



Fashion explorer project

-Jithin Kumar



Main agenda *of the project*

This project is to mainly showcase my knowledge that I have gained throughout the bootcamp all together in a single project.



About the App

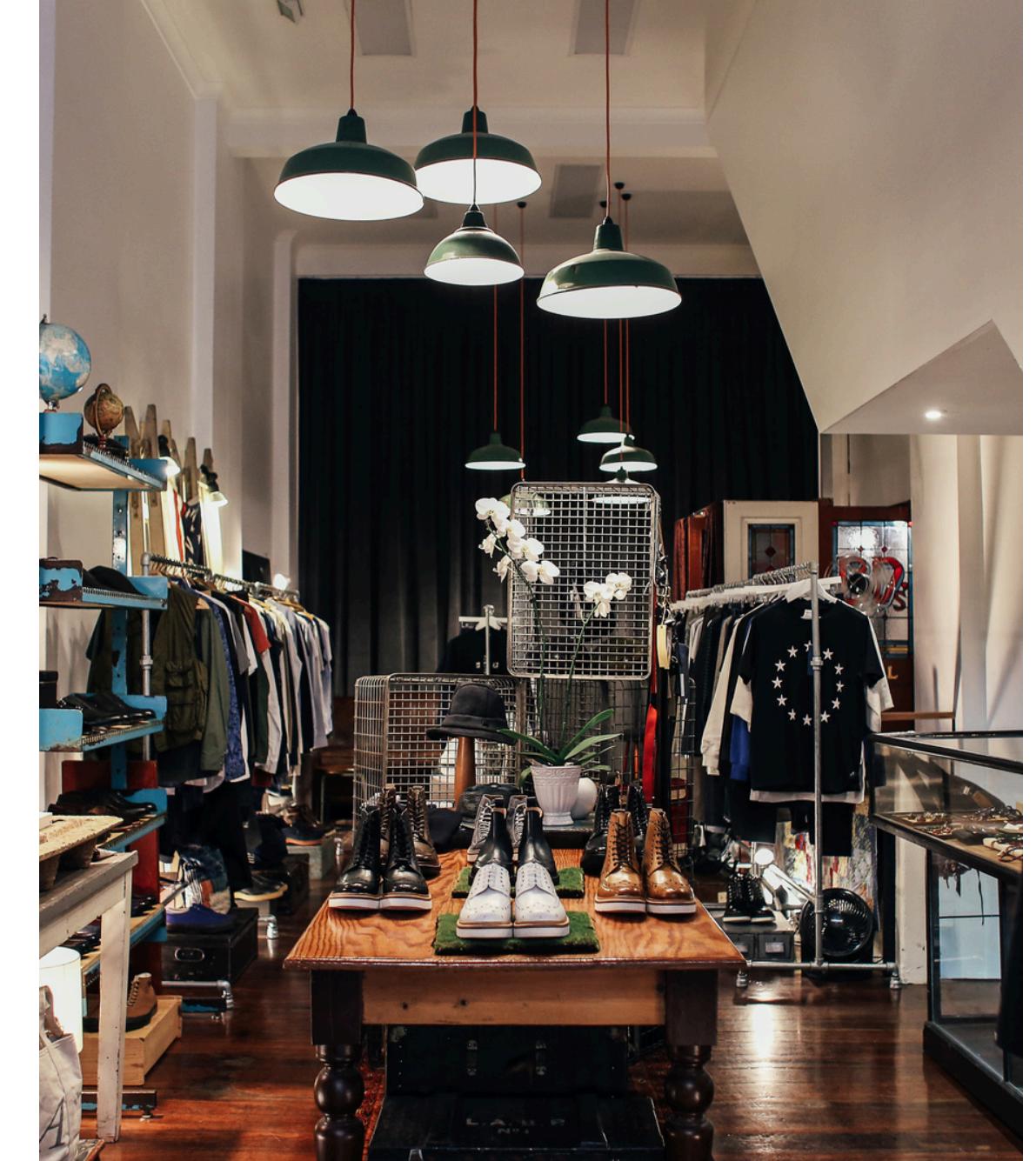
- Upload fashion-related image → Detects items (tops, bottoms, footwear, bags, accessories, full dress, hats, outerwear).
- Enter a text prompt (e.g., “party outfit” or “casual office look”) → Generates matching style ideas.
- Combines:
 1. **Computer Vision** → Image-based outfit detection.
 2. **Text Processing (LLM)** → Natural language styling advice.
- Simple, user-friendly design → Works as a digital fashion assistant.





Why this project

Fashion choices are often confusing — mixing and matching outfits can be time-consuming, and keeping up with trends isn't easy. This app was built to simplify style decisions by showing how AI can act as a personal stylist.





Computer Vision (YOLO + Color Extraction)

- **YOLO Object Detection** → Used to detect fashion items (top, bottom, shoes, accessories) from uploaded or live images.
- **Color Extraction** → Extracts dominant HEX colors for each detected item, so styling advice can include color palettes (e.g., "#b4c6d2 light blue shirt").
- **ML Rules based Suggestions** → Simple predefined logic for pairing detected items (e.g., "If jeans → suggest sneakers").



Generative AI (Stylist Brain)

- **LLM (OpenAI GPT)** → Converts detected items + colors (image) or free-text input (text page) into natural language fashion advice.
- **Expanded Keywords** → Uses AI to turn vague input ("summer outfit") into richer product keywords ("linen shirt, denim shorts, strappy sandals").
- **Personalized Advice** → Generates contextual recommendations (season, gender, event type).

Technology Behind It:



Product Integration (ASOS API + Fallback)

- **ASOS API** → Fetches real products (image, price, purchase link).
- **Fallback Dataset** → Ensures demo works even if API fails.
- **Product Display** → Styled cards showing product image, price, and link.



Frontend (Streamlit App)

- Multi-page design (Home, Image Detection, Text Stylist).
- **UI/UX Enhancements:**
 - 1.CSS animations (fade, pop-in).
 - 2.Color swatches for extracted HEX codes.
 - 3.Expanders for clean layout.

Session State Management

- Prevents re-running detections or AI calls unnecessarily.
- Stores user's selected keyword so dropdown doesn't reset.
- Keeps product results visible even after interaction.

M O O D B O A R D





Steps followed

Computer Vision (YOLO)

- Dataset (Deep Fashion Images) - Annotation
 - Train on YOLOv8- Semi Automation
 - Predict on new images - Check predictions
 - Re annotate corrections - Retrain - Check on Live and Video

Rule-based Suggestions

- Made 1 2 and 3 combination rules (if else conditions) to provide ML based suggestions or Fashion advices.

Generative AI (Stylist Brain)

- Get Labels and Colors from the predictions on images and Using GPT 4o mini give prompts to get Ai based suggestion - For both Male and Female.
- It also Expands the keywords to make much more Precise suggestions.

Product Integration (ASOS API + Fallback)

- Get the Keywords from suggestions - Using ASOS API fetch real products - If no Products use fallbacks for products suggestions.



Streamlit App

Set up multi-page layout

Created separate files (home.py, image.py, text.py) to organize pages

Built Image Detection Page

Added file uploader for images. - Displayed detected items + extracted colors with swatches.

Built Text Stylist Page

Allowed free-text input for fashion queries. - Connected AI for stylist advice + product suggestions.

Improved User Experience

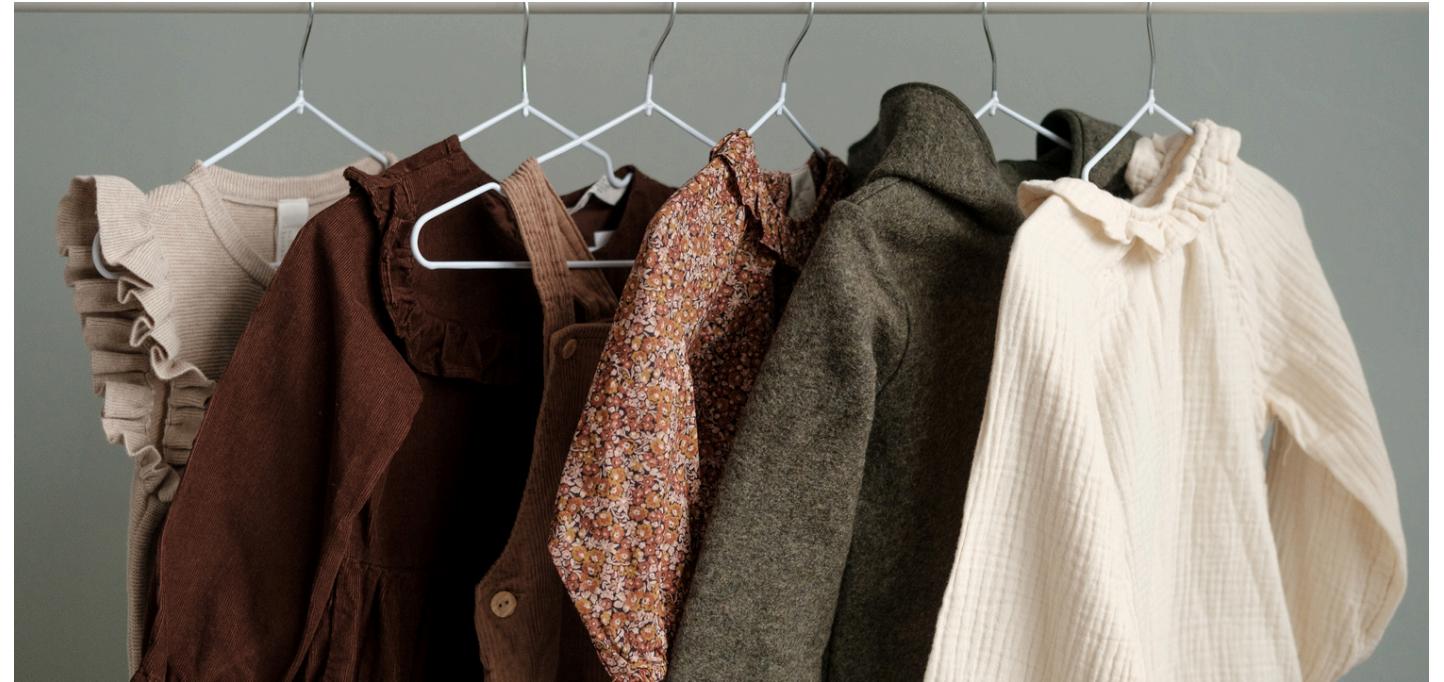
Added CSS animations (fade, pop-in).- Used expanders to keep layout clean. - Styled product cards with images, prices, and links.





Challenges faced

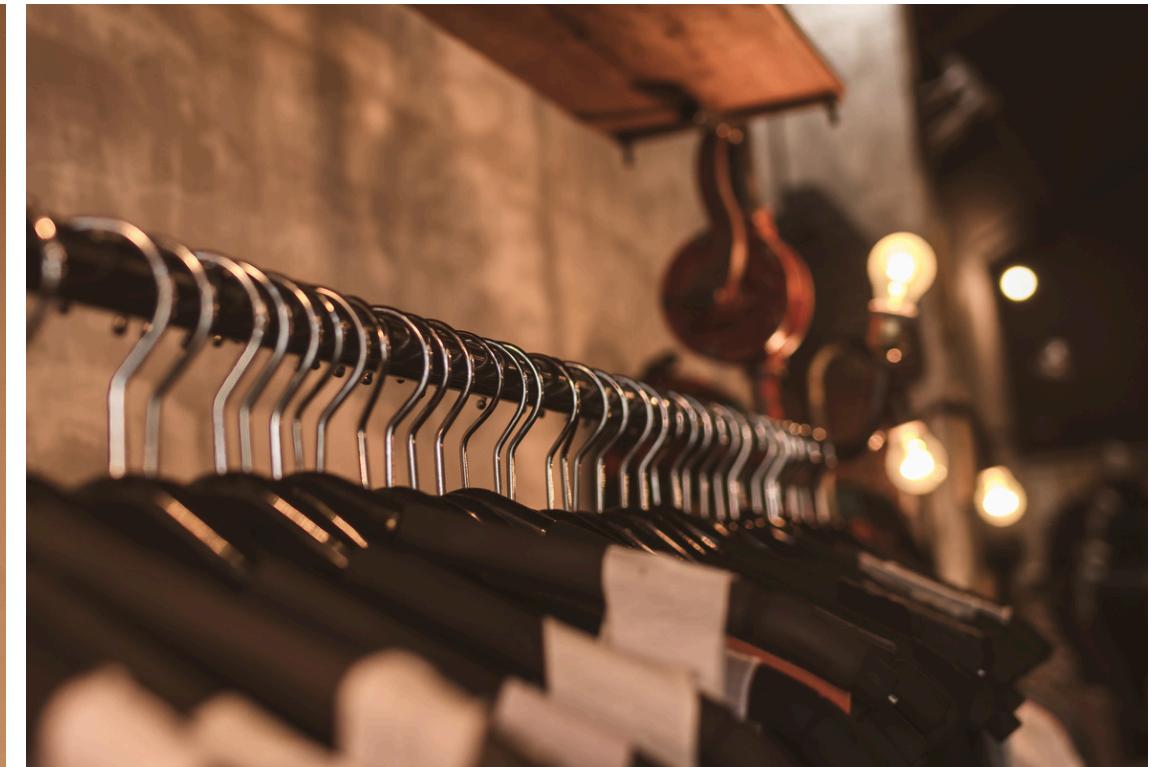
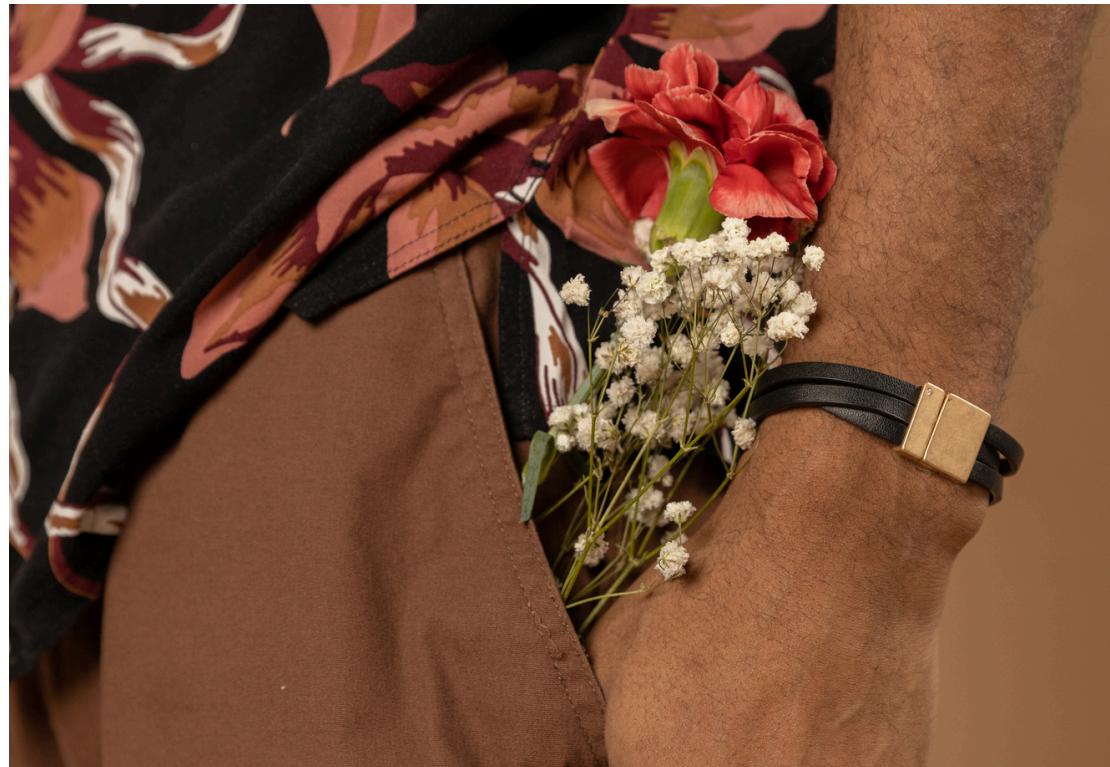
- Lots of Dataset to annotate and train
- Lots of time to train for even less images
- Training on Google colab.
- Writing logics entirely from scratch.
- Using free LLM models for AI advice.
- Frequent refreshes in streamlit if any buttons are pressed
- Web scrapping , Amazon API, eBay API for getting products links.
- Model and ML Logics slow loading issue on streamlit.
- Font changing issue in streamlit.





Future updates

- Extra Features & Planned Extensions
- Live Detection Page → Real-time object detection with webcam.
- Outfit Builder (Mix & Match) → Drag & drop tops, bottoms, shoes into a collage.
- Blog/News Page → Curated links to external fashion blogs & style tips.





Conclusion

- JK's Trendy Explorer demonstrates how Machine Learning and Generative AI can enhance the fashion industry by simplifying outfit selection and personal styling.
- The project showcases a blend of computer vision, NLP, and real-time product integration through an interactive Streamlit app.
- Despite challenges like dataset preparation and API limitations, the system provides a strong foundation for future extensions like live detection, outfit builder, and fashion blogs.
- This project reflects my technical skills, problem-solving ability, and creativity in applying AI to real-world use cases.



Thank You So Much

-Jithin Kumar