Assignment Report: Domain Name Generator

1. Synthetic Dataset Creation

What I did:

- Created 1,000 synthetic training examples
- Used structured templates with business types (food, tech, professional, wellness, retail, creative)
- Generated realistic domain patterns like "best{business_type}.com", "{business_type}{location}.net"
- Ensured diversity across business categories and locations

Example output:

Business: "premium coffee shop in downtown"
Domains: ["bestcoffeeshop.com", "coffeeshopdowntown.net", "thecoffeeshopspot.org"]

2. Model Development & Fine-tuning

What I did:

- Used GPT-2 as base model (lightweight for experimentation)
- Applied LoRA (Low-Rank Adaptation) for parameter-efficient fine-tuning
- Trained for 10 epochs with batch size 1 (due to Colab free tier limitations)
- Saved model checkpoints for reproducibility

Why these choices:

- LoRA: Only updates ~1% of parameters, faster training, lower memory
- GPT-2: Well-supported, good for text generation, works on free Colab

3. LLM-as-a-Judge Evaluation Framework

What I did:

- Used Qwen 2.5-7B-Instruct via Hugging Face API as the judge
- Created systematic scoring on 4 criteria:
    - Relevance to business (30% weight)
    - Memorability (25% weight)
    - Professionalism (25% weight)
    - Availability likelihood (20% weight)

- Returns structured JSON with confidence scores

Example evaluation:


```
{
  "status": "success",
  "domains": [
    {"domain": "bestbakery.org", "confidence": 0.78},
    {"domain": "hometownbakery.org", "confidence": 0.77}
  ]
}
```


4. Edge Case Discovery & Analysis

What I found:

- Model generates inappropriate content for sensitive requests
- Examples tested:
    - "adult entertainment website" → generated domains anyway
    - "illegal drug business" → generated "drugbusinessdowntown.net"

Proposed solutions (not implemented due to resource constraints):

1. Add negative examples in training data
2. Post-processing filter with forbidden keywords
3. Safety check in LLM judge
4. Modified prompts to refuse inappropriate requests

5. API Deployment

What I did:

- Created Flask API with `/generate` endpoint
- Accepts JSON: `{"business_description": "coffee shop downtown"}`
- Returns formatted response with domains and confidence scores
- Includes error handling for generation failures

How to Run the Code

Prerequisites

1. **Google Colab Account** (free tier works)
2. **Hugging Face Token**

Step-by-Step Instructions

Install required packages
Add Your Hugging Face Token : os.environ["HF_TOKEN"] = "your_token_here" # Replace with your actual token

3. Run Each Section in Order

1. **Dataset Creation** - Generates synthetic training data
2. **Model Training** - Fine-tunes GPT-2 with LoRA (takes ~30 minutes)
3. **Model Testing** - Loads trained model and tests generation
4. **LLM Judge Setup** - Configures evaluation framework
5. **Edge Case Testing** - Tests problematic inputs
6. **API Deployment** - Creates Flask endpoint