# **Virtual Try-On**

## **Project Title:**

**Virtual Try-On**

## **Team Name:**

Nitin and team

## **Team Members:**

* G Sai Nitin
* M Thirumalesh
* G Abhigna
* V Shiva Siddhardha

## **Phase-1: Brainstorming & Ideation**

### **Objective:**

Develop a deep learning-based Virtual Try-On (VTON) system that enables users to virtually try on clothing from an image.

### **Key Points:**

1. **Problem Statement:**
   * Online shoppers struggle to visualize how clothing will look on them before purchasing.
   * Existing try-on methods lack realism and fail with different poses and body shapes.
2. **Proposed Solution:**
   * A Virtual Try-On System using deep learning (U-Net, VITON-HD, or TryOnGAN).
   * Given an image of a person and a clothing item, the system generates a realistic image of the person wearing the new outfit.
   * A Streamlit-based UI that allows users to upload images and view results.
3. **Target Users:**
   * **E-commerce platforms**
   * Fashion designers
   * Virtual fitting rooms
4. **Expected Outcome:**
   * A working AI-powered Virtual Try-On system with an interactive frontend for users to upload images and get realistic try-on results.

## **Phase-2: Requirement Analysis**

### **Objective:**

Define technical and functional requirements.

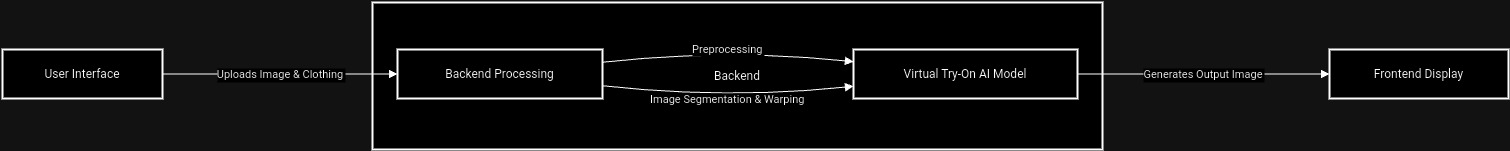
### **Key Points:**

1. **Technical Requirements:**
   * Programming Language: Python
   * Deep Learning Framework: PyTorch
   * Model Architecture: U-Net, VITON-HD
   * Frontend: Streamlit
   * Dataset: VITON-HD dataset
   * GPU: Google Colab with batch processing
2. **Functional Requirements:**
   * User uploads their image and clothing image via Streamlit UI.
   * Preprocessing: Aligning and warping the clothing to fit the person.
   * Inference: Using the trained deep learning model to generate a realistic output.
   * Post-processing: Refining the output image to improve realism.
   * Displaying Output in UI: The generated try-on image is shown on the Streamlit page.
3. **Constraints & Challenges:**
   * Training requires high computational power.
   * Ensuring clothing warps correctly on different body types.
   * Preventing artifacts or distortions in the output.
   * Ensuring seamless integration of the frontend with the deep learning model.

## **Phase-3: Project Design**

### **Objective:**

Develop the architecture and user flow of the Virtual Try-On system.



### **Key Points:**

1. **System Architecture:**
   * System Architecture
   * 1. User Interface (UI) with Streamlit

User uploads their photo and clothing image.

Images are sent to the deep learning model.

* + 2. Image Preprocessing

Resize, normalize, and align images.

* + 3. Deep Learning Model

Warps the clothing image to fit the person.

Generates the final try-on result.

* + 4. Output Display

The system returns and displays the try-on image on Streamlit.

1. **User Flow:**
   * Step 1: Upload person and clothing image via UI.
   * Step 2: Preprocessing (alignment, normalization).
   * Step 3: Model generates the try-on image.
   * Step 4: Output displayed in the UI.
2. **UI/UX Considerations:**
   * **Simple drag-and-drop interface.**
   * Real-time preview of uploaded images.
   * Download option for the final try-on image.

## 

## **Phase-4: Project Planning (Agile Methodologies)**

### **Objective:**

Break down development tasks for efficient completion.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Sprint** | **Task** | **Priority** | **Duration** | **Deadline** | **Assigned To** | **Dependencies** | **Expected Outcome** |
| Sprint 1 | Dataset Collection & Preprocessing | 🔴 High | 6 hours (Day 1) | End of Day 1 | Sai Nitin | VITON-HD Dataset | Preprocessed images ready |
| Sprint 1 | Model Selection & Setup | 🟡 Medium | 3 hours (Day 1) | End of Day 1 | Abhigna | U-Net / VITON-HD | Model initialized |
| Sprint 2 | Model Training & Optimization | 🔴 High | 6 hours (Day 2) | Mid-Day 2 | Thirumalesh | Preprocessed dataset | Model trained |
| Sprint 2 | Error Handling & Debugging | 🔴 High | 1.5 hours (Day 2) | Mid-Day 2 | Shiva | Training logs | Improved accuracy |
| Sprint 3 | UI Development (Streamlit) | 🟡 Medium | 1.5 hours (Day 2) | Mid-Day 2 | Shiva & nitin | Model output available | User-friendly UI |
| Sprint 3 | Final Presentation & Deployment | 🟢 Low | 1 hour (Day 2) | End of Day 2 | Entire Team | Completed UI & Model | Working prototype |

### 

### **Sprint Planning with Priorities**

**Sprint 1 – Setup & Integration (Day 1)**

(🔴 High Priority) Set up the environment & install dependencies.

(🔴 High Priority) Load and preprocess the VITON-HD dataset.

(🔴 High Priority) Select and configure the deep learning model (U-Net / VITON-HD).

**Sprint 2 – Model Training & Debugging (Day 2)**

(🔴 High Priority) Train the model using the dataset with batch processing in Google Colab.

(🔴 High Priority) Debug issues related to dark/gray outputs and loss function tuning.

(🟡 Medium Priority) Optimize training parameters for better image quality.

Sprint 3 – UI Development & Integration (Day 2)

(🔴 High Priority) Develop a simple Streamlit UI for user input (upload person & clothing images).

(🔴 High Priority) Integrate the trained model with the frontend for real-time inference.

(🟡 Medium Priority) Improve the UI layout and add a download button for results.

Sprint 4 – Testing, Enhancements & Deployment (Day 3)

(🟡 Medium Priority) Test multiple cases with different clothing and body types.

(🟢 Low Priority) Fix UI/UX issues and ensure smooth user interaction.

(🟢 Low Priority) Deploy the final system using Streamlit Sharing or a cloud platform.

## **Phase-5: Project Development**

### **Objective:**

Implement core features of the Virtual Try-On system along with the frontend UI.

### **Key Points:**

1. **Technology Stack Used:**
   * **Frontend: Streamlit**
   * Deep Learning Framework: PyTorch
   * Dataset: VITON-HD
2. **Development Process:**
   * 1. Implement Streamlit UI for user input.
   * 2. Connect UI with the trained deep learning model.
   * 3. Optimize image display and inference time.

## **Phase-6: Functional & Performance Testing**

### **Objective:**

Ensure that the AutoSage App works as expected.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Category** | **Test Scenario** | **Expected Outcome** | **Status** | **Tester** |
| TC-001 | Functional Testing | Upload user and clothing images | Both images should load correctly | Passed | Thirumalesh |
| TC-002 | Model Testing | Generate virtual try-on output | Realistic try-on image should be displayed | Failed | Thirumalesh |
| TC-003 | UI Testing | Streamlit UI should be responsive | Works smoothly on desktop & mobile | ⚠ Needs Optimization | Thirumalesh |
| TC-004 | Model Performance | Inference time < 2 sec | Output should be generated quickly | Failed | Thirumalesh |

## **Final Submission**

1. **Project Report Based on the templates**
2. **Demo Video (3-5 Minutes)**
3. **GitHub/Code Repository Link**
4. **Presentation**