train_on_batch(real_imgs, real_labels)
        d_loss_fake = discriminator.train_on_batch(fake_imgs, fake_labels)
        d_loss = 0.5 * np.add(d_loss_real, d_loss_fake)

        noise = np.random.normal(0, 1, (batch_size, z_dim))
        valid_y = np.ones((batch_size, 1), dtype=np.float32)

        g_loss = combined.train_on_batch(noise, valid_y)

        print(f"{epoch} [D loss: {d_loss[0]:.4f}, acc: {100*d_loss[1]:.2f}%] [G loss: {g_loss:.4

        if epoch % sample_interval == 0:
            sample_images(epoch)
```

```python
train(epochs=1000, batch_size=64, sample_interval=200)
```

Activate Windows
Go to Settings to activate Windows.



```
...  1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 464ms/step
     0 [D loss: 0.7131, acc: 28.12%] [G loss: 0.6924]
     1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 495ms/step
     1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 29ms/step
     1 [D loss: 0.7038, acc: 41.54%] [G loss: 0.6466]
     1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 27ms/step
     2 [D loss: 0.7030, acc: 44.64%] [G loss: 0.6073]
     1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 28ms/step
     3 [D loss: 0.7037, acc: 44.03%] [G loss: 0.5719]
     1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 26ms/step
     4 [D loss: 0.7038, acc: 42.74%] [G loss: 0.5404]
     1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 28ms/step
     5 [D loss: 0.7039, acc: 41.02%] [G loss: 0.5136]
     1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 27ms/step
     6 [D loss: 0.7050, acc: 38.20%] [G loss: 0.4889]
     1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 26ms/step
     7 [D loss: 0.7051, acc: 35.90%] [G loss: 0.4669]
     1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 27ms/step
     8 [D loss: 0.7053, acc: 33.06%] [G loss: 0.4479]
     1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 28ms/step
     9 [D loss: 0.7058, acc: 30.46%] [G loss: 0.4304]
     1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 26ms/step
     10 [D loss: 0.7065, acc: 28.57%] [G loss: 0.4146]
     1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 27ms/step
     11 [D loss: 0.7073, acc: 26.94%] [G loss: 0.4004]
     ...
     1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 41ms/step
     998 [D loss: 1.2454, acc: 6.03%] [G loss: 0.1418]
     1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step
     999 [D loss: 1.2455, acc: 6.04%] [G loss: 0.1418]
```

Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...

Activate Windows
Go to Settings to activate Windows.