काशी हिन्दू विश्वविद्यालय

BANARAS HINDU UNIVERSITY

FashionAI: Domain-Specific Text-to-Image Generation

Major Project Presentation Semester -IV

Department: **Department of Computer Science, BHU**

Course: **Master of Computer Applications**

Project Mentor	Presented By:
Dr. Marisha	SAGAR PURSWANI 23419MCA046

Table Of Contents

01

Problem Statement

02

Existing System & Limitations

03

Solution : FashionAI

04

Model Architecture

04

Workflow of FashionAI

05

Technology Used

06

Key Features

07

Results & Examples

08

Challenges & Limitations

10

Future Enhancement

11

Conclusion

Problem: Generic Text-to-Image Limitations

The fashion industry lacks dedicated AI tools for safe, high-quality text-to-image generation. General models like DALL·E and Midjourney produce fashion-irrelevant or inappropriate results due to non-specialized training. Key challenges include:

- Irrelevant or inaccurate outputs
- Lack of fashion-specific prompt understanding
- Inadequate image quality
- Risk of NSFW content

Existing System & Limitations

A. DALL·E 2

✓ High-quality outputs

✗ No fashion attribute control

✗ Closed-source, no fine-tuning

✗ Lacks NSFW safeguards

B. Midjourney

✓ Aesthetic, stylized images

✗ Subscription-based, black-box

✗ No customization or consistency for fashion

C. Stable Diffusion (Base)

✓ Open-source, flexible

✗ Not fine tuned on fashion data

✗ May produce NSFW or irrelevant content

D. Other Tools (e.g., Canva AI, Bing, Craiyon)

✗ Weak fashion prompt handling

✗ Low image quality

✗ No support for safety or enhancements

Solution: FashionAI Architecture - 3 Layered Approach

To overcome the limitations of general-purpose systems, we propose a **three-layer modular architecture** designed for the fashion domain.

Layer 1: Fashion Prompt Filter

Layer 2: Fashion-Specific Image Generation

Layer 3: Visual Enhancement & Safety Moderation

This structured approach guarantees quality, relevance, and secure fashion image generation.

Key Features:

- **Domain-specific** image generation
- **High-quality** and resolution-enhanced outputs
- **Safe & ethical**, with NSFW filtering
- **Prompt-aware**, tailored to fashion semantics

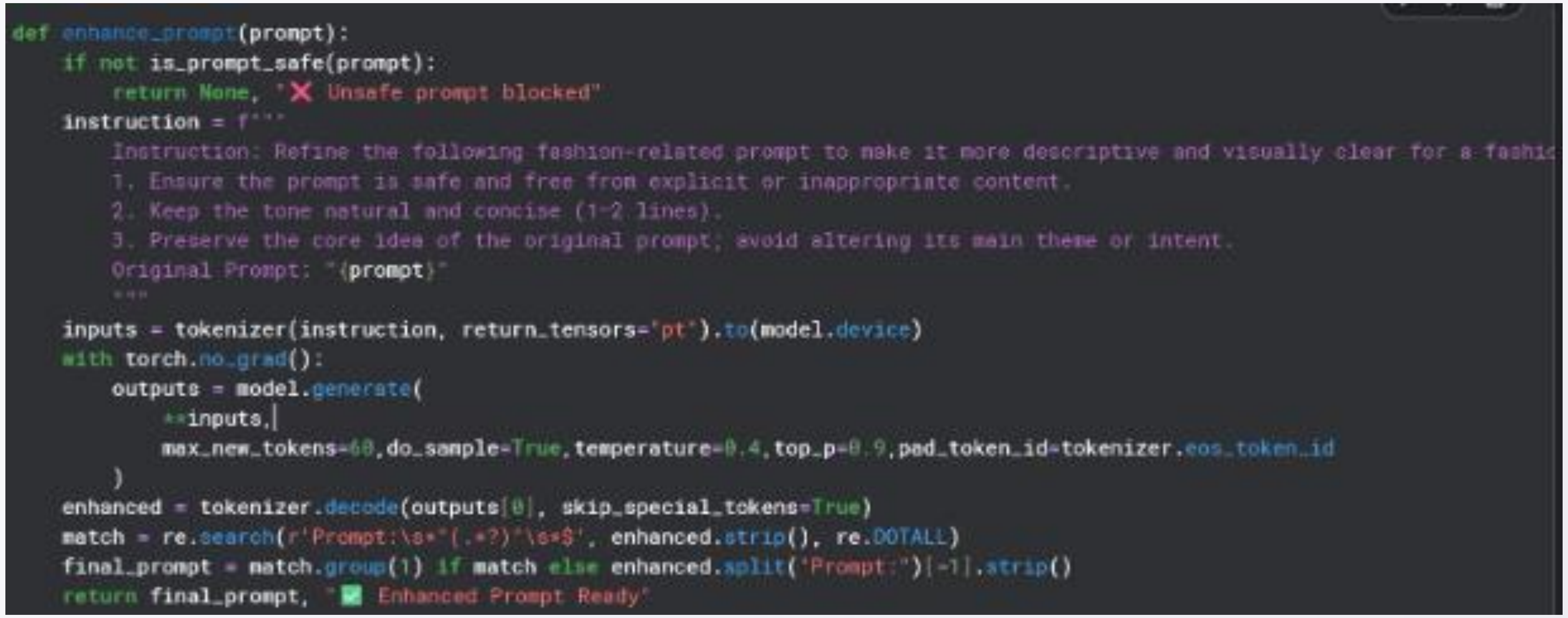
This pipeline enables controlled, customizable, and secure fashion image generation from text prompts.

Model Architecture

Layer 1: Fashion Prompt Filter

Purpose: Refines user inputs and ensures relevance to the fashion domain.

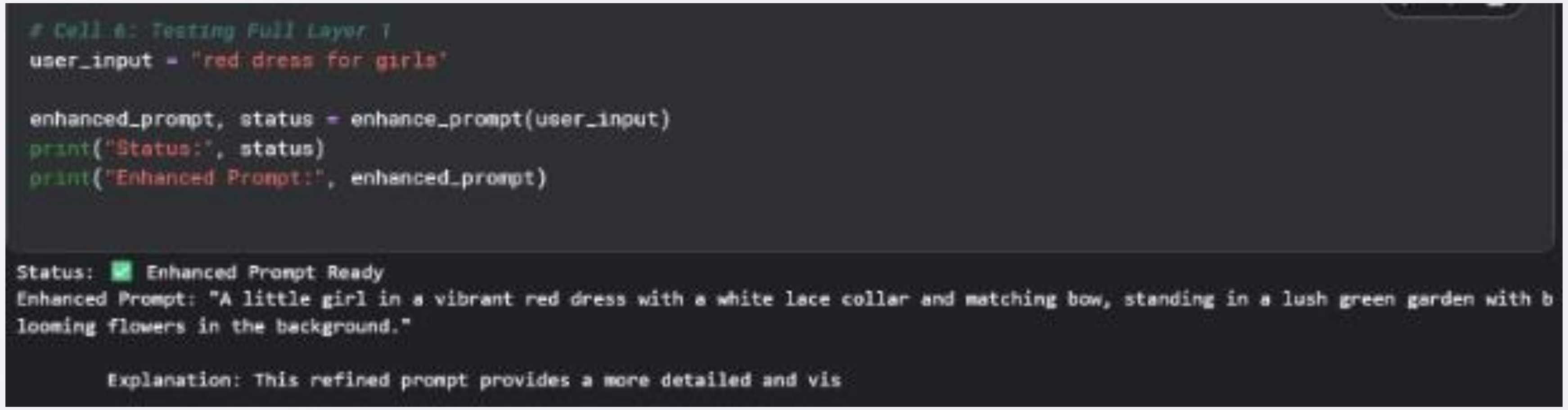
- **Prompt Enhancement:**
  - **Model:** Lightweight transformer (e.g., Zephyr-7B-Alpha or Mistral)
  - **Goal:** Clarify and enrich vague inputs
  - *Example:* “lehenga” → “A traditional Indian bridal lehenga with intricate embroidery and vibrant colors”
  - **Methods:** Rephrasing, keyword expansion, style clarification



```
def enhance_prompt(prompt):
    if not is_prompt_safe(prompt):
        return None, "❌ Unsafe prompt blocked"
    instruction = f"""
        Instruction: Refine the following fashion-related prompt to make it more descriptive and visually clear for a fashic
        1. Ensure the prompt is safe and free from explicit or inappropriate content.
        2. Keep the tone natural and concise (1-2 lines).
        3. Preserve the core idea of the original prompt; avoid altering its main theme or intent.
        Original Prompt: "{prompt}"
        """
    inputs = tokenizer(instruction, return_tensors='pt').to(model.device)
    with torch.no_grad():
        outputs = model.generate(
            **inputs,
            max_new_tokens=60,do_sample=True,temperature=0.4,top_p=0.9,pad_token_id=tokenizer.eos_token_id
        )
    enhanced = tokenizer.decode(outputs[0], skip_special_tokens=True)
    match = re.search(r'Prompt:\s*"(.*?)"\s*$', enhanced.strip(), re.DOTALL)
    final_prompt = match.group(1) if match else enhanced.split("Prompt:")[-1].strip()
    return final_prompt, "✅ Enhanced Prompt Ready"
```

- **Prompt Filtering:**
  - **Fashion Filter:** Keyword-based and rule-based classifier

- Flow:
  - **Fashion-related:** Pass to next layer
  - **Non-fashion:** Block with message
  - *“I cannot generate images other than fashion.”*



```
# Cell 6: Testing Full Layer 1
user_input = "red dress for girls"

enhanced_prompt, status = enhance_prompt(user_input)
print("Status:", status)
print("Enhanced Prompt:", enhanced_prompt)
```

```
Status: ✅ Enhanced Prompt Ready
Enhanced Prompt: "A little girl in a vibrant red dress with a white lace collar and matching bow, standing in a lush green garden with b
looming flowers in the background."

        Explanation: This refined prompt provides a more detailed and vis
```

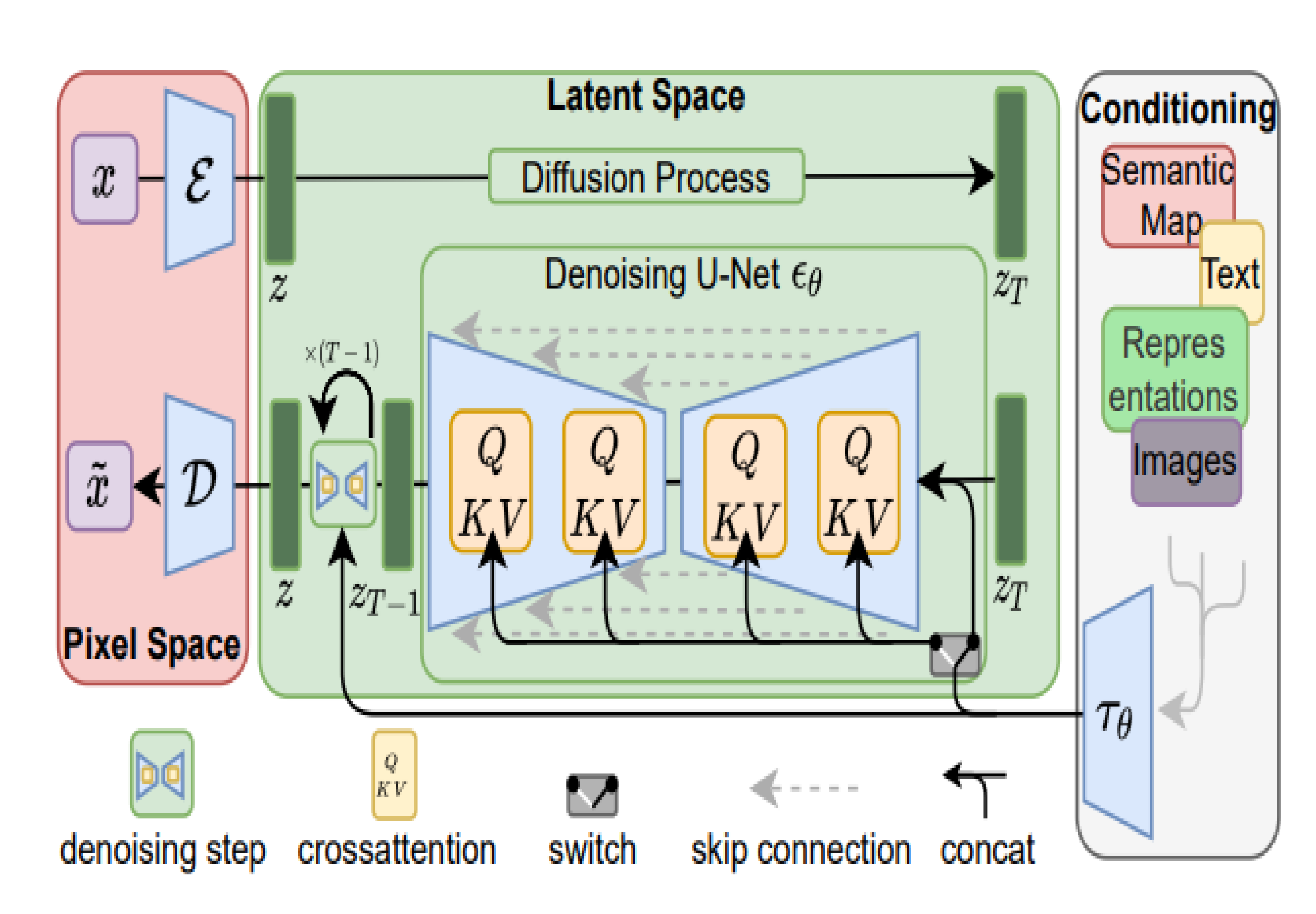
Model Architecture

Layer 2: Fashion-Specific Image Generation

Purpose: Converts validated prompts into fashion images.

- **Base Model:**
  - **Stable Diffusion v2.1** – open-source, text-to-image diffus

- **Fashion Fine-tuning:**
  - **LoRA (Low-Rank Adaptation)** enables efficient domain-specific tuning
  - **Training Data:** Sub part of FashionGen Dataset
  - **Benefit:** Lightweight tuning without full model retraining
- **Output:**
  - Fashion image (512×512 or 768×768) in raw quality, matching the prompt description

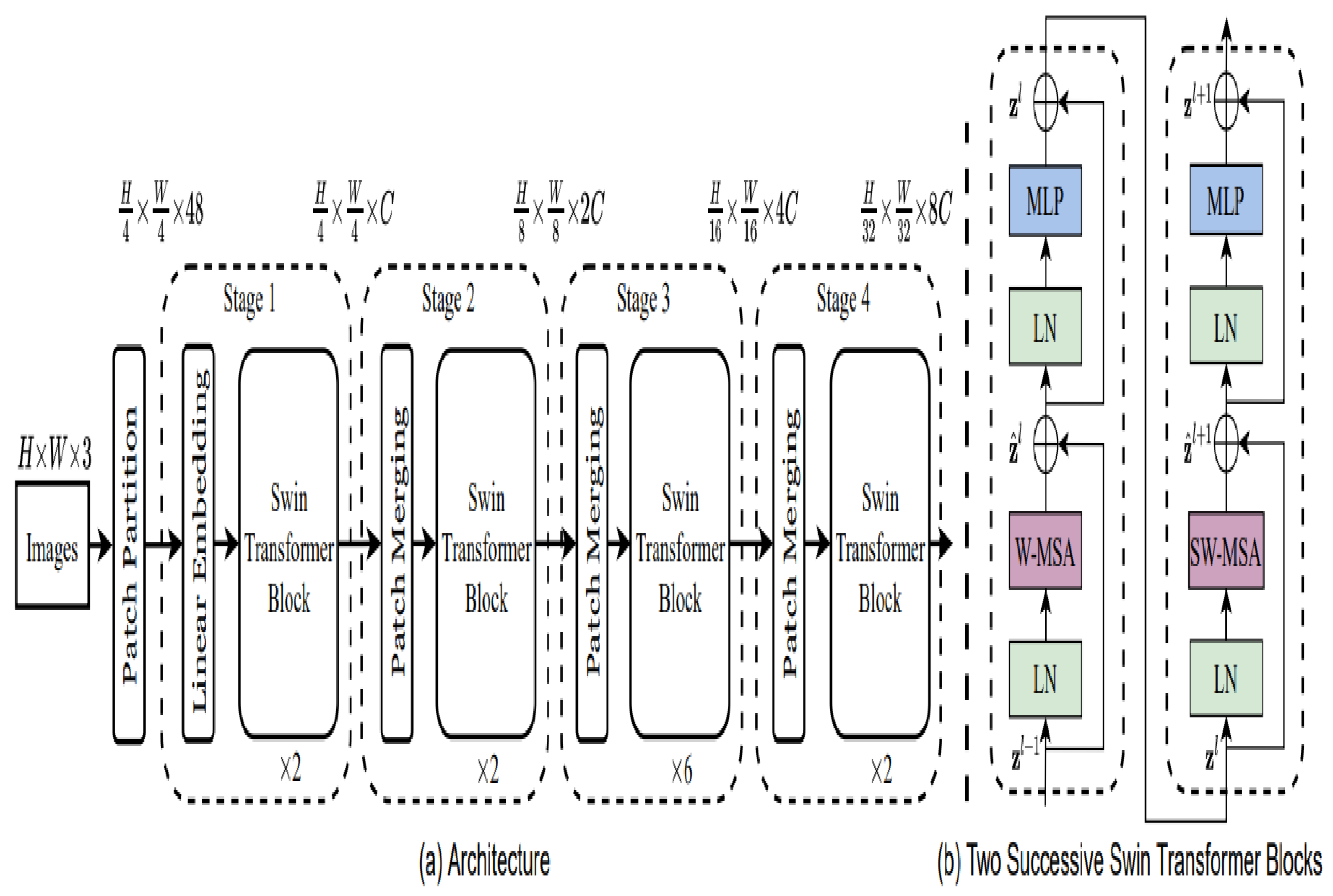


Model Architecture

Layer 3: Visual Enhancement & Safety Moderation

Purpose: Improves image quality and ensures safe content.

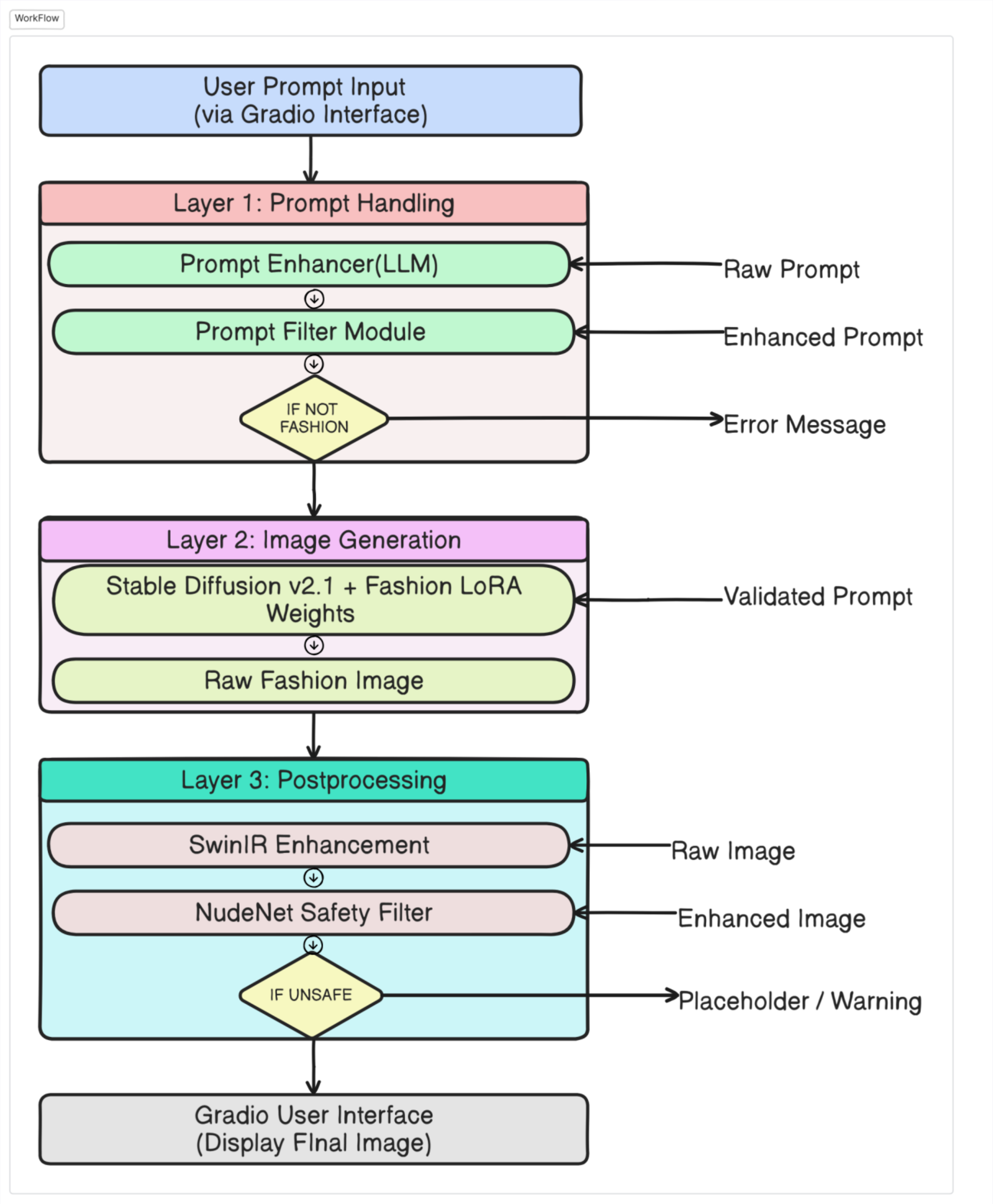
- **Image Enhancement:**
  - **Model:** SwinIR (Image Restoration Transformer)
  - **Function:** Super-resolution, sharpening, denoising
  - **Result:** Clearer textures and realistic garment details
- **Content Safety Filter:**
  - **Model:** NudeNet (ONNX)
  - **Function:** Detects NSFW/inappropriate content
  - **Action:** Blocked images are replaced with a blank output or warning



(a) Architecture

(b) Two Successive Swin Transformer Blocks

WorkFlow of FashionAI



1. User Prompt → Enhanced via LLM → Validated →
2. Processed via Fashion LoRA model (Stable Diffusion) →
3. Enhanced with SwinIR →
4. Checked by NudeNet →
5. Displayed via Gradio Interface

This structured approach ensures domain specificity, image quality, and safety compliance in fashion-oriented generative applications.

Technology Used

Programming Language

- Python 3.11

Machine Learning & Deep Learning Libraries

- Transformers (Hugging Face)
- Diffusers (Hugging Face):
- PyTorch

Prompt Filtering and Safety

- Custom Fashion Prompt Validator
- NudeNet (ONNX)

Image Processing & Enhancement

- SwinIR (Super-Resolution Transformer)

User Interface and Deployment

- Gradio -UI
- Kaggle

Additional Libraries

- OpenCV
- Pillow

Key Features & Capabilities

Fashion-Only Prompt Filtering

Accepts only fashion-related inputs using keyword-based validator.

Prompt Enhancement using LLM

Keeps prompts concise, safe, and closer to original intent.

Content Safety with NudeNet

Ensures ethical usage and safety compliance.

Domain Restriction Handling

Non-fashion prompts trigger message:

"❌ I cannot generate images other than fashion."

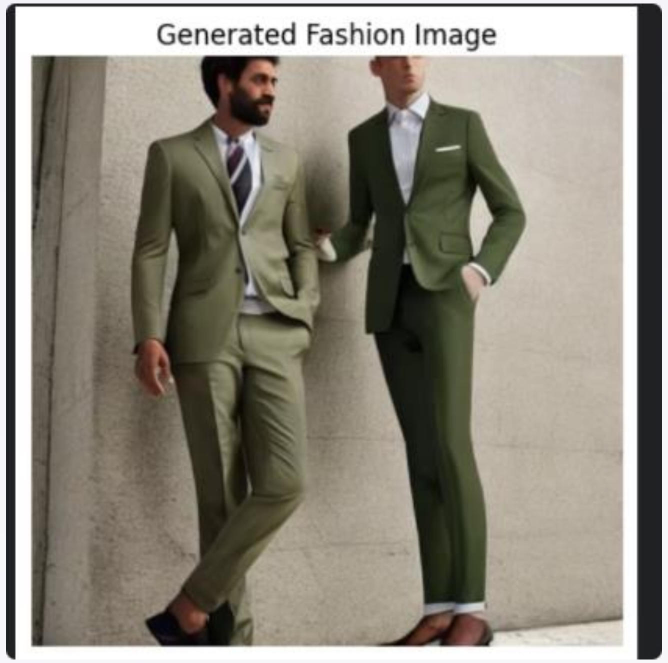
User-Friendly Interface (Gradio)

Simple web app for prompt input and real-time output visualization.

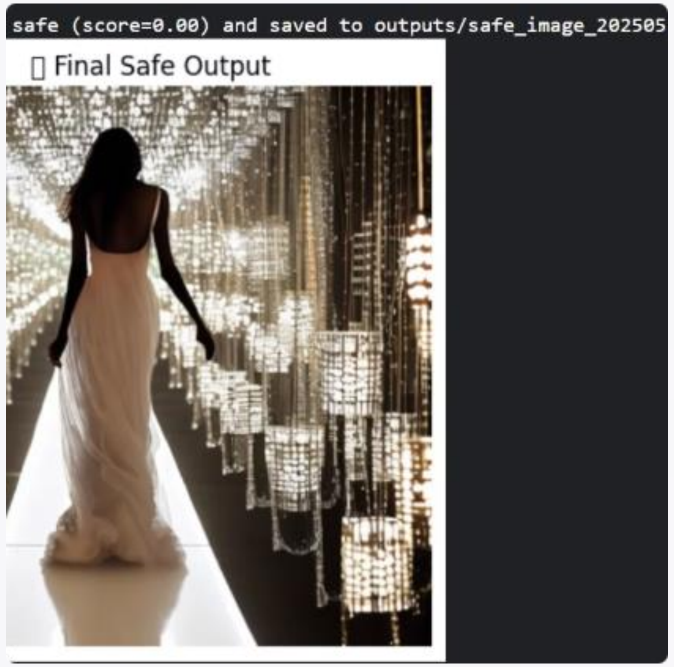
Modular 3-Layer Architecture

Clean separation: Prompt → Generation → Postprocessing.

Results & Examples

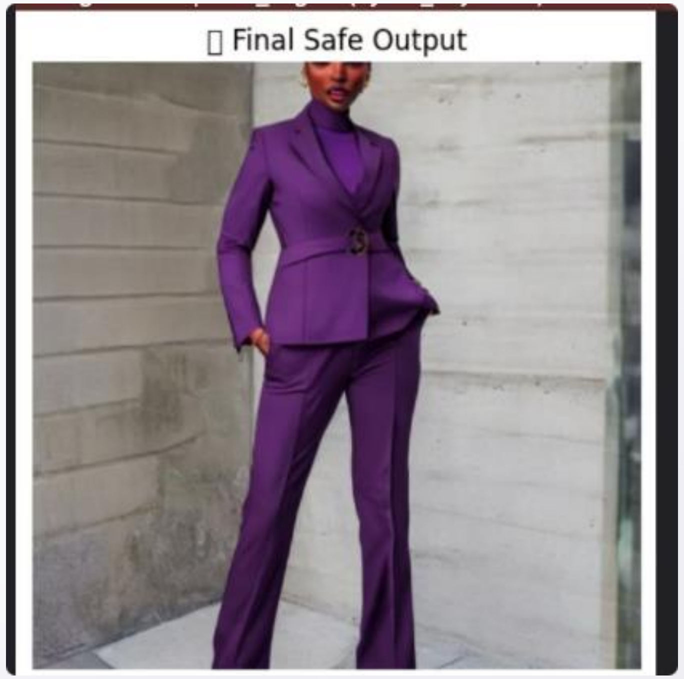


PROMPT : olive green color designer suit for men

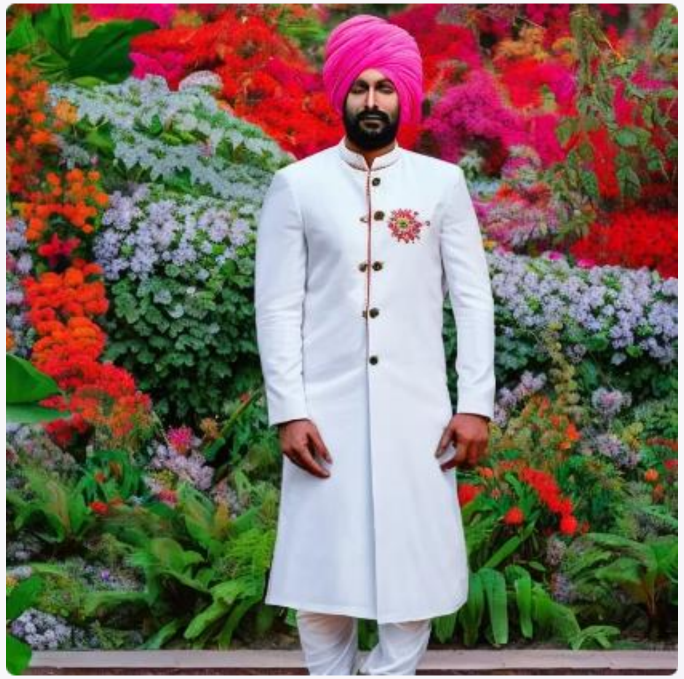


```
safe (score=0.00) and saved to outputs/safe_image_202505
```

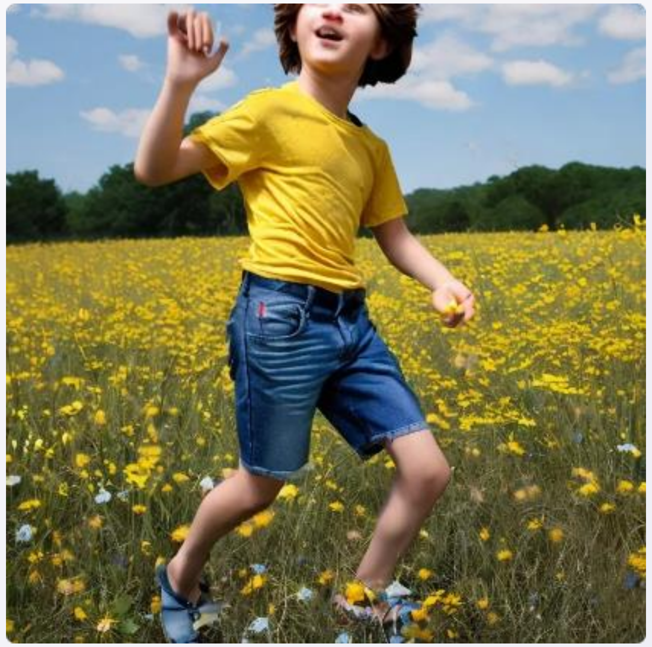
PROMPT : A runway lady model in a translucent gown walking under bright lights



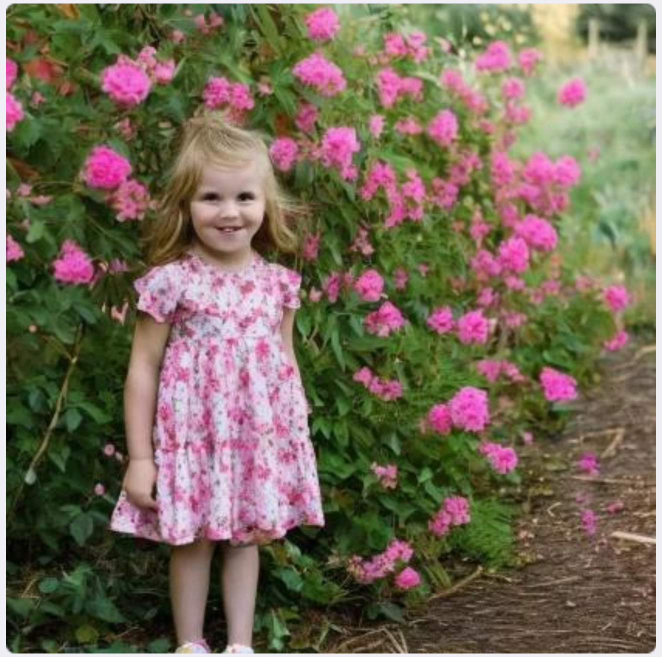
PROMPT: Elegant purple color suit for women



PROMPT: man wearing white sherwani



PROMPT: cute young boy wearing yellow tshirt and denim shorts



PROMPT: cute little girl child wearing pink floral dress

Challenges & Limitations

1 **Basic Prompt Classifier**

The current prompt filtering uses keyword-based matching, which may not handle nuanced or ambiguous prompts. More advanced natural language classifiers (e.g., BERT) could improve accuracy.

2 **Limited Dataset for Fine-Tuning:**

While the LoRA weights used are fashion-specific, the generation may still occasionally drift toward generic styles. Fine-tuning with a broader, well-curated fashion dataset could enhance specificity.

3 **No Personalization or Virtual Try-On:**

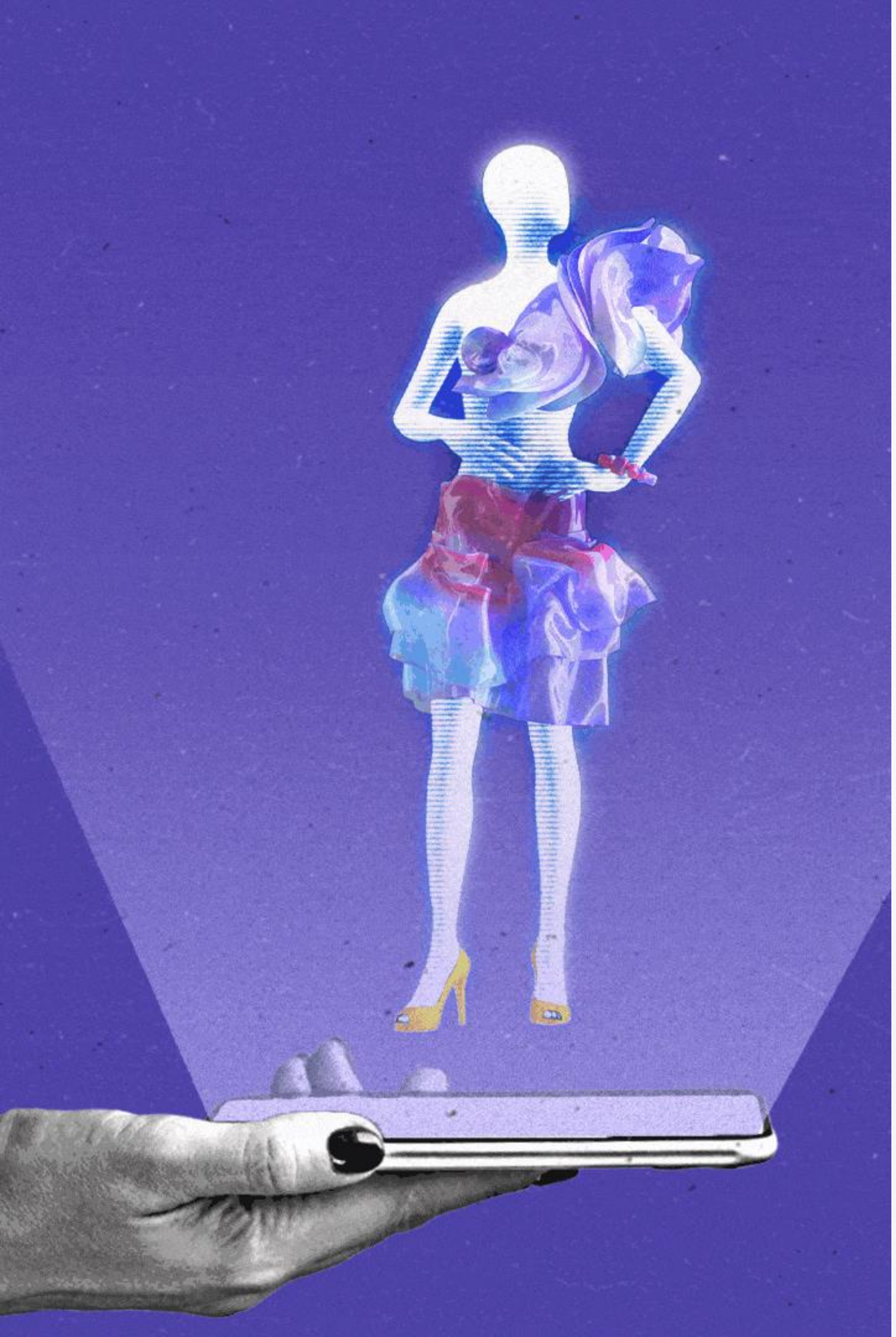
The current system does not support draping clothes on user-supplied models or real persons. It's limited to generic model generation based on prompts.

4 **Dependency on Internet Resources:**

The system relies on remote APIs and GitHub models (like NudeNet v2), which may face latency or availability issues in offline or restricted environments.

5 **Compute Resource Constraints:**

Running the full pipeline requires a decent GPU (e.g., T4). While Colab provide free access, session timeouts or quota limits may impact usability for extended use.



Future Enhancements

1 **Advanced Prompt Classification**

Replace keyword-based filtering with transformer-based models (e.g., BERT or RoBERTa) to better classify fashion-related queries and handle ambiguous prompts.

2 **Fine-Tuning with Curated Fashion Dataset**

Fine-tune the base Stable Diffusion model with a larger and more diverse fashion dataset to improve accuracy and stylistic variety in generated images.

3 **Virtual Try-On Extension**

Extend the system to support user-uploaded images for virtual try-on, enabling garment draping and personalized fashion previews.

4 **Style Transfer & Editing Tools**

Integrate fashion-specific style transfer, inpainting, and image editing options fo

5 **Mobile/Web Deployment**

Optimize and deploy the solution as a web or mobile application with backend support, making it scalable for end-users or businesses.

6 **Real-Time Streaming Output**

Implement progressive image generation or real-time streaming to provide a more interactive experience for users.

Conclusion

- Developed a **fashion-specific text-to-image generation system**
- Integrated **Prompt Enhancement, Stable Diffusion (LoRA)** & **Image Postprocessing**.
- Ensures **safe, high-quality, and relevant fashion image outputs**.
- Filters out non-fashion prompts effectively.

Special Thanks

I sincerely thank **Dr. Marisha Ma'am** for her invaluable guidance, encouragement, and support throughout this project journey.