Project Synopsis

AI-Driven Virtual Fashion & E-Commerce Platform (with Virtual Try-On, Recommendations, Smart Search, Fit Prediction, and Analytics)

Project Members

S.No.	Name	University Roll No.	Signature
1.	Anurag Khanduri	2218436	
2.	Mansi Nainwal	2219058	
3.	Pratyaksh Bhandari	2219286	
4.	Manya Rajput	2261356	

Under the Guidance of:

Department of Computer Science & Engineering
Graphic Era Hill University

Abstract

Online fashion shopping suffers from uncertainty about **fit, style, and appearance**, causing high returns and low customer confidence. We propose a **multi-module AI platform** that combines: **(1) A GAN-based Virtual Try-On engine, (2) a Fashion Recommendation system, (3) Smart Search (text + image), (4) Body Measurement & Fit Prediction, and (5) an Admin Analytics Dashboard.** The system delivers an end-to-end experience: users upload a photo and select garments to visualize try-on, discover relevant items through recommendations and search, receive data-driven size suggestions, and admins track engagement and demand. The project blends **Computer Vision, Deep Learning, NLP, Recommender Systems, and Full-Stack engineering**.

Problem Statement

- Users cannot visualize garments on themselves before buying → uncertainty and high return rates.
- Size selection is error-prone → poor fit, dissatisfaction.
- Catalogs are huge → discovering the right item is hard without personalization.
- Sellers lack insights into try-on engagement and product demand.

Objectives

1. Build a **2D Virtual Try-On** module that realistically overlays garments on a user's photo.
2. Develop an **AI Recommendation Engine** (hybrid: content + collaborative + style attributes).
3. Implement **Smart Search** that supports **text queries** (NLP parsing) and **image queries** (visual similarity via embeddings).
4. Create **Body Measurement & Fit Prediction** to recommend sizes (S/M/L/XL).
5. Provide an **Admin Analytics Dashboard** with product trends, try-on stats, and demand indicators.
6. Deliver a cohesive web application with modular APIs and a clean UI.

Scope & Deliverables

- **User-facing web app** (React/Next.js or Streamlit) with:
    - Photo & garment upload
    - Try-on preview
    - Recommended items
    - Smart search (text/image)
    - Size suggestion

- **Backend (Flask/FastAPI)** exposing model inference APIs.
- **Databases** for users, products, embeddings, logs, analytics.

- **Admin dashboard** with charts (traffic, try-on frequency, conversions proxy).
- **Documentation**: report, API specs, user manual, test results.

[ML Services]

- Virtual Try-On (CP-VTON/HR-VITON, PyTorch)

- Recommender (Content+CF+Attributes)

- Smart Search (CLIP text/image embeddings)

- Fit Prediction (Pose + Regression)

Modules (with Methods & Tech)

1) Virtual Try-On (Core)

- **Input:** person photo (frontal), garment image (product photo).
- **Process:**
  - Pose estimation (MediaPipe/OpenPose).
  - Person/cloth segmentation (U^2Net/DeepLabV3).
  - Cloth warping + composition via **CP-VTON / HR-VITON (PyTorch)**.
- **Output:** synthesized image with person “wearing” the garment.
- **Datasets:** VITON, DeepFashion (paired images where possible).
- **Metrics:** SSIM/LPIPS + user study (MOS for realism).

2) Fashion Recommendation Engine

- **Hybrid approach:**
  - **Content-based:** garment embeddings (CLIP/ResNet) + attributes (color, pattern, style).
  - **Collaborative filtering (CF):** simple matrix factorization or item-item similarity using interactions (views/try-ons).
  - **Rule layer:** balance novelty/diversity; filter by availability/size.

- **Personalization signal:** click/try-on history, likes, dwell time (project-scale proxies).
- **Metrics:** Precision@K, nDCG, CTR uplift in A/B against popularity baseline.

3) Smart Search (Text + Image)

- **Text:** Parse queries like “red floral dress under ₹2000” → extract color/style/price filters; rank via **BM25/ElasticSearch** + embedding relevance.
- **Image:** Upload reference outfit → compute **CLIP** image embedding; retrieve nearest items from a **vector index (FAISS/PG-vector)**.
- **Metrics:** Recall@K / mAP on a labeled subset; qualitative gallery comparisons.

4) Body Measurement & Fit Prediction

- **Pose Estimation:** MediaPipe/OpenPose keypoints; optional height normalization.
- **Features:** torso ratios, shoulder width proxies, waist cues (project-scale approximations).
- **Model:** regression/MLP to predict chest/waist; map to size charts per brand.
- **Calibration:** user-provided height as a scaling anchor (simple, effective).
- **Metrics:** MAE for measurements; size accuracy (% within correct or ±1 size).

5) Admin Analytics Dashboard

- **KPIs:** try-on counts, rec click-through, search usage, top-viewed items, size distribution, simple demand forecast (ARIMA/Prophet or seasonal naive).
- **UI:** Recharts/Chart.js graphs; downloadable CSV reports.
- **Value:** evidence for decisions (inventory, merchandising, UX tweaks).

Data Flow

1. **Catalog ingestion**: images → attributes → embeddings → DB & vector index.
2. **User session**: upload photo → try-on API → synthesized result saved to object store.
3. **Interactions** (views/likes/try-ons) logged → feed recommender & analytics.
4. **Search**: text→NLP parse or image→CLIP embedding → retrieve → rank → display.
5. **Fit**: pose→features→size → shown alongside product.

Tech Stack

- **ML/CV/DL:** PyTorch, OpenCV, MediaPipe/OpenPose, U^2Net/DeepLabV3, CP-VTON/HR-VITON, CLIP (HuggingFace).
- **Backend:** FastAPI/Flask, Celery/RQ (optional async), Gunicorn/Uvicorn.
- **Frontend:** React/Next.js or Streamlit (MVP).
- **Databases:** Postgres or MongoDB; **Vector DB** (FAISS / PG-vector / Elastic kNN).
- **Deployment:** Docker, NVIDIA CUDA container, AWS/GCP/Render; Nginx; object storage (S3-compatible).

API Sketch

- POST /tryon → body:{user_image, cloth_image|id} → returns synthesized image URL.
- GET /recommend?user_id=... → returns top-N product IDs.
- POST /search/text → body:{query} → returns ranked products.
- POST /search/image → body:{image} → returns visually similar products.
- POST /fit → body:{user_image, height(optional)} → returns predicted size.
- GET /admin/kpis?range=last_30d → charts/data for dashboard.

Evaluation Plan

- **Try-On realism:** SSIM/LPIPS + human ratings (1–5) on 50 images.
- **Recommender:** Precision@10 / nDCG on interaction logs (or simulated labels).
- **Search:** Recall@K / mAP on a tagged subset; qualitative top-K galleries.
- **Fit:** MAE (cm) vs. measured ground truth (small volunteer set); % correct size.
- **Ablations:** With/without attributes, different embedding backbones, etc.

Project Timeline (11–12 Months)

- **M1–M2:** Literature review; dataset setup; baseline CLIP & segmentation; system design.
- **M3–M4:** Implement **Virtual Try-On** baseline (CP-VTON/HR-VITON) + preprocessing.
- **M5–M6: Smart Search** (CLIP embeddings, vector index) + text parsing.
- **M7–M8: Recommendation Engine** (content + simple CF) + feedback logging.
- **M9: Fit Prediction** (pose features → size model).
- **M10: Frontend & Integration**; consistent APIs; caching; UX polish.
- **M11: Analytics Dashboard**; evaluation, user study, ablations.
- **M12:** Hardening, documentation, demo video, final report/PPT.

Team Roles

- **Member 1 (CV/DL):** Virtual Try-On (preprocess, training, inference).
- **Member 2 (IR/NLP/RecSys):** Smart Search + Recommendation Engine.
- **Member 3 (CV/ML):** Pose & Fit Prediction + data collection protocol.
- **Member 4 (Full-Stack/DevOps):** Backend APIs, DBs, Frontend, Analytics, Deployment.

Risks & Mitigations

- **Input variability (poses/lighting):** constrain MVP to frontal, good-light photos; simple background guidance.
- **Hardware limits:** use Colab/Kaggle; quantized inference; caching of embeddings and try-on results.
- **Data/IP concerns:** prefer open datasets; avoid scraping restricted/editorial content; clear consent for user photos.
- **Latency:** precompute product embeddings; async jobs for heavy inference; thumbnails first, HD on demand.

Expected Outcomes

- A **working multi-module platform** demonstrating end-to-end AI for fashion e-commerce.
- Clear **quantitative and qualitative** evaluation across modules.
- Clean web UI + deployable Docker images + complete documentation.
- Strong basis for a **paper/poster** or extended thesis.

Signature
Mentor/Guide