FASHION RECOMMENDATION SYSTEM

A Fashion Recommendation System using Image Features leverages computer vision and machine learning techniques to analyze fashion items' visual aspects (like colour, texture, and style) and recommend similar or complementary products to users. So, if you want to learn how to build a Fashion Recommendation System by utilizing image features, this article is for you. In this article, I'll take you through the task of building a Fashion Recommendation System utilizing Image Features using the Python programming language.

Building a fashion recommendation system using image features involves several key steps, leveraging both computer vision and machine learning techniques. Below is a detailed process you can follow to build a fashion recommendation system using image features:

Assemble a diverse dataset of fashion items. This dataset should include a wide variety of items with different colours, patterns, styles, and categories.
Ensure all images are in a consistent format (e.g., JPEG, PNG) and resolution.
Implement a preprocessing function to prepare images for feature extraction.
Choose a pre-trained CNN model such as VGG16, ResNet, or InceptionV3. These models, pre-trained on large datasets like ImageNet, are capable of extracting powerful feature representations from images.
Pass each image through the CNN model to extract features.
Define a metric for measuring the similarity between feature vectors.
Rank the dataset images based on their similarity to the input image and recommend the top N items that are most similar.
Implement a final function that encapsulates the entire process from pre-processing an input image, extracting features, computing similarities, and outputting recommendations.

In [2]:
```
from zipfile import ZipFile
import os

zip_file_path = "C:\\Users\\kourg\\Downloads\\women-fashion.zip"
extraction_directory = '/content/women_fashion/'

if not os.path.exists(extraction_directory):
    os.makedirs(extraction_directory)

with ZipFile(zip_file_path, 'r') as zip_ref:
    zip_ref.extractall(extraction_directory)

extracted_files = os.listdir(extraction_directory)
print(extracted_files[:10])
```

```
['women fashion', '__MACOSX']
```

In the above code, a zip file named 'women fashion.zip' located at the path: '/content/women fashion.zip' on Google Colab is being extracted to a specified directory: '/content/women_fashion/'. Initially, we check if the extraction directory exists, and if it does not, the directory is created using os.makedirs(). Then, using Python's ZipFile module, the zip file is opened in read mode, and its contents are extracted to the designated directory.

The zip file contains a directory named women fashion and some metadata used by macOS (**__MACOSX). Let's ignore the macOS metadata and focus on the women fashion directory, listing its contents to understand the types and number of images we have:**

In [3]:
```
# correcting the path to include the 'women fashion' directory and listing its contents
extraction_directory_updated = os.path.join(extraction_directory, 'women fashion')

# list the files in the updated directory
extracted_files_updated = os.listdir(extraction_directory_updated)
extracted_files_updated[:10], len(extracted_files_updated)
```

Out[3]:
```
(['.DS_Store',
  'anarkali suit with a long, olive green kurta adorned with intricate embroidery around the neckline and cuffs, paired with matching fitted trousers.jpg',
  'Anarkali suit with a modern twist.jpg',
  'Anarkali suit with fitted bodice with a high neckline.jpg',
  'anarkali suit with intricate silver embellishments on the neckline, sleeves.jpg',
  'anarkali suit with lavender in color with intricate white patterns throughout the fabric.jpg',
  'anarkali suit. It consists of a turquoise skirt with detailed golden embroidery, a multicolored blouse with floral patterns, and an orange dupatta with lace borders.jpg',
  'ark green, knee-length dress with short sleeves and a white, patterned neckline.jpg',
  'beige top adorned with black dots and a green skirt.jpg',
  'black and white gingham checkered A-line dress with a flared skirt.jpg'],
 97)
```

In [9]:

```
from PIL import Image
import os
import matplotlib.pyplot as plt

# function to load and display an image
def display_image(file_path):
    image = Image.open(file_path)
    plt.imshow(image)
    plt.axis('off')
    plt.show()

# display the first image to understand its characteristics
extraction_directory_updated = os.path.join(extraction_directory, 'women fashion')
first_image_path = os.path.join(extraction_directory_updated, extracted_files_updated[1])
display_image(first_image_path)
```

In [10]:

```
import glob

# directory path containing your images
image_directory = '/content/women_fashion/women fashion'

image_paths_list = [file for file in glob.glob(os.path.join(image_directory, '*.*')) if file.endswith(('.jpg', '.png', '
# print the list of image file paths
print(image_paths_list)
```

['/content/women_fashion/women fashion\\anarkali suit with a long, olive green kurta adorned with intricate embroidery around the neckline and cuffs, paired with matching fitted trousers.jpg', '/content/women_fashion/women fashion\\Anarkali suit with a modern twist.jpg', '/content/women_fashion/women fashion\\Anarkali suit with fitted bodice with a high neckline.jpg', '/content/women_fashion/women fashion\\anarkali suit with intricate silver embellishments on the neckline, sleeves.jpg', '/content/women_fashion/women fashion\\anarkali suit with lavender in color with intricate white patterns throughout the fabric.jpg', '/content/women_fashion/women fashion\\anarkali suit. It consists of a turquoise skirt with detailed golden embroidery, a multicolored blouse with floral patterns, and an orange dupatta with lace borders.jpg', '/content/women_fashion/women fashion\\ark green, knee-length dress with short sleeves and a white, patterned neckline.jpg', '/content/women_fashion/women fashion\\beige top adorned with black dots and a green skirt.jpg', '/content/women_fashion/women fashion\\black and white gingham checkered A-line dress with a flared skirt.jpg', '/content/women_fashion/women fashion\\black double-breasted blazer with gold buttons, paired with a mid-length skirt.jpg', '/content/women_fashion/women fashion\\black dress with lace detailing.jpg', '/content/women_fashion/women fashion\\black dress with sparkling details.jpg', '/content/women_fashion/women fashion\\black floral saree.jpg', '/content/women_fashion/women fashion\\black jumpsuit with a diagonal stripe of glittering material running across the body.jpg', '/content/women_fashion/women fashion\\black lace bustier top paired with high-waisted black trousers.jpg', '/content/women_fashion/women fashion\\black off-shoulder dress with belt.jpg', '/content/women_fashion/women fashion\\black top with spaghetti straps and a black skirt adorned with a fringe detail at the hem.jpg', '/content/women_fashion/women fashion\\black top, white trousers.jpg', '/content/women_fashion/women fashion\\black, sequined dress with thin shoulder straps.jpg', '/content/women_fashion/women fashion\\blue dress with a floral pattern.jpg', '/content/women_fashion/women fashion\\bright red kurta with an intricate patterned design.jpg', '/content/women_fashion/women fashion\\bright red, form-fitting, strapless dress with a high slit on one side revealing part of the leg.jpg', '/content/women_fashion/women fashion\\bright red, sequined dress with thin shoulder straps.jpg', '/content/women_fashion/women fashion\\burgundy off-the-shoulder dress with an asymmetrical hemline.jpg', '/content/women_fashion/women fashion\\burnt orange long-sleeve top, a plaid skirt with brown tones, and a matching burnt orange beret.jpeg', '/content/women_fashion/women fashion\\chic and elegant outfit consisting of a pair of high-waisted wide-leg trousers paired with a long-sleeve sheer lace top.jpg', '/content/women_fashion/women fashion\\classic black slip dress with a midi length.jpg', '/content/women_fashion/women fashion\\colorful saree.jpg', '/content/women_fashion/women fashion\\cream-colored, long, open-front coat with wide lapels.jpg', '/content/women_fashion/women fashion\\dark blue, knee-length dress with thin straps.jpg', '/content/women_fashion/women fashion\\dark, elegant, sleeveless dress that reaches down to about mid-calf.jpg', '/content/women_fashion/women fashion\\deep burgundy, silky dress with long sleeves and a wrap-style front.jpg', '/content/women_fashion/women fashion\\dress with a classic black and white houndstooth pattern.jpg', '/content/women_fashion/women fashion\\fitted black dress that reaches down to mid-calf.jpg', '/content/women_fashion/women fashion\\fitted dress with a classic black and white houndstooth pattern.webp', '/content/women_fashion/women fashion\\fitted, cream-colored dress with long sheer sleeves.jpg', '/content/women_fashion/women fashion\\fitted, off-the-shoulder white dress with horizontal ribbed texture.jpg', '/content/women_fashion/women fashion\\fitted, ruched dress with a sweetheart neckline and long mesh sleeves.jpg', '/content/women_fashion/women fashion\\fitted, short, yellow dress with short sleeves.jpeg', '/content/women_fashion/women fashion\\floral dress with long sleeves.jpeg', '/content/women_fashion/women fashion\\flowing green off-shoulder dress.jpg', '/content/women_fashion/women fashion\\fluffy, possibly faux fur or eyelash knit sweater with a cropped cut top.jpg', '/content/women_fashion/women fashion\\form-fitting dress with a vibrant pattern of yellow, blue, and black.jpg', '/content/women_fashion/women fashion\\glamorous two-piece outfit featuring a sequined design.jpeg', '/content/women_fashion/women fashion\\green dress with a floral pattern, paired with matching green trousers.jpg', '/content/women_fashion/women fashion\\high-waisted white trousers paired with a structured, corset-style bodice featuring mesh.jpg', '/content/women_fashion/women fashion\\l-htdrss7133-honky-tonky-original-imagszfhfghhs2rp.jpeg', '/content/women_fashion/women fashion\\ladies-party-wear-cotton-short-dress-full.jpg', '/content/women_fashion/women fashion\\Latest-Party-Wear-One-Piece-Gown-For-Women.webp', '/content/women_fashion/women fashion\\light blue, sequined dress with a V-neckline, long sleeves.jpg', '/content/women_fashion/women fashion\\light brown sleeveless double-breasted blazer paired with matching straight-leg trousers.jpg', '/content/women_fashion/women fashion\\light green, short dress with lace detailing around the waist and the edges of the sleeves and hem.jpg', '/content/women_fashion/women fashion\\light peach, knee-length dress with off-the-shoulder sleeves.webp', '/content/women_fashion/women fashion\\long, elegant blue dress with an off-the-shoulder design with sheer sleeves.jpg', '/content/women_fashion/women fashion\\long, elegant, teal dress with a high slit on one side.jpg', '/content/women_fashion/women fashion\\long, flowing dress with a full skirt.jpg', '/content/women_fashion/women fashion\\long, flowing, pink dress with a sparkly texture.jpg', '/content/women_fashion/women fashion\\long, intricately designed dress with full sleeves.jpg', '/content/women_fashion/women fashion\\metallic-looking jacket with intricate designs and fringes hanging from the sleeves.jpg', '/content/women_fashion/women fashion\\mustard yellow fitted dress with white off-the-shoulder sleeves.webp', '/content/women_fashion/women fashion\\off-the-shoulder black ball gown.jpg', '/content/women_fashion/women fashion\\one-shoulder, black, sequined dress with fringe detailing at the hem.jpg', '/content/women_fashion/women fashion\\one-shoulder, fitted dress that features sequin embellishments and sheer panels.jpg', '/content/women_fashion/women fashion\\pink, knee-length dress that sparkles.jpg', '/content/women_fashion/women fashion\\red dress adorned with an intricate white pattern.jpg', '/content/women_fashion/women fashion\\red dress with a pattern of small white flowers.jpg', '/content/women_fashion/women fashion\\red velvet sheath midi dress.jpg', '/content/women_fashion/women fashion\\royal blue off-shoulder dress.jpg', '/content/women_fashion/women fashion\\salwar kameez in a vibrant red color with detailed embroidery.jpg', '/content/women_fashion/women fashion\\shiny purple sequined dress with a one-shoulder design.jpg', '/content/women_fashion/women fashion\\shiny, silver, wrap-style dress with long sleeves.jpg', '/content/women_fashion/women fashion\\short, blue floral dress with long puffed sleeves.jpg', '/content/women_fashion/women fashion\\short, white, sleeveless dress with thin straps.jpg', '/content/women_fashion/women fashion\\sleeveless dress adorned with pink sequins.jpg', '/content/women_fashion/women fashion\\sleeveless dress with a high neckline.jpeg', '/content/women_fashion/women fashion\\Sleeveless-Ruched-Tight-Fashion-Sexy-Bodycon-Party-Women-Club-Dresses.webp', '/content/women_fashion/women fashion\\solid dark blue top and a floral pattern on the bottom.jpg', '/content/women_fashion/women fashion\\sophisticated black dress with an asymmetrical design with one sleeve and a strapless neckline on the opposite side.jpg', '/content/women_fashion/women fashion\\sparkling black dress.jpg', '/content/women_fashion/women fashion\\sparkling blue dress.jpeg', '/content/women_fashion/women fashion\\sparkling white dress with long sleeves.jpg', '/content/women_fashion/women fashion\\sparkling, fitted dress 2.jpeg', '/content/women_fashion/women fashion\\sparkling, fitted dress with long sleeves.jpg', '/content/women_fashion/women fashion\\sparkling, fitted dress.jpeg', '/content/women_fashion/women fashion\\sparkling, sequined dress.jpg', '/content/women_fashion/women fashion\\strapless red midi dress with a mermaid silhouette.jpg', '/content/women_fashion/women fashion\\strapless, sequined dress that sparkles with multiple colors.jpg', '/content/women_fashion/women fashion\\tight-fitting, off-the-shoulder white dress 2.jpg', '/content/women_fashion/women fashion\\tight-fitting, off-the-shoulder white dress.webp', '/content/women_fashion/women fashion\\vibrant blue and a neutral tone adorned with colorful floral patterns.jpg', '/content/women_fashion/women fashion\\well-fitted beige suit.jpg', '/content/women_fashion/women fashion\\white dress adorned with colorful floral patterns.jpg', '/content/women_fashion/women fashion\\white knee-length dress with a fitted design and long sleeves.webp', '/content/women_fashion/women fashion\\white, intricately detailed top and a flowing dark blue skirt.jpg', '/content/women_fashion/women fashion\\Women-off-the-shoulder-sexy-embroidery-fashion-party-dress-1.png', '/content/women_fashion/women fashion\\yellow, intricately designed Anarkali suit.jpeg']

In [14]:
```
pip install tensorflow
```

```
Collecting tensorflow
  Obtaining dependency information for tensorflow from https://files.pythonhosted.org/packages/93/21/9b035a4f823d6aee2917c75415be9a95861ff3d73a0a65e48edbf210cec1/tensorflow-2.15.0-cp311-cp311-win_amd64.whl.metadata (https://files.pythonhosted.org/packages/93/21/9b035a4f823d6aee2917c75415be9a95861ff3d73a0a65e48edbf210cec1/tensorflow-2.15.0-cp311-cp311-win_amd64.whl.metadata)
  Downloading tensorflow-2.15.0-cp311-cp311-win_amd64.whl.metadata (3.6 kB)
Collecting tensorflow-intel==2.15.0 (from tensorflow)
  Obtaining dependency information for tensorflow-intel==2.15.0 from https://files.pythonhosted.org/packages/4c/48/1a5a15517f18eaa4ff8d598b1c000300b20c1bb0e624539d702117a0c369/tensorflow_intel-2.15.0-cp311-cp311-win_amd64.whl.metadata (https://files.pythonhosted.org/packages/4c/48/1a5a15517f18eaa4ff8d598b1c000300b20c1bb0e624539d702117a0c369/tensorflow_intel-2.15.0-cp311-cp311-win_amd64.whl.metadata)
  Downloading tensorflow_intel-2.15.0-cp311-cp311-win_amd64.whl.metadata (5.1 kB)
Collecting absl-py>=1.0.0 (from tensorflow-intel==2.15.0->tensorflow)
  Obtaining dependency information for absl-py>=1.0.0 from https://files.pythonhosted.org/packages/a2/ad/e0d3c824784ff121c03cc031f944bc7e139a8f1870ffd2845cc2dd76f6c4/absl_py-2.1.0-py3-none-any.whl.metadata (https://files.pythonhosted.org/packages/a2/ad/e0d3c824784ff121c03cc031f944bc7e139a8f1870ffd2845cc2dd76f6c4/absl_py-2.1.0-py3-none-any.whl.metadata)
  Downloading absl_py-2.1.0-py3-none-any.whl.metadata (2.3 kB)
Collecting astunparse>=1.6.0 (from tensorflow-intel==2.15.0->tensorflow)
```

In the above code, the glob module is used to generate a list of file paths for images stored in the directory. The glob.glob function searches for files that match a specified pattern, in this case, *.*, which matches all files within the directory. The list comprehension then filters these files to include only those with specific image file extensions (.jpg, .png, .jpeg, .webp).

It ensures that image_paths_list contains paths to only the image files, excluding any other file types that might be present in the directory.

Now, we will extract features from all the fashion images:

In [17]:

```
from tensorflow.keras.preprocessing import image
from tensorflow.keras.applications.vgg16 import VGG16, preprocess_input
from tensorflow.keras.applications.vgg16 import preprocess_input
from tensorflow.keras.models import Model
import numpy as np

base_model = VGG16(weights='imagenet', include_top=False)
model = Model(inputs=base_model.input, outputs=base_model.output)

def preprocess_image(img_path):
    img = image.load_img(img_path, target_size=(224, 224))
    img_array = image.img_to_array(img)
    img_array_expanded = np.expand_dims(img_array, axis=0)
    return preprocess_input(img_array_expanded)

def extract_features(model, preprocessed_img):
    features = model.predict(preprocessed_img)
    flattened_features = features.flatten()
    normalized_features = flattened_features / np.linalg.norm(flattened_features)
    return normalized_features

all_features = []
all_image_names = []

for img_path in image_paths_list:
    preprocessed_img = preprocess_image(img_path)
    features = extract_features(model, preprocessed_img)
    all_features.append(features)
    all_image_names.append(os.path.basename(img_path))
```

```
1/1 [==============================] - 0s 175ms/step
1/1 [==============================] - 0s 108ms/step
1/1 [==============================] - 0s 119ms/step
1/1 [==============================] - 0s 115ms/step
1/1 [==============================] - 0s 119ms/step
1/1 [==============================] - 0s 134ms/step
1/1 [==============================] - 0s 133ms/step
1/1 [==============================] - 0s 133ms/step
1/1 [==============================] - 0s 118ms/step
1/1 [==============================] - 0s 114ms/step
1/1 [==============================] - 0s 107ms/step
1/1 [==============================] - 0s 107ms/step
1/1 [==============================] - 0s 114ms/step
1/1 [==============================] - 0s 120ms/step
1/1 [==============================] - 0s 131ms/step
1/1 [==============================] - 0s 152ms/step
1/1 [==============================] - 0s 126ms/step
1/1 [==============================] - 0s 130ms/step
1/1 [==============================] - 0s 128ms/step
1/1 [==============================] - 0s 136ms/step
1/1 [==============================] - 0s 126ms/step
1/1 [==============================] - 0s 125ms/step
1/1 [==============================] - 0s 127ms/step
1/1 [==============================] - 0s 126ms/step
1/1 [==============================] - 0s 128ms/step
1/1 [==============================] - 0s 149ms/step
1/1 [==============================] - 0s 167ms/step
1/1 [==============================] - 0s 131ms/step
1/1 [==============================] - 0s 136ms/step
1/1 [==============================] - 0s 144ms/step
1/1 [==============================] - 0s 135ms/step
1/1 [==============================] - 0s 112ms/step
1/1 [==============================] - 0s 128ms/step
1/1 [==============================] - 0s 138ms/step
1/1 [==============================] - 0s 133ms/step
1/1 [==============================] - 0s 174ms/step
1/1 [==============================] - 0s 110ms/step
1/1 [==============================] - 0s 111ms/step
1/1 [==============================] - 0s 119ms/step
1/1 [==============================] - 0s 115ms/step
1/1 [==============================] - 0s 121ms/step
1/1 [==============================] - 0s 117ms/step
1/1 [==============================] - 0s 123ms/step
1/1 [==============================] - 0s 123ms/step
1/1 [==============================] - 0s 139ms/step
1/1 [==============================] - 0s 125ms/step
1/1 [==============================] - 0s 124ms/step
1/1 [==============================] - 0s 127ms/step
1/1 [==============================] - 0s 133ms/step
1/1 [==============================] - 0s 137ms/step
1/1 [==============================] - 0s 134ms/step
1/1 [==============================] - 0s 148ms/step
1/1 [==============================] - 0s 161ms/step
1/1 [==============================] - 0s 143ms/step
1/1 [==============================] - 0s 125ms/step
1/1 [==============================] - 0s 124ms/step
1/1 [==============================] - 0s 134ms/step
1/1 [==============================] - 0s 120ms/step
1/1 [==============================] - 0s 110ms/step
1/1 [==============================] - 0s 134ms/step
1/1 [==============================] - 0s 121ms/step
1/1 [==============================] - 0s 114ms/step
1/1 [==============================] - 0s 116ms/step
1/1 [==============================] - 0s 135ms/step
1/1 [==============================] - 0s 136ms/step
1/1 [==============================] - 0s 128ms/step
1/1 [==============================] - 0s 114ms/step
1/1 [==============================] - 0s 135ms/step
1/1 [==============================] - 0s 130ms/step
1/1 [==============================] - 0s 122ms/step
1/1 [==============================] - 0s 120ms/step
1/1 [==============================] - 0s 136ms/step
1/1 [==============================] - 0s 114ms/step
1/1 [==============================] - 0s 111ms/step
1/1 [==============================] - 0s 134ms/step
1/1 [==============================] - 0s 135ms/step
1/1 [==============================] - 0s 107ms/step
1/1 [==============================] - 0s 134ms/step
1/1 [==============================] - 0s 107ms/step
1/1 [==============================] - 0s 114ms/step
1/1 [==============================] - 0s 117ms/step
1/1 [==============================] - 0s 109ms/step
1/1 [==============================] - 0s 114ms/step
```

```
1/1 [==============================] - 0s 114ms/step
1/1 [==============================] - 0s 110ms/step
1/1 [==============================] - 0s 106ms/step
1/1 [==============================] - 0s 112ms/step
1/1 [==============================] - 0s 125ms/step
1/1 [==============================] - 0s 113ms/step
1/1 [==============================] - 0s 120ms/step
1/1 [==============================] - 0s 112ms/step
1/1 [==============================] - 0s 113ms/step
1/1 [==============================] - 0s 113ms/step
1/1 [==============================] - 0s 120ms/step
1/1 [==============================] - 0s 106ms/step
1/1 [==============================] - 0s 109ms/step
```

In the above code, a feature extraction process is implemented using the VGG16 model, a popular convolutional neural network pre-trained on the ImageNet dataset, to extract visual features from images stored in image_paths_list.

Initially, the VGG16 model is loaded without its top classification layer (include_top=False), making it suitable for feature extraction rather than classification. Each image path from image_paths_list is processed through a series of steps: the image is loaded and resized to 224×224 pixels to match the VGG16 input size requirements, converted to a NumPy array, and preprocessed to fit the model's expected input format.

The preprocessed images are then fed into the VGG16 model to extract features, which are subsequently flattened and normalized to create a consistent feature vector for each image. These feature vectors (all_features) and their corresponding image filenames (all_image_names) are stored, providing a structured dataset for the next steps in building a fashion recommendation system using image features.

Now, I'll write a function to recommend fashion images based on image features:

In [22]:

```python
from scipy.spatial.distance import cosine

def recommend_fashion_items_cnn(input_image_path, all_features, all_image_names, model, top_n=5):
    # pre-process the input image and extract features
    preprocessed_img = preprocess_image(input_image_path)
    input_features = extract_features(model, preprocessed_img)

    # calculate similarities and find the top N similar images
    similarities = [1 - cosine(input_features, other_feature) for other_feature in all_features]
    similar_indices = np.argsort(similarities)[-top_n:]

    # filter out the input image index from similar_indices
    similar_indices = [idx for idx in similar_indices if idx != all_image_names.index(input_image_path)]

    # display the input image
    plt.figure(figsize=(15, 10))
    plt.subplot(1, top_n + 1, 1)
    plt.imshow(Image.open(input_image_path))
    plt.title("Input Image")
    plt.axis('off')

    # display similar images
    for i, idx in enumerate(similar_indices[:top_n], start=1):
        image_path = os.path.join('/content/women_fashion/women fashion', all_image_names[idx])
        plt.subplot(1, top_n + 1, i + 1)
        plt.imshow(Image.open(image_path))
        plt.title(f"Recommendation {i}")
        plt.axis('off')

    plt.tight_layout()
    plt.show()
```

In the above code, we defined a function recommend_fashion_items_cnn, which recommends fashion items similar to a given input image using deep learning-based feature extraction. It utilizes the VGG16 model to extract high-dimensional feature vectors from images, capturing their visual essence.

For a specified input image, the function preprocesses the image, extracts its features, and calculates the cosine similarity between this feature vector and those of other images in the dataset (all_features). It ranks these images based on similarity and selects the top N most similar images to recommend, explicitly excluding the input image from being recommended to itself by filtering out its index from the list of similar indices.

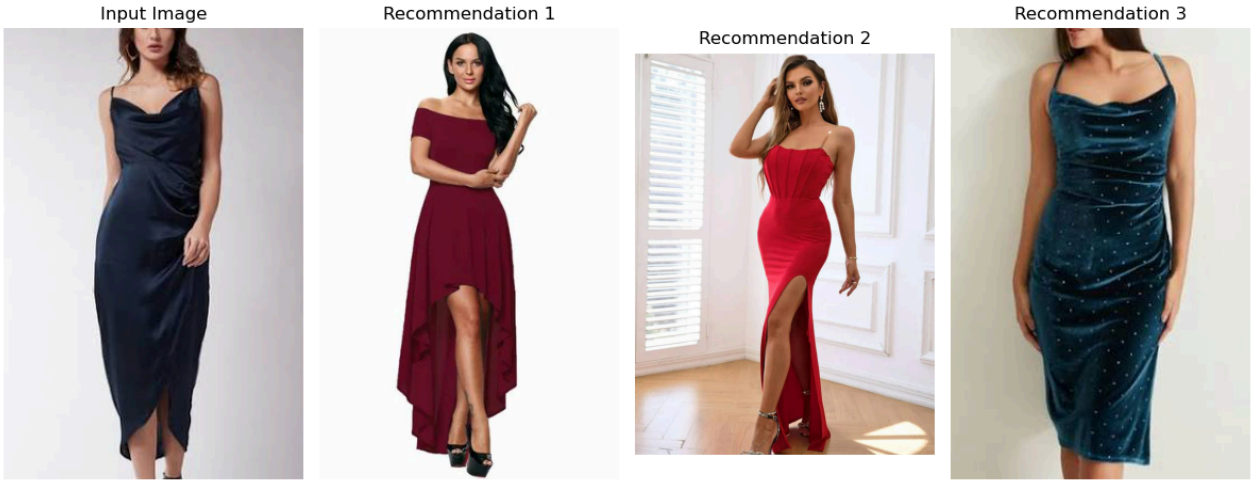
In the end, the function will visualize the input image and its recommendations by displaying them.

Now, here's how we can use this function to recommend images based on a similar fashion in the input image:

In [24]:

```
def recommend_fashion_items_cnn(input_image_path, all_features, all_image_names, model, top_n=5):
    # pre-process the input image and extract features
    preprocessed_img = preprocess_image(input_image_path)
    input_features = extract_features(model, preprocessed_img)

    # calculate similarities and find the top N similar images
    similarities = [1 - cosine(input_features, other_feature) for other_feature in all_features]
    similar_indices = np.argsort(similarities)[-top_n:]

    # filter out the input image index from similar_indices
    input_filename = os.path.basename(input_image_path)
    try:
        input_index = all_image_names.index(input_filename)
        similar_indices = [idx for idx in similar_indices if idx != input_index]
    except ValueError:
        print(f"Input image {input_filename} not found in the list.")

    # display the input image
    plt.figure(figsize=(15, 10))
    plt.subplot(1, top_n + 1, 1)
    plt.imshow(Image.open(input_image_path))
    plt.title("Input Image")
    plt.axis('off')

    # display similar images
    for i, idx in enumerate(similar_indices[:top_n], start=1):
        image_path = os.path.join('/content/women_fashion/women fashion', all_image_names[idx])
        plt.subplot(1, top_n + 1, i + 1)
        plt.imshow(Image.open(image_path))
        plt.title(f"Recommendation {i}")
        plt.axis('off')

    plt.tight_layout()
    plt.show()

# Example usage:
input_image_path = '/content/women_fashion/women fashion/dark, elegant, sleeveless dress that reaches down to about mid-c
recommend_fashion_items_cnn(input_image_path, all_features, all_image_names, model, top_n=4)
```

```
1/1 [==============================] - 0s 111ms/step
```



SUMMARY

So, this is how you can build a Fashion Recommendation System using Image Features using the Python programming language. A Fashion Recommendation System using Image Features leverages computer vision and machine learning techniques to analyze fashion items' visual aspects (like colour, texture, and style) and recommend similar or complementary products to users.