**Question 1**

**GAN Architecture**

**Adversarial Process in GAN Training:** Generative Adversarial Networks (GANs) consist of two models: a **Generator** and a **Discriminator**, trained simultaneously in a competitive setting.

- **Generator (G):** Takes random noise (z) as input and generates fake data samples (G(z)) that mimic the real data distribution. Its goal is to "fool" the discriminator by producing samples indistinguishable from real data.
- **Discriminator (D):** Takes both real data (x) and fake data (G(z)) as input and outputs a probability indicating whether the input is real (close to 1) or fake (close to 0). Its goal is to correctly distinguish real data from fake data.
- **Adversarial Process:** The generator and discriminator are trained in opposition:
    - The discriminator is trained to maximize its ability to classify real vs. fake samples (optimizing D to output 1 for real data and 0 for fake data).
    - The generator is trained to minimize the discriminator's ability to detect fake samples (optimizing G to make D(G(z)) close to 1).
    - This is a minimax game where the generator improves by trying to "trick" the discriminator, and the discriminator improves by becoming better at detecting fakes.
- **Improvement Through Competition:** As the discriminator gets better at identifying fakes, the generator must produce more realistic samples to fool it. Conversely, as the generator produces better fakes, the discriminator must refine its classification. This competition drives both models toward equilibrium, where the generator produces samples indistinguishable from real data ($D(G(z)) \approx 0.5$).

**GAN Architecture Diagram :**

**[Random Noise (z)] --> [Generator (G)] --> [Fake Data (G(z))]**

**|**

**v**

**[Real Data (x)] ----> [Discriminator (D)] <---- [Fake Data (G(z))]**

**|**

**v**

**[Output: P(real) ∈ [0,1]]**

- **Data Flow:**
  Noise (z) → Generator → Fake Data (G(z)).
  Real Data (x) and Fake Data (G(z)) → Discriminator → Probability score.
- **Objectives:**
  Generator: Minimize log(1 - D(G(z))) or maximize D(G(z)).
  Discriminator: Maximize log(D(x)) + log(1 - D(G(z))).