Build and train a GAN for generating hand-written digits

1. **Hyperparameters**:
   - `latent_dim`: Size of the random noise vector used as input for the generator.
   - `hidden_dim`: Size of the hidden layers for both the generator and discriminator.
   - `image_dim`: Flattened size of the MNIST image (28x28 pixels = 784).
   - `num_epochs`: Number of training epochs.
   - `batch_size`: Batch size for training.
   - `lr`: Learning rate for the Adam optimizer.
   - `beta1`: The exponential decay rate for the first moment estimates in the Adam optimizer.
2. **Device Setup**: The code will automatically use a GPU if available.
3. **MNIST Dataset**: The MNIST dataset is loaded and normalized to a range of [-1, 1] to match the output range of the generator.
4. **Generator Model**:
   - Takes a random noise vector (`latent_dim`) as input.
   - Uses fully connected layers with ReLU activations and batch normalization.
   - The output is a 28x28 image with pixel values between [-1, 1] (tanh activation).
5. **Discriminator Model**:
   - Takes an image (either real or generated) and classifies it as real or fake.
   - Uses fully connected layers with LeakyReLU activations.
6. **Loss Function**:
   - **Adversarial loss** (`BCELoss`): Binary Cross-Entropy loss is used for both the generator and discriminator to measure how well each can distinguish real from

fake images.

7. **Optimizers**: Adam optimizers are used for both the generator and discriminator with a learning rate of `0.0002` and β1 of `0.5`.

8. **Training Loop**:

   ○ **Discriminator**: It is trained using both real images (from MNIST) and fake images (generated by the generator).

   ○ **Generator**: It is trained to generate images that can fool the discriminator into classifying them as real.

9. **Image Generation**:

   ○ Every 10 epochs, the generator creates 16 fake images, which are saved as `.png` files to monitor the progress of training.

   ○ At the end of training, the generator creates a final set of images and displays them.

10. **Saving the Model**: The trained generator model is saved for later use.